Power Management - Multi-Channel Fan Speed Control System

AN2357

Author: Volodymyr Sokil

Associated Project: Yes
Associated Part Family: CY8C2xxxx
GET FREE SAMPLES HERE
Software Version: PSoC® Designer™ 4.2
Associated Application Notes: AN2180, AN2246, AN2249, AN2314
PSoC Application Notes Index

Application Notes Abstract

This Application Note describes the implementation of a multi-channel fan speed control system. The proposed implementation shows how to use low-cost, two-wire fans in different temperature control systems.

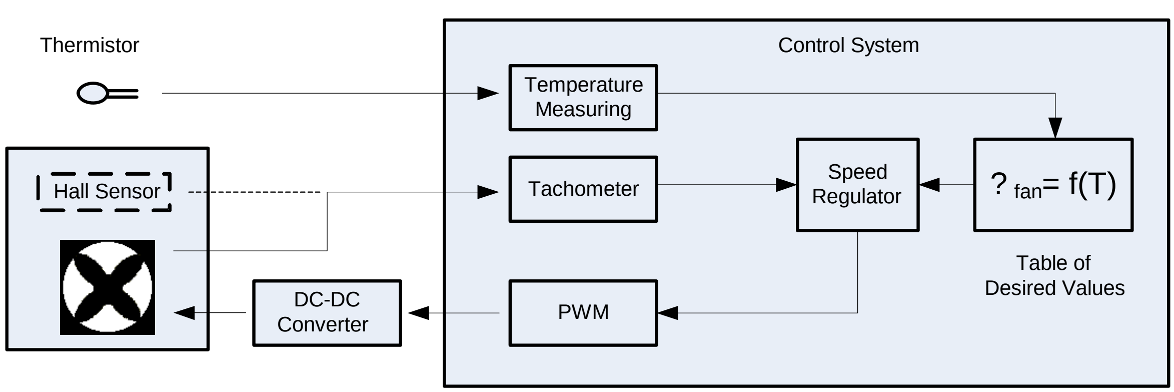
Introduction

Increasing the performance of electronic systems causes an increase in power consumption and, consequently, heat dissipation. The most popular solution to thermal problems is the use of air-cooling, provided by multiple fans.

The flowchart of a typical temperature control system is shown in Figure 1. The actual temperature and fan speed values are used as input parameters for the regulation system. This regulator provides the desired fan speed by changing the output of the PWM duty cycle signal. The number of independent channels and regulator type can change according to the task specification.

Traditionally in such systems, three- or four-wire fans with separate tachometer outputs are used. There are many different low-cost models of fans without separate tachometer outputs. This Application Note demonstrates how to extract the rotation speed of a two-wire fan by processing its supply current ripple.

Figure 1. General Temperature Assurance System Flowchart



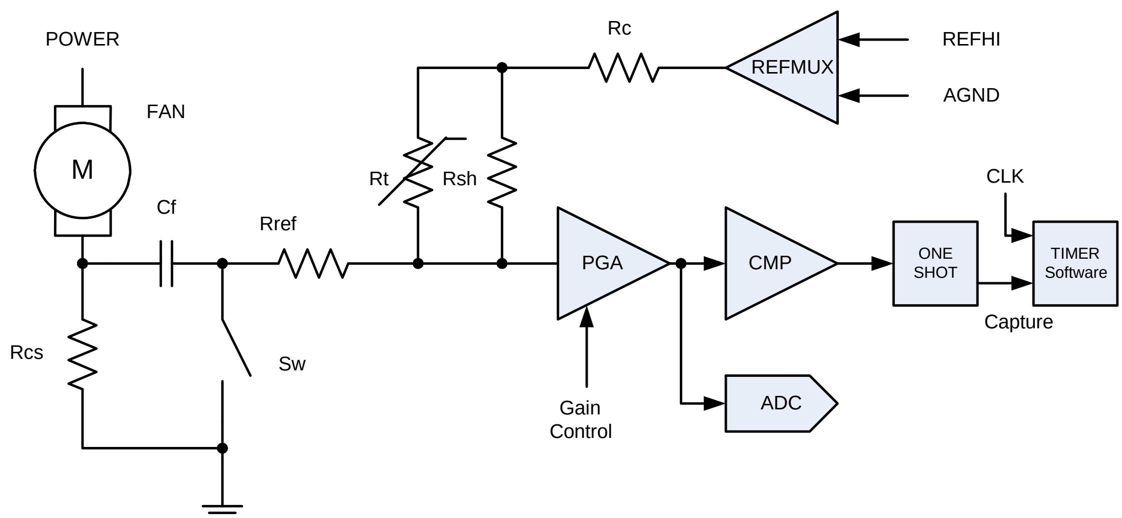
Fans Speed Control System Implementation

A fan motor has several windings that are commutated by an IC during fan revolution. These commutations cause the fan's current consumption to fluctuate. If the frequency of the fluctuations and the number of permanent magnetic poles of the rotor is known, the mechanical rotation speed of the fan can be easily calculated.

The flowchart of a measuring system for one separate fan control channel is shown in Figure 2. Such a scheme provides a temperature measuring circuit and a tachometer connection circuit via the same single analog input port. The measuring mode is selected by switching the open-drain digital output, Sw. Let's briefly consider these modes separately.

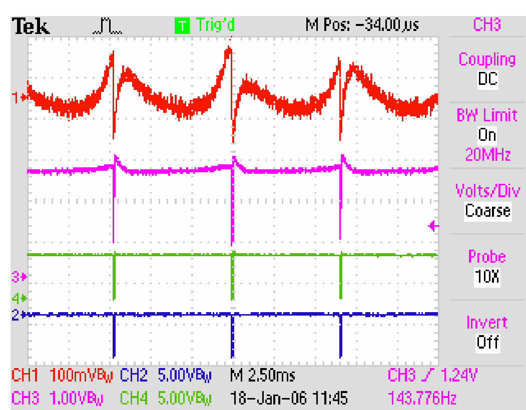
The circuit consists of a voltage divider, built on Rref, Rt, and Rsh, and a decoupling capacitor, Cf. The temperature measuring mode requires the switch, Sw, to be closed. The decoupling capacitor, Cf, removes the DC component of current sense resistor, Rcs, signal. Temperature is measured by the resistance divider, which consists of four resistors, Rc (constrain resistor), Rt (thermistor), Rsh (shunt resistor), and Rref (reference resistor). The shunt resistor, Rsh, prevents the divider coefficient from increasing when the thermistor, Rt, resistance is high (temperature is low). The constrain resistor, Rc, in turn prevents the divider coefficient from decreasing when the thermistor, Rt, resistance is low (temperature is high). The PGA only buffers the input signal, so its gain is set to 1. This temperature measuring technique is similar to the technique described in AN2314, "Thermistor-Based Temperature Measurement in Battery Packs." This technique applies two different voltages to the thermistor and measures the difference.

Figure 2. One Channel Measuring Circuit Flowchart



When switch Sw is open, the external circuit works in tachometer mode. The signal from the current sense resistor, R_{cs}, via the differentiator, built on $C_f.R_t$ R_{sh} R_{ref}, is fed to the PGA. The PGA gain is set to 24. It provides the necessary signal level for normal threshold comparator CMP operation. The CMP forms a digital signal from the amplified signal. The one-shot module suppresses possible signal bounce. More detailed information about one-shot modules can be found in AN2249, "User Module as a One-Shot Pulse Width Discriminator and Debouncer." Using the one-shot output signal as the capture signal for the timer gives the time period between two pulses, and, therefore, the fan motor speed. The oscilloscope waveforms of the tachometer function are shown in Figure 3.

Figure 3. Tachometer Operation Waveforms



CH1 – Fan current consumption
CH2 – PGA output
CH3 – CMP output
CH4 – One-shot output

This technique has a very useful feature. If the environment temperature increases, the voltage of the fan power supply is increased by a control loop. This, in turn, causes an increase in the amplitude of the measured current signal. This increase is compensated by a simultaneous decrease in thermistor resistance, and vice versa. So, the signal from current sense resistor, R_{cs}, would have approximately the same amplitude in a wide range of temperatures and fan power supply voltages. This has a positive influence on the operational reliability of the tachometer.

Since the fan motor uses a 12V power supply, the output PWM signal of the control system cannot be used directly. A step-up switching regulator is used to control the fan drive voltage. A high-frequency (1-1.5 MHz) PWM signal must be used so the regulator coil can be small. This reduces both the resulting inductance and the cost of the regulator.

The 8-bit PWM User Module can be used as the signal. It has a maximum frequency of 48 MHz. When using a 1-MHz output signal, a theoretical maximum of 48 steps of control are available. But the fan starts operation from some threshold power supply voltage (not zero). Moreover, the fan speed does not increase linearly as the power supply voltage is increased. These factors reduce the step count to approximately 20.

Software sigma-delta modulation is used to increase the resolution of the output voltage. The periodic change of the least significant bit of the PWM duty cycle register gives an intermediate (fractional) value between two neighboring whole values. The buck converter filter smothers the ripples (see Reference [2]). More detailed information about sigma-delta modulation can be found in AN2246, “PWM Source - High Frequency, High Resolution” and Reference [1].

The firmware of this modulator is easy to understand. The resolution of the PWM is extended up to 13 bits (5 bits hardware plus 8 bits software) in the firmware. The duty cycle of the PWM output signal is set with a 2-byte variable. The most significant byte (PWM MSB) defines the duty cycle of the output PWM signal. The least significant byte (PWM LSB) is used to manipulate the least significant bit of the PWM MSB. The frequency of these updates is defined by the system counter VC3 interrupt frequency and is equal to 10 kHz in this project.

Code 1. Software Sigma-Delta Modulator

```
push A
mov A, [bCH0_Tmp_mod]
add A, [_wCH0_PWM_value + 1] // LSB value
mov [bCH0_Tmp_mod], A
mov A, [_wCH0_PWM_value]     // MSB value
adc A, 0
mov REG[PWM_FAN0_COMPARE_REG], A
pop A
```

Implementation of each fan control channel needs three PSoC pins (analog input, mode switch and PWM output) and one digital block for an 8-bit PWM. Shared resources for the system are:

- Two pins for the I2C bus
- One analog output pin for AGND/REFHI
- Two digital blocks for the ADC and One-Shot modules
- Three analog CT blocks for the CMP, PGA, and RefMux user modules
- One or two analog SC blocks for the ADC

The system counter, VC3, is used as the tachometer timer. Depending on channel demands, different PSoC device families can be used to implement the control system. Table 1 contains information about the maximum number of channels for all PSoC device families.

Table 1. Number of Channels Depending on PSoC Family

Recommended Part	Maximum Number of Channels
CY8C21xxx	Not Applicable
CY8C24x23A	2
CY8C27x43	6
CY8C29x66	7 (14[1])

1 Since only eight analog inputs are available, seven of 14 fans must have separate tachometer output.

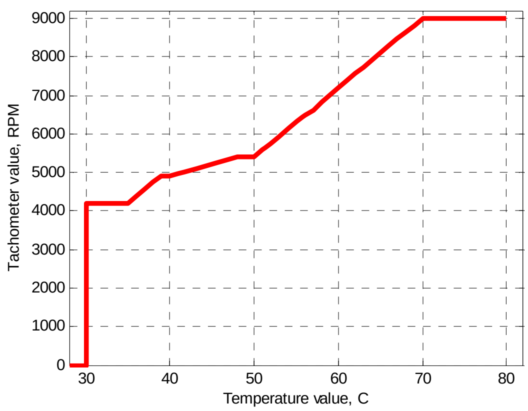
The schematic of a six channel fan control system based on CY8C27443-24PXI is shown in Appendix A on page 5. Note that the buck DC-DC converter in continuous conductivity mode operates as a voltage regulator. The output voltage, V_{out} in this case, is the uniform function of input voltage V_{in} and PWM signal duty cycle D:

$$V_{out} = V_{in} \cdot D \quad \text{Equation 1}$$

This decreases the ripples due to load changes. More detailed information about DC-DC converter creation and operation can be found in AN2180, “Switch Mode Pump in a Step-Down Converter Using PSoC,” and Reference [2]. Note that the converter can easily be modified to support fans with different operating voltages. Recommended values of DC-DC converter components for different power supply voltages and fan power values are shown in Appendix B on page 6.

The firmware “brain” of the control system is the regulator module. The goal of regulator module is to provide the desired fan speed depending on environment temperature. An example of user-defined dependence between temperature and desired fan speed is shown in Figure 4. Desired fan speed values are manually defined in table form by the user.

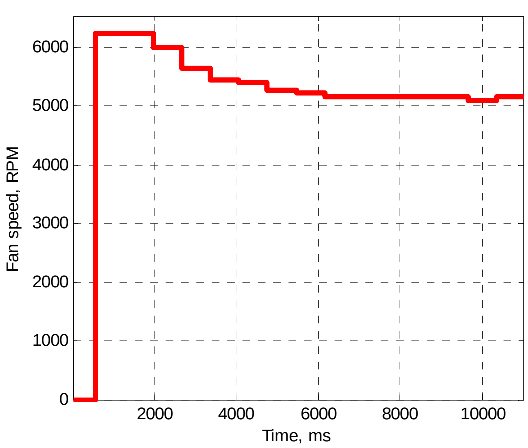
Figure 4. Desired Fan Speed vs. Temperature



There are several types of regulators. In this project the proportional-integral (PI) regulator is used. More detailed information about PI regulator operation, implementation and tuning can be found in Reference [3]. Time diagrams that demonstrate regulator operation are shown in Figure 5. An I^2C to USB bus bridge is used to debug the system and tune the regulator, as described in AN2352, "I2C-USB Bridge Usage." A quick start guide is presented in Appendix C on page 7.

Note The regulator must be tuned according to the fan type, defined by task specification. Once tuned, the regulator must be tested for operability and stability.

Figure 5. Fan Transient Process After Manual Stop



All necessary information about the current state of the fan (temperature, speed, and PWM value) is stored in a memory buffer. This can be sent to the host through the I^2C bus. The EzI^2Cs Hardware User Module is used For I^2C slave operation. Figure 6 depicts the memory buffer format.

The I^2C interface allows you to change the table of fan speed values and temperatures at runtime. The I^2C interface memory buffer contains the command register and area for the table of fan speed values vs. temperature. Table 2 contains a list of commands for this purpose.

Figure 6. I^2C Memory Buffer Format

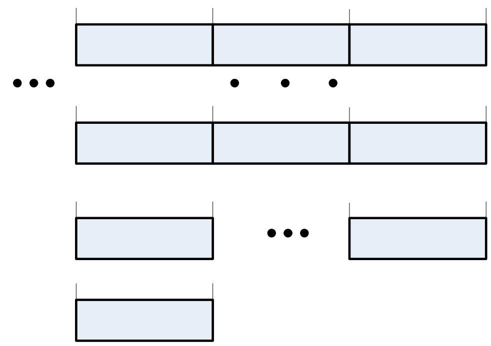


Table 2. I^2C Commands

Command	Description
0x01	Write table to RAM
0x02	Write table to FLASH
0x03	Read table from RAM
0x04	Read table from FLASH
0x05	Reload table from FLASH to RAM

Conclusion

This Application Note describes a multi-channel fan speed control system. The system uses low-cost, two-wire fans without separate tachometer outputs.

A demonstration project on a CY8C27443-24PXI was developed to test the system. The project contained six fan control channels. The system demonstrated excellent operational stability on all six channels.

References

1. Tim Wescott "Sigma-delta Techniques Extend DAC Resolution"
2. Abraham I. Pressman "Switching Power Supply Design", McGraw-Hill Professional; 2nd edition, 1997.
3. C. Hernandez-Rosales, R. Femat-Flores, and G. Quiroz-Compean "Make a PI Controller On an 8 Bit Micro"

Appendix A. Schematics

Figure 7. Schematic of the Six Channel Fan Control System

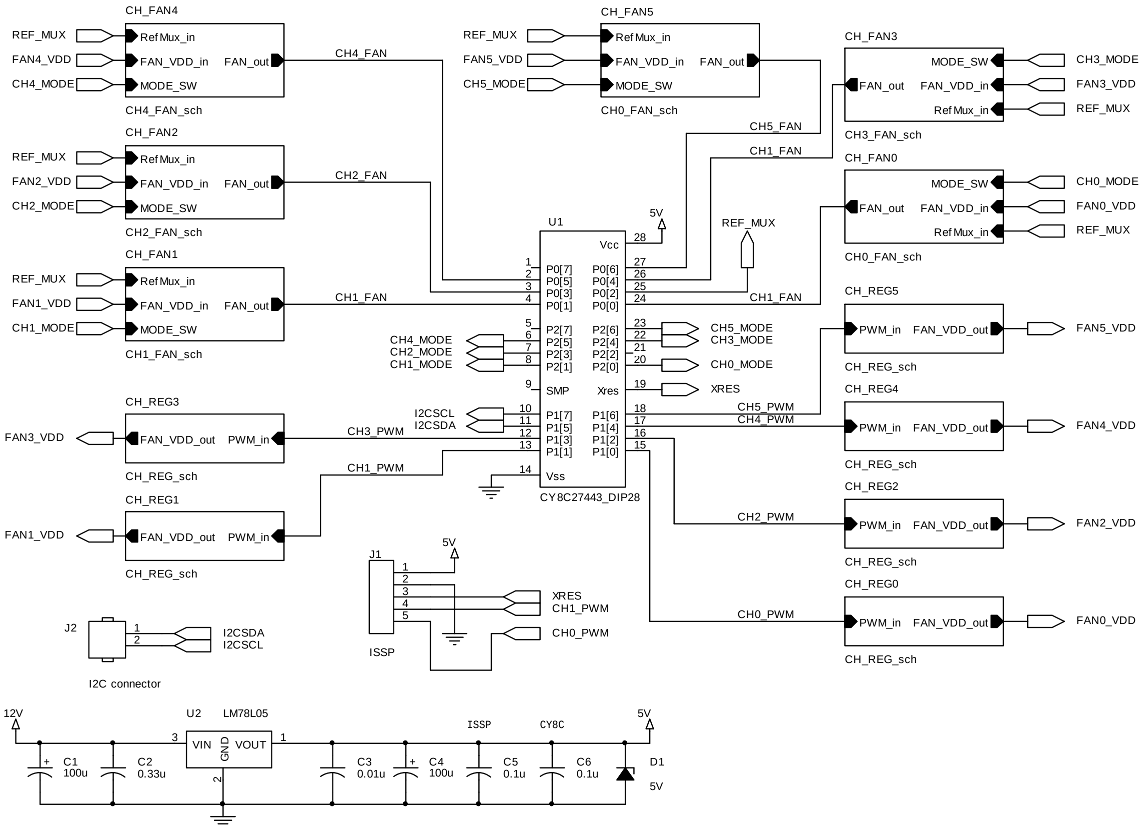


Figure 8. Schematic of CH_FAN Module

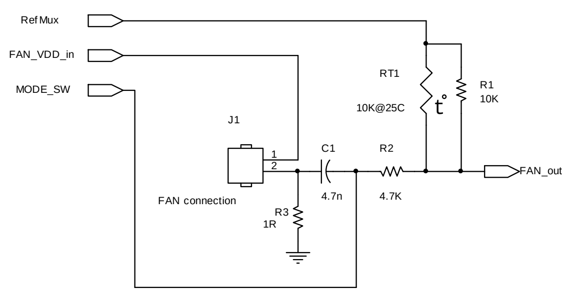
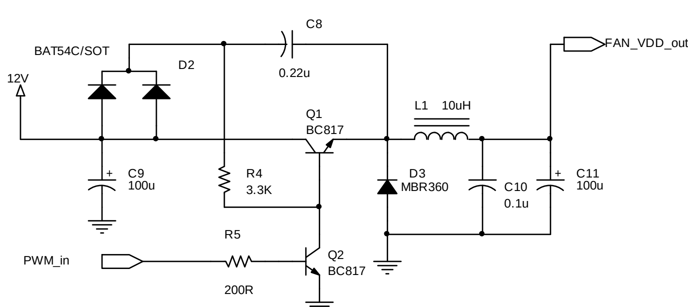


Figure 9. Schematic of the CH_REG Module (Buck Converter)



Appendix B. Recommended Values of DC-DC Converter Components for Different Fan Powers and Power Supply Voltages

Fan Voltage, V	Fan Power, W	R5[1]	D3[2]	L1[3]	Q1[4]	Q2
12	1	1.5K	1N5817 SR102 MBR120	LPO6013-153K	2N4401 MMBT4401 MPS 2222A MBT2222ADW	BC847 2N3904A MMBT3904A
	2	750				
	5	330				
24	1	5.6K	1N5818 SR103 11DQ03 MBR130 10JQ030	DO1605T-333M		
	2	3.3K				
	5	1.5K				
	10	750				
30	1	7.2K	1N5819 SR104 11DQ04 11JQ04 MBR140	DO1605T-473M		
	2	4.3K				
	5	2K				
	10	820				
48	1	11K	MBR160 SR106 11DQ06 11JQ06	DO1605T-104M	MMBTA05 BC489B	BC846
	2	6.8K				
	5	3K				
	10	1.3K				

1 Resistance value is calculated with the following equation:

$$R_5 = \frac{V_{FAN}}{2 \cdot \left(I_{L1\max} / h_{FE\min} \right)},$$

where h_{EFmin} is the minimum DC current gain of Q_1.

2 Repetitive reverse voltage value V_{RRM} must be less then V_{FAN}.

3 Maximum current value of inductor is calculated with the following equation:

$$I_{L1\max} = V_{FAN} \cdot \left[\frac{D}{R_{FAN}} + \frac{D(1-D)}{2\, F_{PWM}\, L} \right];,$$

$$R_{FAN} = \frac{V_{FAN}^2}{P_{FAN}}$$

where D=0.9.
F_{PWM} is PWM signal frequency.
L is inductance value.
I_{L1max} value must be less then coil saturation current, I_{SATUR}.
Inductors are selected from the Coilcraft Inc. catalog.

4 The main constraints for selecting Q_1 are:

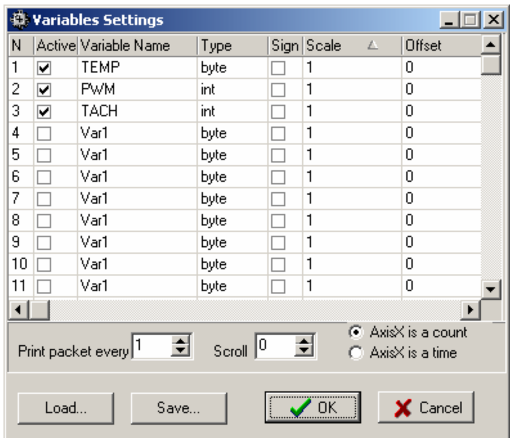
- collector-emitter voltage $V_{CEO} > 1.25 \cdot V_{FAN}$;
- collector DC current $I_C > I_{FAN}$;
- switching time < 250 ns (for PWM signal frequency 1 MHz)

Appendix C. I2C-USB Bridge Settings Quick Start Guide

An I2C-USB bridge can be used to tune and debug the fan controller, as described in AN2352. To set up the I2C-USB bridge:

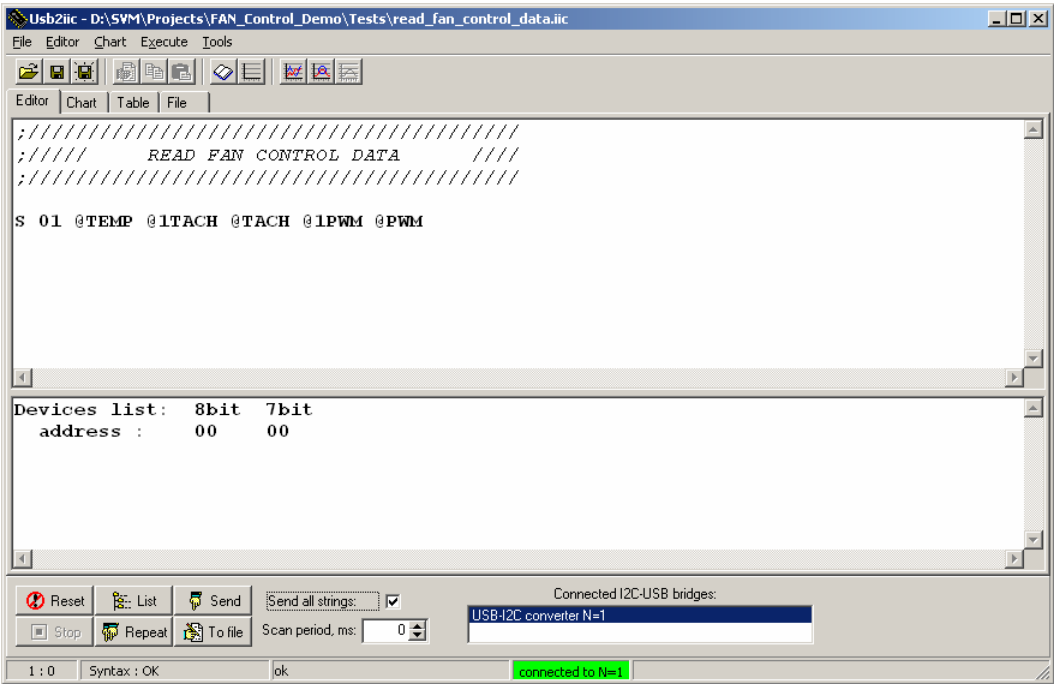
1. Connect the test fan to the fan control demonstration board.
2. Connect the power supply to the fan control demonstration board.
3. Connect the I2C–USB bridge board via the I^2C connector to the fan control demonstration board.
4. Connect the I2C–USB bridge board via the USB connector to the PC.
5. Run the I2C–USB bridge software.
6. Set the I^2C bus speed to 100K. From the **Tools** menu, select **IIC Speed – 100K**.
7. On the **Chat** menu, click **Data settings** and declare your variables.

Figure 10. Variable Settings Window



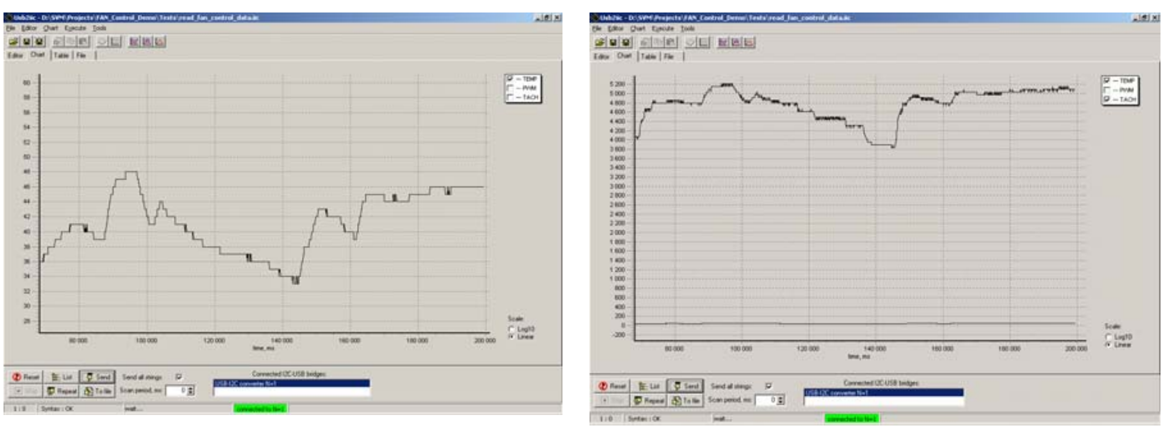
1. Click the **List** button and check the I2C–USB bridge address. It must be 0x00.
2. Click the **Editor** tab. On the **File** menu, click **Open File**. Locate the file, *fan_control_data.iic*, and open it. This loads the command file into the Command Window:

Figure 11. Command Line Example



1. Select the **Send all strings** check box.
2. Click the **Send** button to start data transmission.
3. Select the **Chart**, **Table**, or **File** tab to display realtime data.

Figure 12. Operation Profile Example



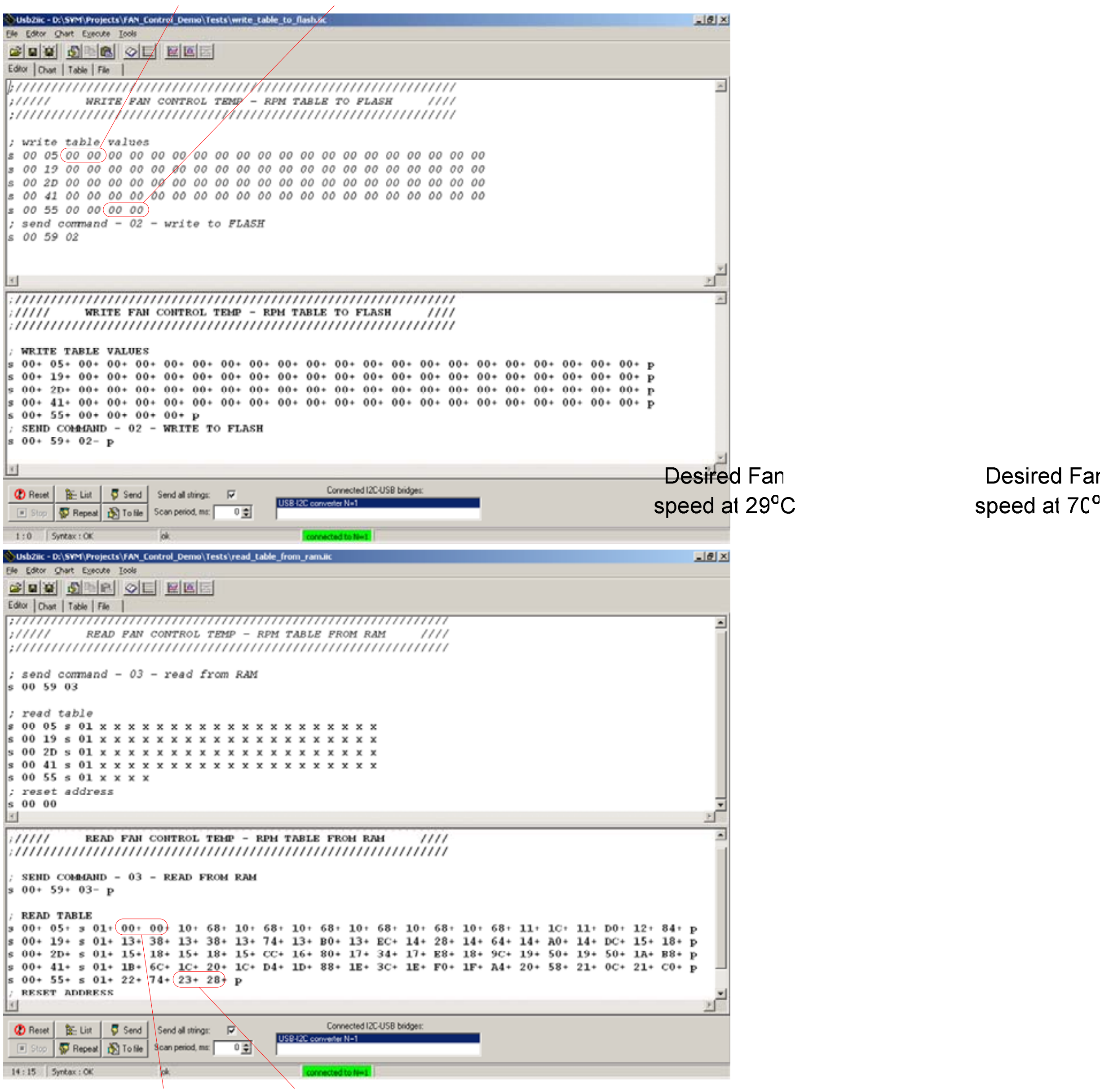
You can download your own table of desired fan speed vs. temperature to, or read the existing table from, RAM or FLASH memory.

To download a table or read an existing table, select the appropriate command file:

1. Click the Editor tab.
2. On the File menu, click Open File.

3. Click the command file that wish to use:
   - *write_table_to_ram.iic*
   - *write_table_to_flash.iic*
   - *read_table_from_ram.iic*
   - *read_table_from_flash.iic*
4. Verify that the **Send all strings** check box is selected. This option must be enabled.
5. Click the **Send** button to start data transmission.

Figure 13. Setting Custom Fan Speed vs. Temperature Table Script Example



About the Author

Name: Volodymyr Sokil
Title: Post-Graduate Student
Background: Volodymyr earned a diploma in computer-engineering in 2001 from National University "Lvivska Polytechnika" (Lviv, Ukraine), and is currently a post-graduate student at this University. His interests include embedded systems design and information security.
Contact: sokilvm@ukr.net

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.

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