Implementation and Analysis of Fan Controlling System on Programmable System on Chip (PSoC)

Km. Mamta

Abstract— Increases in circuit density and clock speed in modern computer system have brought thermal issues into the spotlight of VLSI design. Local overheating in one part of a high-density circuit, such as a CPU or high-speed data routing circuit can cause a whole system to crash. Heat sinks and fan cooling for critical devices are the traditional methods of heat dissipation. A fan controller is an essential component of such high-quality thermal management systems. This paper gives an alternative solution to thermal management system by using PSoC technology, which reduce chip count, power consumption and design time of a product. We also use temperature sensor here for reducing noise according to ambient condition and PWM technology to control a fan as it is more sufficient now a days for highly dense and battery operated systems.

Index Terms— Noise, Programmable System on Chip (PSoC), Pulse Width Modulation (PWM), Thermal Management System,

1 Introduction

Increases in circuit density and clock speed in modern computer system have brought thermal issues into the spotlight of VLSI design. Local overheating in one part of a high-density circuit, such as a CPU or high-speed data routing circuit can cause a whole system to crash. Heat sinks and fan cooling for critical devices are the traditional methods of heat dissipation [1].

A fan controller is an essential component of such high-quality thermal management systems. Linear drive and PWM (Pulse-Width-Modulation) drive are two major classifications of fan controllers. Linear drive provides a simple circuit design by using a DAC (digital-analog converter), but it requires a tachometer output as a feedback mechanism. However, PWM drive is mare attractive since it results in less power consumption for the fan controller, with a linear correlation between the input controlling signal and the active duty cycle of the fan [1].Acoustic noise produced by fan is not only a nuisance to the listener, but can also pose a significant health risk to humans. In the workplace in particular, acoustic noise emitted from electric motors has become a recognized health hazard, especially for persons who are exposed to noisy environments for long periods of time [2]. There is main three factors which generate noise in a fan.

- The first one is fan speed.
- The second is the magnetic noise, generated by vibration of the core due to the electromagnetic forces in the air-gap of the motor.
- The third is the noise generated by the bearings of fan motor

Regulating fan speed according to the temperature inside an enclosure is the most effective method utilized to cool electronic products. PWM is the preferred approach to regulating motor speed for three main reasons. First, PWM is energy efficient since it doesn't generate additional heat. Second, it improves the acoustics of the fan with high frequency driving signals. Finally, it provides thermal engineers with added operational speed bandwidth because the fan can run at either remarkably low or high speeds [3].

Presently, we give an alternative solution to thermal management system by using PSoC as a controller, which reduce chip count, power consumption and design time of a product. We also use temperature sensor here for reducing noise according to ambient condition and PWM technology to control a fan as it is more sufficient now a days for highly dense and battery operated systems. Fan noise in a system is dependent on speed of fan, so it can be reduced by reducing fan speed. And when we reduce speed correspondingly power is also get reduced. Hence we take fan speed as a variable for analyzing fan controlling system using PSoC. We use an axial fan as for cooling approach, because compared to a centrifugal blower, axial fans offer higher volumetric flow rates (VFR's) at relatively lower pressure drop conditions.

2 Theory

2.1 Need for fan control

As modern PCs grow more powerful so do their requirements for electrical power. Computers convert most of this electrical power into heat generated by all major components. Some early generation PCs did not need active ventilation. Power supplies eventually needed forced cooling. The byproduct of increased heat generation is that the fan(s) need to move in-

Km. Mamta currently working as an Assistant Professor in electronics and communication department of Maharaja Agarsain Institute of Technology, Ghaziabad, India, E-mail: kmmamtasiddhu@gmail.com

creasing amounts air and thus need to be more powerful. Since they must move more air through the same area of space, fans will naturally become noisier.

In fact, if one installs extra fans in a PC case, the noise levels can reach 70 dB. Since fan noise increases with the fifth power of the fan [4] rotation speed, reducing rotations per minute (RPM) by a small amount potentially means a reduction in fan noise. This must be done cautiously, as excessive reduction in speed may cause components to overheat and be damaged. If done properly fan noise can be drastically reduced.

In addition to that if fan get stall then we can't find fan failure without monitoring or controlling a fan. Suppose a fan get stall, then it causes a component to get overheat and be damaged. That's why the need of fan controlling comes into mind.

2.2 Pulse-Width Modulation

Pulse-width modulation (PWM) is a common method of controlling computer fans. A PWM-capable fan is usually connected to a 4-pin connector (Pin out: Ground, +12V, sense, control). The sense pin is used to relay the rotation speed of the fan and the control pin is an open-drain or open-collector output, which requires a pull-up to 5V or 3.3V in the fan. Unlike linear voltage regulation, where the fan voltage is proportional to the speed, the fan is driven with a constant supply voltage; the speed control is performed by the fan based on the control signal. The control signal is a square wave operating at 25 KHz, with the duty cycle determining the fan speed. Typically a fan can be driven between about 30% and 100% of the rated fan speed, using a signal with up to 100% duty cycle. The exact speed behavior (linear, off until a threshold value or a minimum speed until a threshold) at low control levels is manufacturer dependent [5].

2.3 Programmable System On Chip

PSoC stands for Programmable System on Chip made by Cypress Semiconductor. Cypress also calls their PSoC devices mixed-signal arrays. PSoC is a true programmable embedded system-on-chip integrating configurable analog and digital peripheral functions, memory and a microcontroller on a single chip. A PSoC device consists of several sub systems on a single chip. The systems them self and the connections between them may be configured to achieve a system on chip. Often times for the embedded designer, using a PSoC means that the bill of materials (BOM) for their product is reduced because the system that used to take several ICs can now be implemented in a single PSoC device. While BOM count reductions vary, with the right application reductions can be significant [6].

You can read detailed explanation of PSoC Technology on Cypress Semiconductor Website [10]. There are three types of architecture present in the market, which are as follows:

- PSoC1 [7]
- PSoC3 [8]
- PSoC5 [9]

Block diagram of proposed fan controlling system using PSoC is shown in figure 1 and we are using PSoC 3 architecture for implementation.

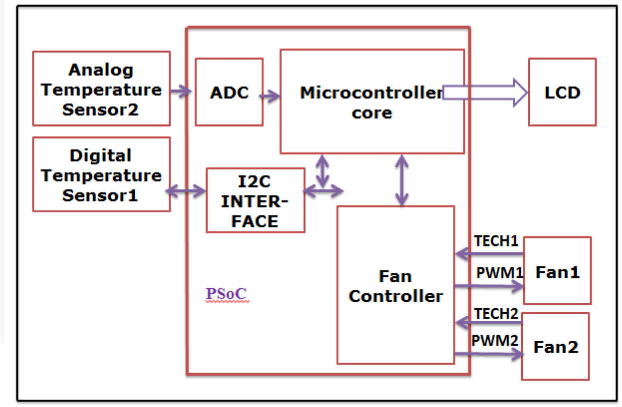

Fig. 1. Block diagram of Fan controlling system on PSoC

3 METHODOLOGY

To implement this project research methodology is divided into two parts:-

- Software implementation
- Hardware implementation

3.1 Software implementation

- In software implementation first Select the needed user modules and connect them using wire tool in PSoC Creator Topdesign file.
- Next step is to configure the Design wide Resource file for pin configuration.
- Next, Write a C code in Main.c file.
- Then, Build the project.
- If there is no error, then Program the target Device otherwise correct the errors, rebuild the project and program the device.

3.2 Hardware implementation

- In hardware implementation first step is to connect external Component to PSOC on DVK (Development kit) prototype area.
- Second step is to connect input from prototype area to PSOC pins and output from PSOC pins to prototype area using jumper wires.
- Next, Connect Miniprog3 to PSOC using 10-pin ribbon cable and USB cable to PC.
- Then, Connect power supply to DVK Board.
- And Program the target device.
- Reset the PSOC, see the results and analyze them.

4 HARDWARE IMPLEMENTATION

We are using two temperature sensors for two fans independently; means digital temperature sensor is for fan2 and analog temperature sensor for fan1. An ADC is required inbuilt to PSOC to convert analog input into digital form from analog sensor and a I2C component is required to communicate with the I2C (TMP175) temperature sensor. We define two thermal zones corresponding to each fan. Zone2 has digital temperature sensor and fan2 and Zone 1 has an analog temperature sensor and fan1. How temperature to speed is define for these Zones are shown in table 1.

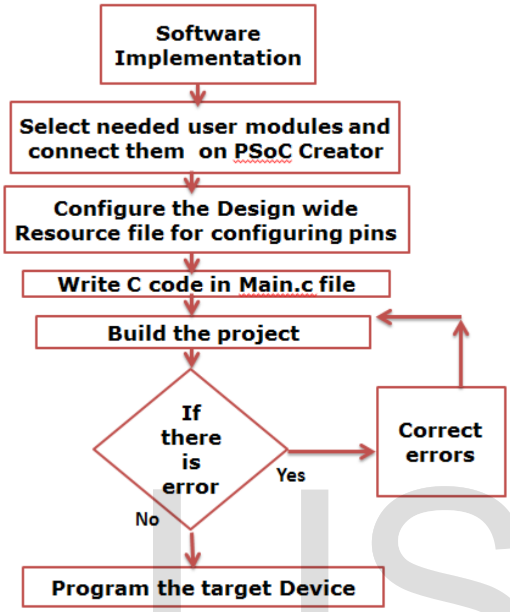
Software Implementation
Select needed user modules and connect them on PSoC Creator
Configure the Design wide Resource file for configuring pins
Write C code in Main.c file
Build the project
If there is error
Yes
No
Correct errors
Program the target Device

Fig. 2. Block diagram of Software implementation

4.1 Working

The working of fan controlling system implemented on PSoC is explained via flow chart. The flow chart of the fan controlling system is shown in figure 4. In this first we initialize all the on chip components and variables then start a 500 ms timer and check for it. If it goes expire then calculate zone1 and zone2 temperature and corresponding fan speed from lookup tables otherwise do nothing. After calculating speed if need to change then send a command to fan controller to change it and also launch the next temperature measurement for next time otherwise do nothing. Finally check for switch whether a switch is pressed or not and according to that switch update the LCD for that particular zone.

Table 1: Temperature to speed lookup table for both zones

ZONE 1		ZONE 2	
Temperature	Fan Speed	Temperature	Fan Speed
0-14	1500	0-19	1500
15-24	2000	20-24	2000
25-34	2500	25-29	2500
35-54	3000	30-34	3000
55-74	3500	35-39	3500
75-99	3800	40-44	3800

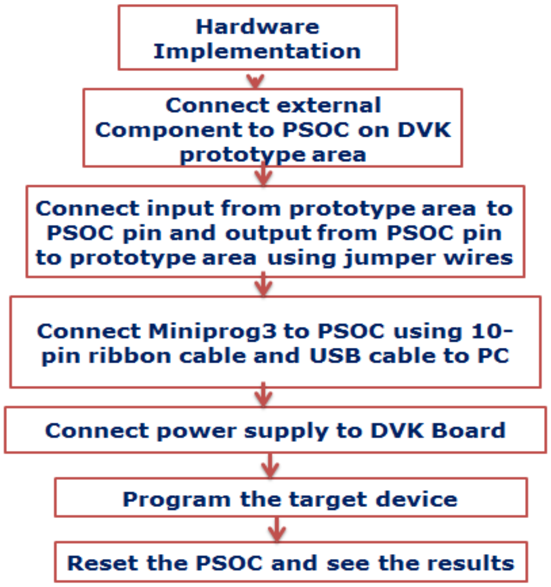
Hardware Implementation
Connect external Component to PSOC on DVK prototype area
Connect input from prototype area to PSOC pin and output from PSOC pin to prototype area using jumper wires
Connect Miniprog3 to PSOC using 10-pin ribbon cable and USB cable to PC
Connect power supply to DVK Board
Program the target device
Reset the PSOC and see the results

Fig. 3. Block diagram of Hardware implementation

The circuit diagram for fan controlling system implemented on PSoC Creator Software is shown in figure 5. We connect pin 1 (SDA) and pin 2 (SCL) of TMP175 digital temperature senor to P0.6 and P0.7 respectively on DVK. As these lines are open drain type and need external pull ups so we connect resistance between pins and power supply. Pin 3 is an alert signal, pin 4 (GND) is connected to ground. Pin 5,6,7 are the address lines and as we are using only 1 digital sensor so we connect these line to ground and address for this configuration of IC is 48H (for more detail please refer datasheet of TMP175) and pin 8 is connected to power supply. P0.0 and P0.1 pin of PSoC is connected to Tech_1 and Tech_2 signal of fan controller. P0.2 and P0.3 are connected to output of fancontroller. We are using a potentiometer on DVK as a sensor for testing purpose we can also replace this potentiometer to any other analog temperature sensor like diode, thermistor etc after that. And we connect potentiometer on P1.7 pin of PSoC. For changing the thermal zone we connect a switch on P0.5 pin of PSoC.

The first zone, Zone 1, combines temperature measurements from two temperature sensors (1 analog and 1 digital). The analog sensor is simulated using a variable potentiometer to allow easy demonstration of fan control over a wide simulated temperature range without the need of an environmental chamber to cycle through temperatures. In Zone 1, the temperature senso1rs are combined using a weighted average where the potentiometer is given 90% weight and the digital I2C temperature sensor is given 10% weight. Adjust the potentiometer (R20 on the CY8CKIT-001 PSoC Development Kit) to vary the temperature value in the approximate range of 15 to 100 degrees C.

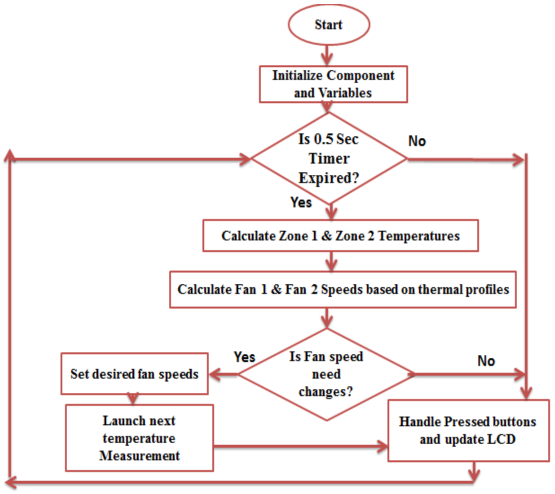
Start
Initialize Component and Variables
Is 0.5 Sec Timer Expired?
No
Yes
Calculate Zone 1 & Zone 2 Temperatures
Calculate Fan 1 & Fan 2 Speeds based on thermal profiles
Yes
Is Fan speed need changes?
No
Set desired fan speeds
Launch next temperature Measurement
Handle Pressed buttons and update LCD

Fig. 4. Flow chart of fan controlling system

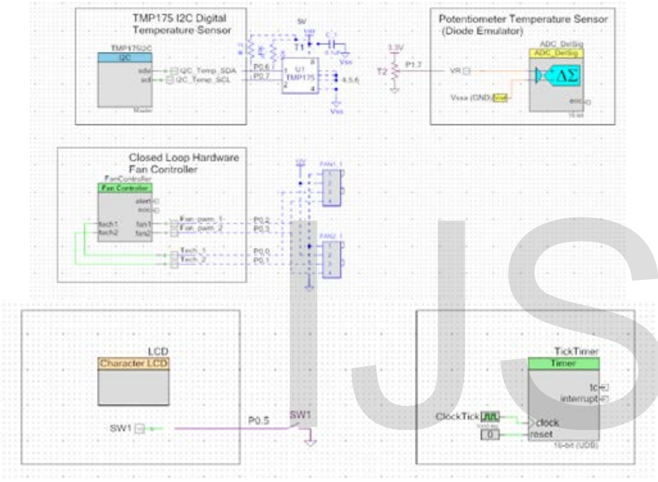
TMP175 I2C Digital Temperature Sensor
I2C_Temp_SDA
I2C_Temp_SCL
U1 TMP175
Potentiometer Temperature Sensor (Diode Emulator)
ADC_DelSig
Closed Loop Hardware Fan Controller
Fan Controller
LCD
Character LCD
SW1
P0.5
TickTimer
Timer
ClockTick

Fig. 5. Circuit diagram of fan controlling system on PSoC3

In, Zone 2 Note that the temperature range is very narrow and close to room temperature. This is to allow for simple testing at room by just touching a temperature sensor with a warm finger to cause a fan speed change. In Zone 2, the temperature sensors are combined using a weighted average where the digital I2C temperature sensor is given approximately 99% of the weight. And the potentiometer is given 1% weight. In this example TMP175 temperature reading will dominate the overall zone temperature calculation. The hardware set up for fan controlling system on PSoC is shown in figure 6. The LCD screen displays status information about fan controlling system across two screens. The user can cycle through the status screens by pressing SW1 on the CY8CKIT-001 PSoC Development Kit. The screens are:

- **Screen 1 - Zone 1 Summary:** This screen displays the current status of Zone 1. Line 1 displays the zone number, the current Composite zone temperature and the zone temperature calculation algorithm used. Line 2 displays the fan number, the desired fan speed for Zone 1.
- **Screen 2 - Zone 2 Summary:** This screen displays the current status of Zone 2. Line 1 displays the zone number, the current Composite zone temperature and the zone temperature calculation algorithm used. Line 2 displays the fan number, the desired fan speed for Zone 2.

The output observed in zone 1 by varying potentiometer on DVK, the output for zone 1 at 24 and 36°C is shown in figure 7, 8 respectively. The output for zone 2 at 31° C is shown in figure 9.

5 RESULTS AND DISCUSSION

Fan controlling system By Partha Roy, Erin Wofford, Abe Arredondo for two fans is shown in figure 10. For this architecture Partha use ADM1027 fan controller IC, this can control two fans with temperature sensors. ADM1027 IC can be connected to any host controller using I2C bus interface. It has inbuilt ADC into it and there is no need an external ADC chip. We can only connect diode as a temperature sensor to this IC.

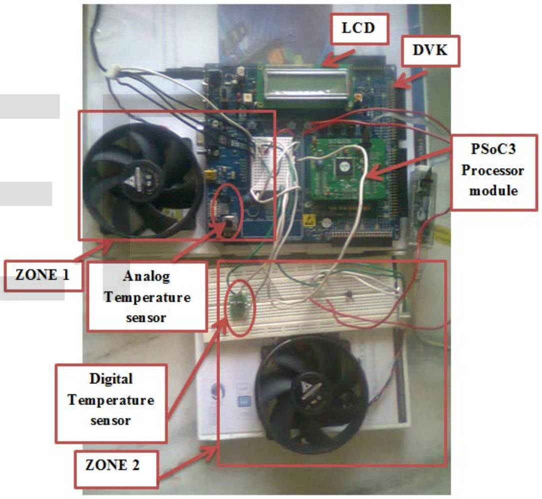

Fig. 6. Hardware setup for fan controlling system on PSoC3

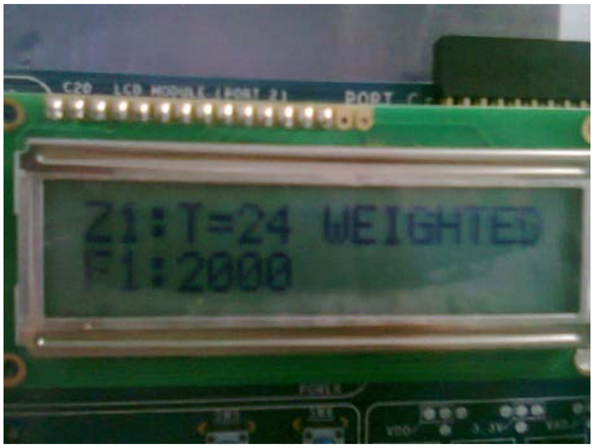

Fig. 7. Output for Zone 1 at 24° C

Fan controlling system implemented on PSoC for two fans is shown in fig. 1. As PSoC has inbuilt ADC, I2C, and fan controller IC, then there is no need for external component and it reduces chip count on PCB board. For implementation of fan controlling system on PSoC we need temperature sensor and fan externally.

As we can see from fig. 1 and 10, two controllers IC is used by Partha Roy to make fan controller system with two fan, while in proposed architecture we replace both controller with a single IC. Hence chip count on board gets reduced.

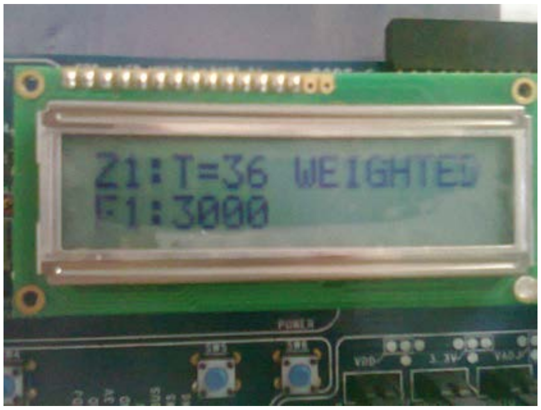

Fig. 8. Output for Zone 1 at 36° C

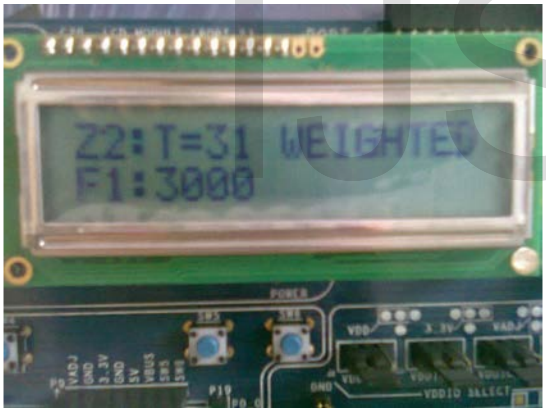

Fig. 9. Output for Zone 2 at 31° C

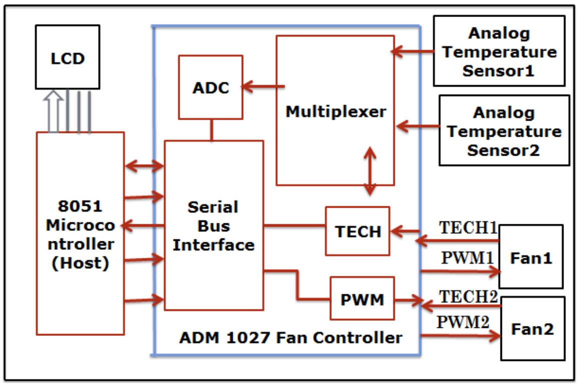

Fig. 10. Block diagram of Fan controlling system By Partha Roy, Erin Wofford, Abe Arredondo for two fan

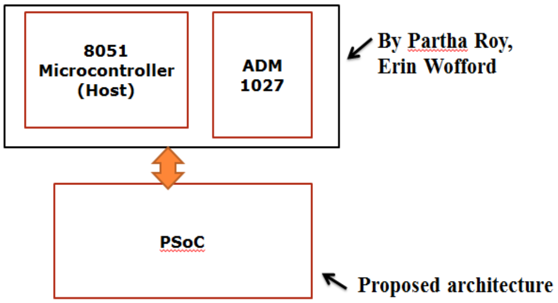

Fig. 11. Comparison of implemented fan controller on PSoC with base paper

As there is less chips on board, then power consumed by component on board is also less. This feature can be more explained by a more complex system. Suppose for a telecommunication server room we need to control 16 fans, then how the system by Partha is used to control 16 fans is shown in figure 12.

As ADM1027 is consist 3 address means we can connect 3 ADM1027 controller IC to a single host controller and 1 ADM1027 IC is responsible for controlling two fan with temperature sensors, so maximum 6 fan can be controlled by system shown in figure 12. To control 16 fans, we require 3 host controllers and 8 fan controllers IC, So total 11 chips on PCB board is required to control 16 fans by Partha architecture.

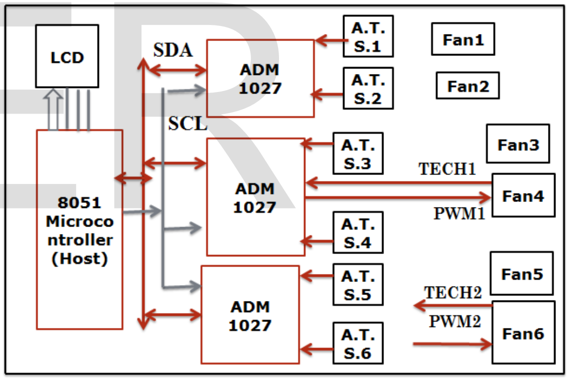

Fig. 12. Block diagram of Fan controlling system By Partha Roy, Erin Wofford, Abe Arredondo, ONLY CONTROL MAXIMUM 6 FAN WITH 1 HOST CONTROLLER

But when we make this system using PSoC, the system is look like as shown in figure 13. In PSoC, 1 fan controller component is able to control 16 fans at a time, so we need 1 fan controller inbuilt and 16 temperature sensor and 16 fans externally. As we use temperature sensor TMP175, which has 27 addresses that means we can connect 27 TMP175 IC to a single host controller via I2C bus interface. As 2 pins are require to connect a single fan, so to connect 16 fan 32 pins are sufficient and 7 pin for LCD and 2 pin for I2C bus interface, so total 42 pins are required by Fan controller on PSoC for 16 fan and there are 72 I/O pin are available.

If we compare two fan controller, then advantages of fan controller implemented on PSoC is given below-

- In the previous architecture, we can only use diode as a sensor, but using PSoC we have flexibility to choose a temperature sensor according to our project requirement.

- Alert generated by fan controller can be directly connected to port pin in PSoC.

- As fan controller IC is inbuilt in PSoC, than there is no need to implement I2C protocol in software, hence complexity in software is also get reduced.

- If we comparing chip count except fan and sensors, than 11 IC's are Required for Previous architecture for 16 fan to control, but using PSoC in place of these IC only 1 IC is used.

- Hence chip count get reduced, so power consumption and board size is also get reduced.

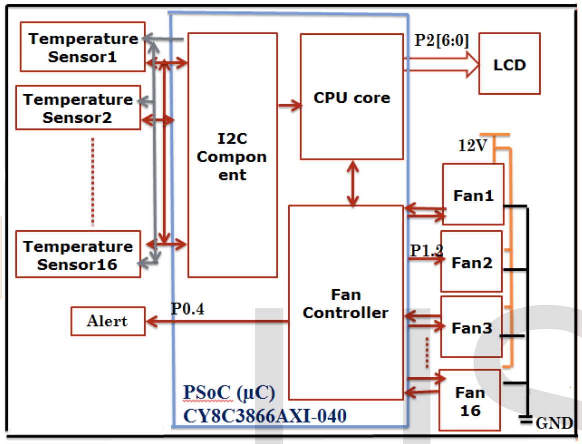

Fig. 13. Block diagram of Fan controlling system on PSoC with 16 Fan

Now how noise of a fan is reduced is explained below:-
Suppose,
$Speed_K$= Known Speed
$Speed_N$ = The new speed we are using for calculation
$Noise_K$ = Known Noise at $Speed_K$
$Noise_N$ = New Noise calculated at new speed

Calculating Noise at different speeds: The effect of speed on noise can best be seen through one of the fan laws: If we know the noise produce by a fan at maximum RPM than noise at new speed can be calculated by the following formula

$$Noise_N = Noise_K + 50 * \log\left(\frac{Speed_N}{Speed_K}\right) \quad (1)$$

We are using a fan with 45 dBA noise at its maximum speed (3800) and if we are able to reduce speed of a fan by 20% by using temperature sensor then around 5dBA noise will get reduce from uncontrolled fan. The mathematical calculation is shown below-

$Noise_N$ = 45 + 50 * log (3040/3800)
$Noise_N$ = 45 + 50 *(-0.0969)
$Noise_N$= 45 – 4.845
$Noise_N$ = 40.15 dBA

So noise gets reduced to 40.15 dBA from 45 dBA for individual fan. Similarly we can calculate noise reduction at different speed as shown in table 2.

And when we plot these points on graph the graph is look like as shown in figure 14. It is clearly shows in the graph that by reducing speed by some amount we can reduce noise of a fan, when fan is not needed to run at full speed.

Table 2: Speed to Noise reduction table

Reduction in speed (%)	New speed (RPM)	Reduction in Noise (dBA)	Total noise = 45- Reduction in noise
20	3040	4.84	40.15
40	2280	11.09	33.90
60	1520	19.89	25.11
70	1140	26.14	18.85
80	760	34.94	10.06

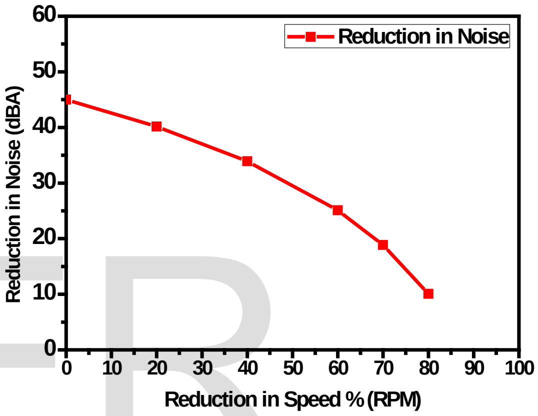

Fig. 14. Reduction in noise

When speed gets reduce, how power consumed by the fan gets reduced is explained below: Suppose,
$Power_K$ = Known Power at $Speed_K$
$Power_N$ = New Power calculated at new speed
Calculating Power Draw at different speeds: If we know power consumed by a fan at its maximum RPM then how we can calculate power at another speed is given by the following formula:

$$Power_N = Power_K \left(\frac{Speed_N}{Speed_K}\right)^3 \quad (2)$$

We are using a fan with 4.8 Watt at its maximum speed (3800) and if we are able to reduce speed of a fan by 20% by using temperature sensor then around 2.35 Watt power will get reduce from uncontrolled fan. The mathematical calculation is shown below-

$$Power_N = 4.8 * \left(\frac{3040}{3800}\right)^3$$
$$Power_N = 4.8 * (0.8)^3$$
$$Power_N = 2.45\ W$$

Table 3: Speed to power reduction table

Reduction in speed	New speed (RPM)	Reduction in power	Total power= 4.8-

(%)		(Watt)	reduction in power
10	3420	1.301	3.499
20	3040	2.342	2.457
40	2280	3.763	1.036
60	1520	4.492	0.3072
70	1140	4.670	0.129
80	760	4.761	0.038

So power gets reduced to 2.45W from 4.8W for individual fan. Similarly we can calculate power reduction at different speed as shown in table 3. And when we plot these points on graph the graph is look like as shown in figure 15. From the graph it is clearly shows that if we reduce speed of a fan linearly, then power consumed by the fan decay around exponentially.

Hence fan controlling system with temperature sensor is used where, system runs at critical condition, means if we know the temperature goes beyond acceptable limit then we can add one more task to the project that system will go to shut down mode after saving all the data. It is also useful where there is no need to run fan at full speed, so fan noise and power also get reduced. This also extends the operating life of the fan compared to systems that run all fans at full speed all of the time.

The size, shape, and pitch of the fan blades all contribute to the fan's capability to move air. Obviously, smaller fans will need to run at a higher speed than larger fans to move the same volume of air in a given time frame. Applications that are space constrained and need smaller fans due to physical dimension limitations will generate significantly more acoustic noise. This is an unavoidable trade-off that needs to be made to meet system level needs.

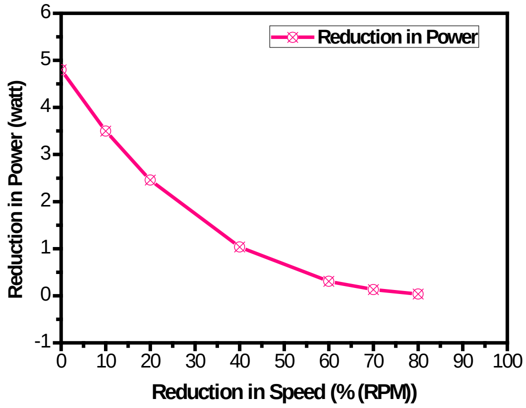

Fig. 15. Reduction in power to speed graph

6 CONCLUSIONS

This work first explains about why we need a fan control system and then we describe steps to implement fan controlling system on PSoC. In this case, we control two fans independently using PSOC with temperature sensor, but we can modify this project to number of fan as we needed just by modifying some firmware part. Here we successfully control the speed of brush less DC fan using temperature sensor and we show the effect of speed on noise and power, draw graph for various points. As a conclusion, Power Monitoring and Fan Control solutions implemented on PSoC simplify system design and deliver highest reliability for telecommunication, server and storage markets, where reliability is critical.

ACKNOWLEDGMENT

Firstly, I want to mention here that this Research work has been conducted between the time periods from July, 2012 to May, 2013 at Lovely Professional University, Punjab, India. Because of some personal problems I was unable to publish my research work at that time. Secondly, With deep sense of gratitude, I express my thanks to my Guide, Mr. Mayank Gupta, Assistant professor, Electronics and Communication Engineering Department, Lovely Professional University, who has constantly been there to provide me technical support and for giving me the opportunity to complete my research work under his direction. I also thankful to university management and Dean of the department of Electronics and Communication Engineering, Prof. Bhupender Verma , Lovely Professional University for giving me permission to commence this thesis and to do necessary research work.

REFERENCES

[1] Herming Chiueh, Louis Luh, "A Novel Fully Integrated Fan Controller for Advanced Computer Systems"

[2] Nasrin Hashemi, Robin Lisner, Donald Grahame Holmes," Acoustic Noise Reduction for an Inverter-Fed Three-Phase Induction Motor" IEEE, vol.3, 2004, pp. 2030 - 2035

[3] Abe Arredondo, Partha Roy, Erin Wofford, "Implementing PWM Fan Speed Control within a Computer Chassis Power Supply," Twentieth Annual IEEE, **Vol.** 1, APEC 2005, pp.148 - 151.

[4] http://books.google.co.uk/books?id=UocszUe2WgC&lpg=PA341&ots=EMHJiQNH__&pg=PA341#v=onepage&q&f=false

[5] "4-Wire PWM Controlled Fans Specification". 2005-09. Retrieved 2011-07-21.

[6] "Cypress Hits Half-Billion Mark in Shipments of PSoC Programmable System-on-Chip Devices" 2009

[7] PSoC® 1: CY8C28 Family Datasheet

[8] PSoC® 3: CY8C38 Family Datasheet

[9] PSoC® 5: CY8C55 Family Datasheet

[10] Cypress Semiconductor Website: http://www.cypress.com