DEMO MANUAL DC285

LTC1695
SOT23 Fan Speed Controller\LDO

Description:

This demo board shows how the LTC1695 SOT-23 provides all the functions necessary for a system controller or microcontroller to regulate the speed of a 5V fan via a 2-wire SMBus interface.

By varying the fan speed according to the system instantaneous requirements, the power consumption of a typical $\leq$ 1W 5V fan is reduced and the battery run-time for portable devices is increased. The LTC1695 accepts a 6-bit digital code, 0 - 3F hex, via the SMBus and regulates its output voltage accordingly over the range 0 - 4.992V. To eliminate fan start-up problems, users can enable the LTC1695's boost start feature which outputs 5V for 250msec before lowering to the programmed output voltage. The LTC1695 has over-current and over-temperature monitors to alert the system during fault conditions.

Also included on this demo board is the LTC1694, a dual SMBus active accelerator/pull-up device for SMBus and Philips I^{2}C applications. Linear Technology Corp. also makes available a companion demo board, the DC157, which interfaces the DC285 to a computer's parallel port for SMBus control using demo & test software.

Board Photo

Schematic

BOM

Quick Start Guide:

1. Install the DC285 software on a P.C. and attach the DC157 demo board to the P.C.'s parallel port (note: the DC285 installs both the DC285 software and the DC157 software).
2. Apply 5V to the DC285 and connect the board to the DC157 via their 5 pin connectors.
3. The operator can now either run the RunDC285 batch file located in C:\DC285 or enter discrete SMBus commands as shown below.
4. Enter SMBus commands at the DOS prompt under the DC157 directory to vary the fan speed or to read fault conditions. The SMBus Write Command format for the LTC1695 whose "Write" address is E8 is as follows:

$$\text{Smb w, E8, 'Command Byte'}$$

Examples:

    C:\DC157\smb w, E8, 00, 00, 00   sets Vout of the LTC1695 to 0V.

    C:\DC157\smb w, E8, 32, 00, 00   sets Vout of the LTC1695 to 2.5V.

    C:\DC157\smb w, E8, 3F, 00, 00   sets Vout of the LTC1695 to 4.922V.

Note: The above examples output an extra word because this is the format of the DC157 SMBus software commands. The LTC1695 which accepts Command Bytes simply ignores the Least Significant Word sent by the DC157 software.

The SMBus Read Command for the LTC1695 whose "Read" address is E9 is as follows:

$$\text{C:\DC157\smb r, E9}$$

Note: The earliest version of the DC157 software does not support SMB Byte reads.

OPERATION

SMBus Serial Interface

The LTC1695 is a SMBus slave device that supports both SMBus send byte and receive byte protocol (Figure 1) with two interface signals, SCL and SDA.

Figure 1

The SMBus Host initiates communication with the LTC1695 with a START bit followed by a 7-bit address code and a write bit. Every SMBus slave in the system compares the transmitted address code with its address. For the Send Byte and Receive Byte protocol, the write bit is low and high respectively. When selected, the LTC1695 replies with an acknowledge bit by pulling the SDA low. During a Send Byte protocol, the host then sends out an 8-bit command code. The LTC1695 acknowledges after it receives the whole command byte. At the falling edge of the acknowledge pulse, the LTC1695's DAC latches in the new command byte from its SMBus shift register.
During the Receive Byte protocol, the LTC1695 similarly acknowledges by pulling the SDA line low after the write bit. Next the LTC1695 sends out the data byte. When the host receives the whole data byte the cycle is terminated by a 'NOT Acknowledge' bit and a STOP bit.

Address, Command, Data Selection

The LTC1695's address is hard-wired internally as '1110 100' (MSB to LSB, A6 to A0). Please consult the factory for parts with alternative address codes. Page **LastPage** shows the Address, Command and Data byte Table for user reference.
As shown in Figure 1, D0 to D5 in the command code, sent from the host to the LTC1695 during Send Byte Protocol, controls the linear regulator's output voltage (and indirectly the fan speed). The LTC1695 latches D0 to D5 as DAC input data at the falling edge of the Data Acknowledge signal. The host must set 'BST' (Boot-Start Enable, bit 6) high if the LTC1695's 250ms boost-start timer option is required. All bits are reset to zero during power up reset and UVLO. As shown in Figure 1, bit 7 and 6 in the data byte register are defined as the Over-Current Fault (OCF) and the Thermal Shutdown Fault (THE) respectively. The LTC1695 sets OCF high if Iload exceeds 330mA and the THE bit high if die junction temperature exceeds 155°C. The rest of the data byte's register (bits 0 to 5) are set low during host read back.

Boost-Start Timer

DC fans typically require a starting voltage that's higher than their minimum stall voltage. For example, a Micronel 5V fan requires a 3.5V starting voltage but once started it will keep running until its terminal voltage drops below 2.1V (its stalling voltage). Thus the user needs to ensure that the fan starts up properly before programming the fan voltage to a value lower than the starting voltage. Monitoring the fan's DC current for stall conditions does not help because some fans consume almost the same amount of current at the same terminal voltage in both stall and non-stall conditions. Another approach is to detect the absence of fan commutation ripple currents. This however is complex and requires user customization for specific brands of fans' characteristics (20Ma – 75mA current spikes). Also bear in mind that these current spikes decrease with terminal voltage.

Thus the LTC1695 has a boost-start timer that offers 3 flexible solutions to the user:

a) Set the Boost-Start Timer enable bit (bit 6 in the DAC command). For every new programmed output voltage the internal timer forces Vout to full-scale (nominal at 4.922V) for 250ms before settling down to the required value. Bench data shows that a typical fan needs about 160 – 200ms to ensure full start-up.

b) Use a software timer routine inside the host controller to power the DC fan at full-scale code for a programmed time period before lowering the Vout to the desired DAC code.

c) Use a tachometer fan that feeds back its speed value to a Intel Herceta compatible system controller like the LM78/9, or ADM9240. The user detects fan stall condition and programs the LTC1695 accordingly.

Thermal Shutdown, Over-Current Clamp

The LTC1695 shuts down the PMOS pass transistor if the die temperature exceeds 155°C (20°C hysteresis). As previously mentioned, bit 6 (THE) in the SMBus data byte register is set high during thermal shutdown. During a fault condition the LTC1695's SMBus logic continues to operate so that the host can poll the fault status data.

During over-current fault, the LTC1695's over-current detector sets the data register's bit 7 (OCF) high and actively clamps the output current to 330mA. This protects the LTC1695's PMOS pass transistor. Under dead short circumstances (Vout = 0), although the LTC1695 clamps the output current, the large power chip dissipation will force the LTC1695 into thermal shutdown after a prolonged die heating time period. These LTC1695 dual protection features protect the IC itself, the fan, and more importantly alerts the host to any system thermal management fault occurrence.

During thermal shutdown recovery (125°C), the LTC1695 automatically sets the fan terminal voltage to full-scale (nominal 4.922V) for 250ms before switching to the user programmed DAC value. This eliminates potential fan stall conditions where the thermal shutdown fault previously occurred while the fan was operating at a DAC code setting below the fan start voltage.

PCB Gerber Files