

ファン制御の進歩： ファンレギュレーションの考慮

ブラシレスDCファンモータの制御方法はいろいろあります。最も簡単な方法は、単一のトランジスタを使用するオン/オフスイッチ方式です。リニアレギュレータ又はパワーオペアンプとDACを組み合わせると、デジタル設定可能な可変速度ファン制御システムを作ることができます。例えばPCの監視ICは、ファン駆動用のアンプに印加できる出力付のDACを提供しています。PWMファン制御チップも提供されていますが、これらはまだデジタルインタフェース用のDACを必要とする上、PWM法の使用には注意が必要です。最新のファン制御の方法としては、完全ファンドライバ回路付(ただしパワートランジスタを除く)の完全デジタルインタフェースICがあり、MAX1669はその一例です。又、MAX6650/MAX6651は、真のファン速度レギュレータであるという点で高度ファン制御の最先端です。

MAX1669はファン制御の最も一般的なタイプで、リニアDAC出力又はPWM出力を提供しています。MAX1669の特長は、リモートジャンクション温度センサを内蔵していることです。これは、検出ジャンクションを備えたIC又は2N3904等のセンサとして使用されるバイポーラトランジスタに接続することができます。図1の回路は、ファンのリニア制御にMAX1669を使用した例です。

MAX1669は、コスト又は技術的条件のためにファンにタコメータ出力を付けられないような場合に特に適しています。

ファン制御の問題と解決法

ファン制御に関する大きな問題は、ファンが入力電圧に対して非直線的な挙動を示すことです。図2に示すように、標準的なファンは3V~8V(ファンに依存)の電圧が印加されるまで回転し始めません。回転し始める電圧の値は、同種のファンであっても個々のファンによって異なり、温度及び使用期間によっても違ってきます。MAX1669のようにシンプルなファンドライバンプ又はPWM出力を備えたデバイスを使用する場合は、ファンを低速に設定する直前にまずフルスピードにする必要があります。どれくらい低速にできるかを予測することは難しく、大きなばらつきがあり、特定のファンの最小値を見つけるには長時間の評価作業を要します。ファンの設定速度に関わらず、これはファン速度が安定化されないオープンループシステムです。

スレッシュホールド電圧が予想できないことから、一部のファン制御ICに「Rube Goldberg」方式が導入されるようになりました。例えば、ファンが回転しているかどうかをチェックして、回転していなければ増分だけドライブを強くし、ファンが回転し始めるまでこれを繰り返すICがあります。

MAX1669とタコメータフィードバックアンプの組み合わせは、ファン速度を安定化することで、この問題の「クリーン」な解決法を提供します(図3)。しかし代償として複雑さが増すと共に、ファン制御アンプの設計

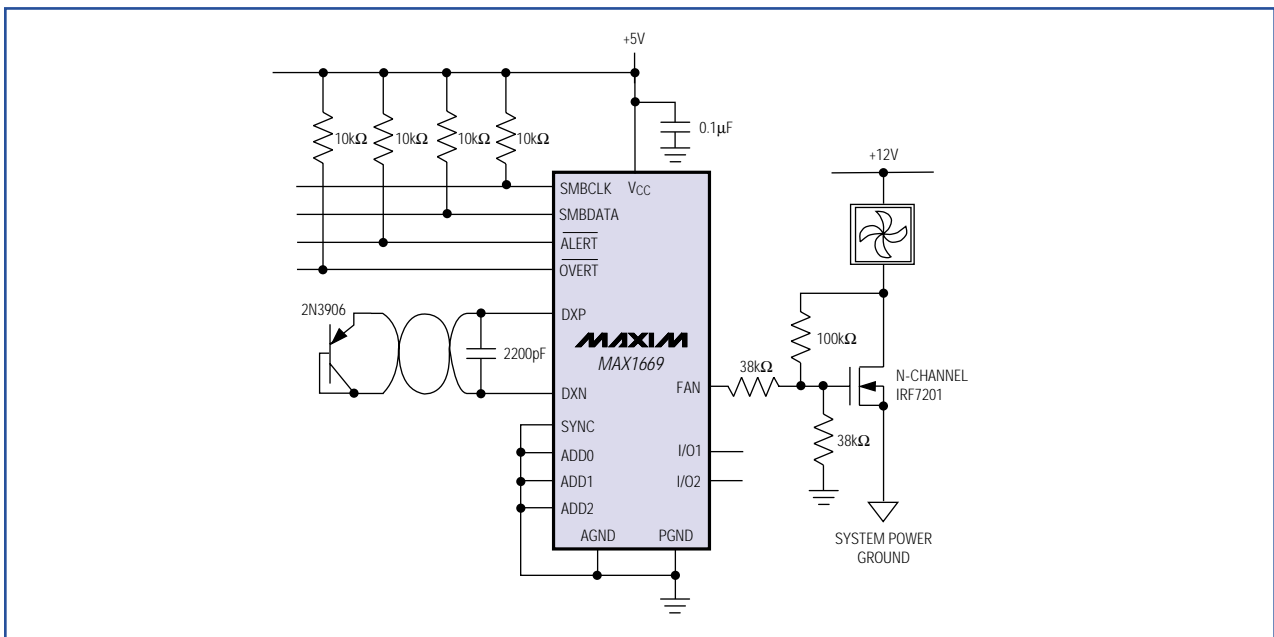


図1. MAX1669を使用したファンのリニア制御用標準回路

作業が必要になります。ファン制御アンプのフィードバックループには微分と積分の両方が含まれており、その時間定数は所与のファンの種類と型に応じて経験的に見つけ出さなければなりません(いったん見つければ製造用に固定することができます)。又、回路追加により部品点数が多くなるのでコスト増になります。

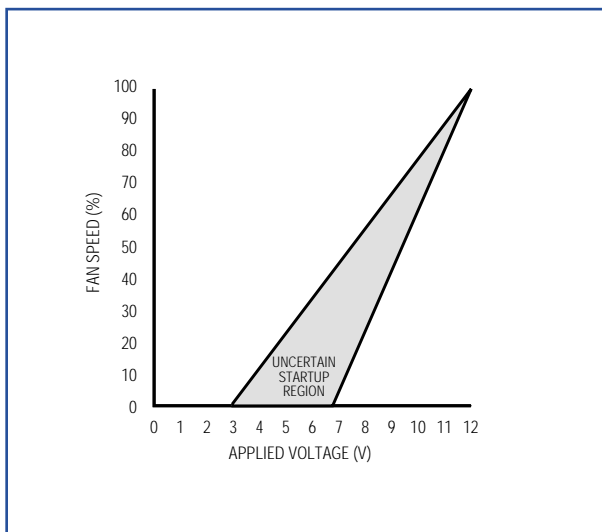


図2. 標準ファン速度対印加電圧

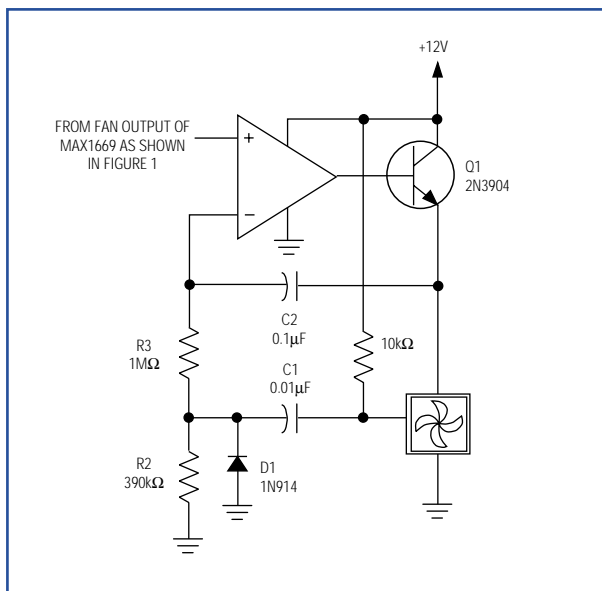


図3. このファン制御アンプをMAX1669と共に使用すると、タコメータ出力を備えたファンのフルレンジのリニアレギュレーション(ファン制御ではなく)を実現できます。

ファン速度のレギュレーション

マキシム社の最新ファンコントローラMAX6650及びMAX6651は、ファン速度のレギュレーションを可能にします。MAX6650/MAX6651はファンのタコメータ出力からのフィードバックを受け付けるように設計されたクローズループのファン制御ICです。これによりファン速度の設定が簡明になり、ユーザはスタートアップや低速時の信頼性について心配しなくてもよくなります。MAX6650/MAX6651はいずれも複数のファンを駆動できます(タコメータフィードバックはこのうちの一つだけからです)。又、MAX6651は最大4つのファンからのタコメータ信号を監視することができます。

図4に、MAX6650の標準アプリケーションを示します。MAX6650は、ファン速度レジスタに設定されたコードに基づき、ファンのタコメータ期間をスケールされたレジスタ値に強制的に一致させることによりファン速度を安定化します。標準的なファンは1回転で2つのタコメータパルスを生成するため、必要なファン速度レジスタ値は次式で計算されます。

$$t_{TACH} = \frac{1}{2 \times \text{ファン速度}} \quad (1)$$

$$K_{TACH} = \left[t_{TACH} \times K_{SCALE} \times \left(\frac{f_{CLK}}{128} \right) \right] - 1 \quad (2)$$

ここで、

K_{TACH} = ファン速度レジスタに設定された値

t_{TACH} = タコメータ信号の期間

K_{SCALE} = プリスケール値 (MAX6650/MAX6651のコンフィギュレーションレジスタに設定されており、値は1~16、デフォルトは4です。)

f_{CLK} = MAX6650/MAX6651のクロック周波数 (254kHz typ)

MAX6650/MAX6651のその他の利点

MAX6650/MAX6651は、ファン速度の安定化の他にも多くの機能を備えています。例えば、制御ループが安定化を行わないか又は設定されたウォッチドッグリミットをファン速度が超えている状態等を検出するウォッチドッグ機能や、その他の汎用デジタル機能があります。しかしこれらの機能は本稿の範囲を超えているため、詳細はMAX6650/MAX6651データシートを参照して下さい。

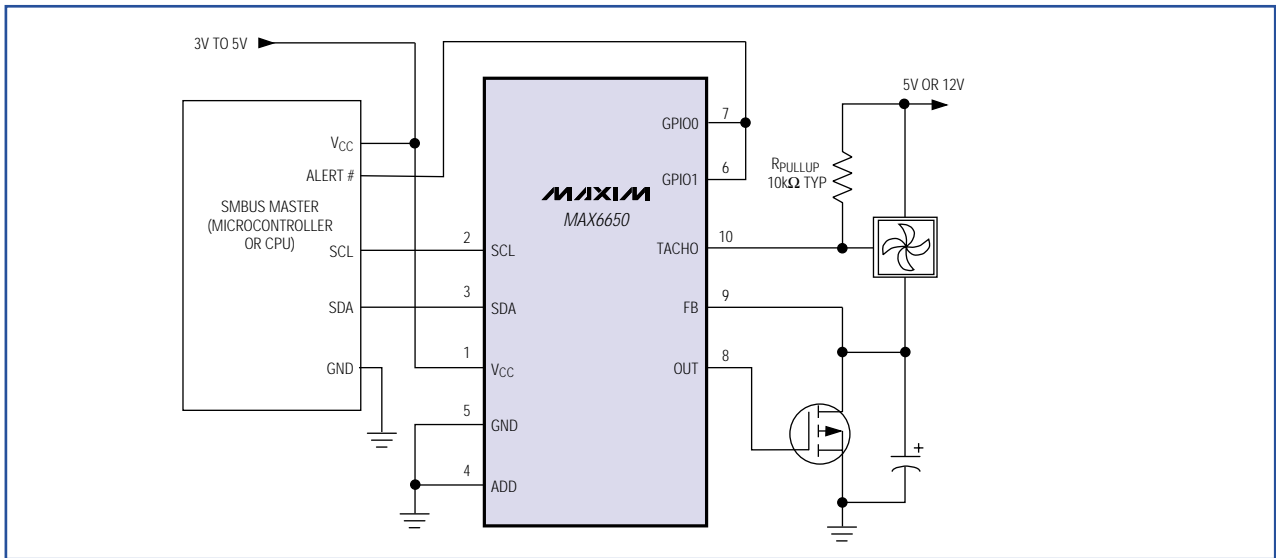


図4. MAX6650は、フィードバックループの中にタコメータ出力ファンを含む真のファン速度レギュレータです。

MAX6650/MAX6651により、設計者は前述のクローズループアンプに伴う複雑な問題から解放され、パストランジスタの選択と取付だけを行えばよいことになります。従来、抵抗性負荷、リニア、コントロールパス素子は、負荷の両端に全電源電圧の半分を供給する時に電力消費が最大になります。しかし、抵抗性負荷はファンのモデルとして好適でしょうか？ファンを抵抗性負荷としてモデル化すると、現実のファンの挙動より条件が厳しくなります。即ち、設計に慎重を期してファンを抵抗性負荷として扱った場合に生じる唯一の問題は、ファン特性を正確に測定した場合よりも大きい目のパストランジスタを使用することです。

12Vで240mAの電流を消費するファンを例にとってみましょう。等価抵抗は半電圧出力点で120mAを消費し、電力消費は0.72Wになります。

$$P_D = 0.5V \times 0.5I = 6 \times 0.12 = 0.72W \quad (3)$$

ここで、PDは最悪のケースの抵抗性負荷電力消費です。次式が満たされれば、特別なヒートシンクは必要ありません。

$$\frac{T_{JMAX} - T_A}{P_D} \leq \theta_{J-A} \quad (4)$$

ここで、

T_{JMAX} = トランジスタメーカーのデータシートに記載されている最大許容ジャンクション温度

T_A = 最大予想周囲温度

P_D = 電力消費(上の式3と同じ)

θ_{J-A} = ジャンクションと周囲間の熱抵抗(トランジスタメーカーのデータシートより)

この式が成り立たない場合は、次式を満たすヒートシンクが必要です。

$$R_{\theta SA} \leq \frac{T_{JMAX} - T_A}{P_D} - R_{\theta JC} \quad (5)$$

ここで、

$R_{\theta SA}$ = ヒートシンクの熱抵抗(ヒートシンクメーカーのデータシートより)

$R_{\theta JC}$ = パストランジスタのジャンクションからケースへの熱抵抗(パストランジスタメーカーのデータシートより)

ファンの両端で完全短絡が起こる可能性についてはまだ論じていませんが、完全短絡が起きた場合は、ファン電源から得られるフル電流がパストランジスタに流れます。この状態が発生する可能性がある場合は、その電流と電圧値を使って全ての電力消費及びヒートシンクの計算をする必要があります。別方法としては、図5に示すような電流制限回路をパストランジスタに含めることができます。電流制限抵抗の値は次式で計算して下さい。

$$R_{OUT} = \frac{0.6}{I_{LIMIT}} \quad (6)$$

ここで、 I_{LIMIT} は所望の電流リミット値です。この電流制限回路は温度に敏感であることに注意して下さい。この式の0.6という項は実際には電流リミットトランジスタのベース・エミッタ電圧であるため、 -2.2mV/ で変化します。これは、温度上昇につれて電流リミットを減少させるので有用かもしれません。

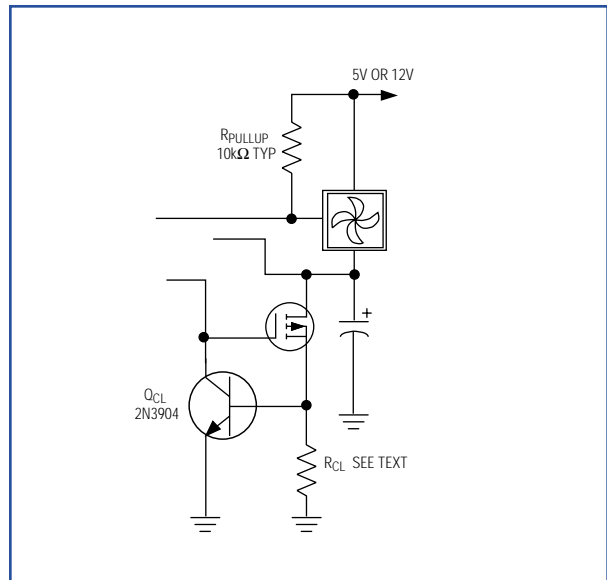


図5. この回路は、バスタランジスタに電流制限が必要な時にその機能を提供します。