AVR446：ｽﾃｯﾋﾟﾝｸﾞ ﾓｰﾀの直線状速度制御

要点

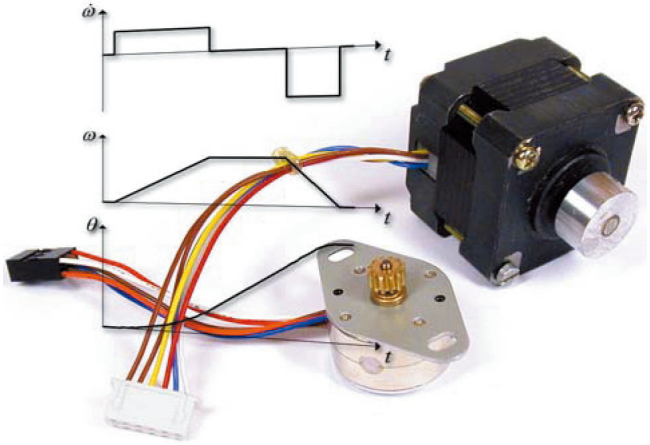
- ■ ｽﾃｯﾋﾟﾝｸﾞ ﾓｰﾀの直線状速度制御
  - ・ 加速、減速、最大速度、移動段数の制御
- ■ 1つのﾀｲﾏ/ｶｳﾝﾀ割り込みによる駆動
- ■ 全段階または半段階での段(ｽﾃｯﾌﾟ)駆動動作
- ■ 16ﾋﾞｯﾄ ﾀｲﾏ/ｶｳﾝﾀを持つ全てのAVR®ﾃﾞﾊﾞｲｽを支援
- ■ 19200bps、8ﾋﾞｯﾄ ﾃﾞｰﾀ、ﾊﾟﾘﾃｨなし、1停止ﾋﾞｯﾄのｼﾘｱﾙ ｲﾝﾀｰﾌｪｰｽを持つ、3.68MHzで走行するATmega48用実演応用

1. 序説

この応用記述はｽﾃｯﾋﾟﾝｸﾞ ﾓｰﾀに対する正確な直線状速度制御器の実装方法を記述します。ｽﾃｯﾋﾟﾝｸﾞ ﾓｰﾀはﾃﾞｼﾞﾀﾙ ﾊﾟﾙｽを機械的な軸回転に変換する電磁装置です。ﾌﾞﾗｼまたは接触子が全く存在せず、安価、高信頼性、低速での高ﾄﾙｸ、そして高い移動精度のため、この種のﾓｰﾀを使用してより高い簡潔性のような多くの利点が達成されます。ｽﾃｯﾋﾟﾝｸﾞ ﾓｰﾀを持つ多くのｼｽﾃﾑは速度変更時の加減速制御を必要とします。この応用記述は位置と速度だけでなく加減速を制御する能力がある実演応用を持つﾄﾞﾗｲﾊﾞを提供します。

この直線状速度制御器はD. Austinによる'実時間でのｽﾃｯﾋﾟﾝｸﾞ ﾓｰﾀ速度特性生成(Generate stepper-motor speed profles in real time)'、2005年1月の'組み込みｼｽﾃﾑのﾌﾟﾛｸﾞﾗﾐﾝｸﾞ(Embeded Systrems Programming)'に存在する算法に基きます。この算法はﾃﾞｰﾀ表なしで簡単な固定小数点演算だけを使用して、実時間でのﾊﾟﾗﾒｰﾀ化と計算を許します。

図1-1. ｽﾃｯﾋﾟﾝｸﾞ ﾓｰﾀ



2. 原理

2.1. ｽﾃｯﾋﾟﾝｸﾞ ﾓｰﾀ

この応用記述は制御器自体の実現だけでなく直線状勾配ﾓｰﾀ速度制御についての原理も網羅します。読者は基本的なｽﾃｯﾋﾟﾝｸﾞ ﾓｰﾀ動作を熟知しているけれども、殆どの関連する話題の大筋が与えられていないとの仮定です。ｽﾃｯﾋﾟﾝｸﾞ ﾓｰﾀについての更なる詳細はD. W. Jonesの'ｽﾃｯﾋﾟﾝｸﾞ ﾓｰﾀの制御(Control of Stepper Motors)''で得られます。

8ﾋﾞｯﾄ AVR®
マイクロ コントローラ

応用記述

Rev. 8017A-06/06, 8017AJ1-01/14

2.1.1. 双極対単極のｽﾃｯﾋﾟﾝｸﾞ ﾓｰﾀ

ｽﾃｯﾋﾟﾝｸﾞ ﾓｰﾀの一般的な2つの形式は双極ﾓｰﾀと単極ﾓｰﾀです。双極と単極のﾓｰﾀは図2-1.で示されるように双極が各巻き線上に中点引き出し口を持つことを除いて同じです。

単極ﾓｰﾀは巻き線を通して双方向に駆動されるべき電流を必要とし、図2-2.で示されるような全ﾌﾞﾘｯｼﾞ駆動部が必要とされます。双極ﾓｰﾀの中点引き出し口は図2-2.でも示される、より簡単な駆動開路を許し、電流の流れを1方向に限定します。双極ﾓｰﾀの主な欠点は常に全ての巻き線を励磁するための制限された能力で、単極ﾓｰﾀに比べてより低いﾄﾙｸに帰着します。双極ｽﾃｯﾋﾟﾝｸﾞ ﾓｰﾀは中点引き出し口を未接続にすることによって単極ﾓｰﾀとして使用することができます。

図2-1. 双極と単極のｽﾃｯﾋﾟﾝｸﾞ ﾓｰﾀ

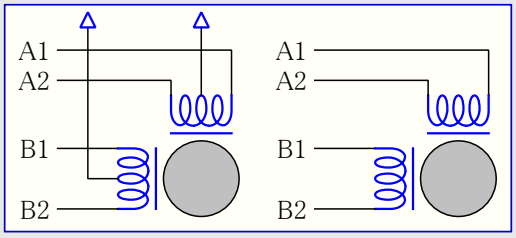
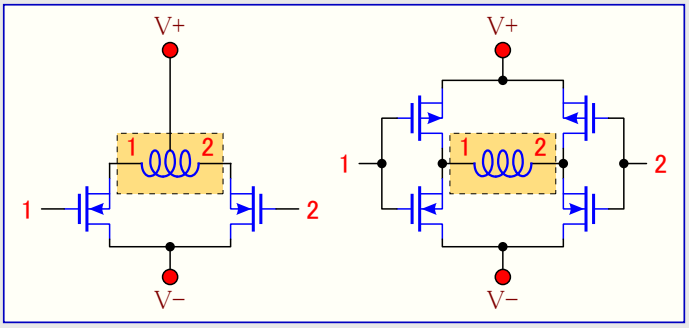


図2-2. MOSﾄﾗﾝｼﾞｽﾀでの双極と単極の駆動



2.1.2. 1相、2相、1-2相駆動 (訳注:内容加筆)

1相駆動で使用されるｽﾃｯﾋﾟﾝｸﾞ ﾓｰﾀは一度に1つの巻き線を励磁します。このようにして表2-1.の1相駆動列で示されるように4つの異なる設定(位置)が可能です。両方の巻き線を同時に励磁することにより、ｽﾃｯﾋﾟﾝｸﾞ ﾓｰﾀは1相駆動時に得られる位置間で止まり、これは2相駆動としても知られます。更に1相駆動と2相駆動の全てを行うと、半段階毎の進行となり、これは1-2相駆動として知られます。これは表2-1.の1-2相駆動列で示されるように8つの位置を与えます。両巻き線励磁時、ﾄﾙｸは1巻き線だけの励磁時よりも概ね1.4倍高くなりますが、2倍の電力消費でです。表2-1.の電気的周回部は電気的な1周の全てに於ける部分です。機械的な1周(回転)は一般的に電気的な多数周回から成ります。

表2-1. 1相駆動、2相駆動、1-2相駆動

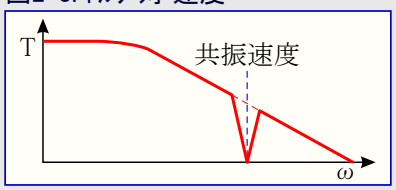
電気的極性	A巻き線	+	+	OFF	−	−	−	OFF	+
	B巻き線	OFF	+	+	+	OFF	−	−	−
電気的周回部分	1相駆動	1	−	2	−	3	−	4	−
	2相駆動	−	1	−	2	−	3	−	4
	1-2相駆動	1	2	3	4	5	6	7	8

2.1.3. 速度特性

ｽﾃｯﾋﾟﾝｸﾞ ﾓｰﾀでの1つの欠点はｽﾃｯﾋﾟﾝｸﾞ ﾓｰﾀのﾄﾙｸが速度増加と共に減少するための高速での制限されたﾄﾙｸ能力です。図2-3.で示されるようにﾄﾙｸは共振速度でも落ち込みます。共振速度はｽﾃｯﾋﾟﾝｸﾞ ﾓｰﾀの駆動の仕方と負荷に依存します。

最大ﾄﾙｸは低速で達成され、これは多くの応用に於いて有利です。

図2-3. ﾄﾙｸ 対 速度



2.2. ｽﾃｯﾋﾟﾝｸﾞ ﾓｰﾀ基本均分法

ｽﾃｯﾋﾟﾝｸﾞ ﾓｰﾀで回転運動を生成するために、巻き線を通る電流が正しい順序で変更されなければなりません。これは(ｽﾃｯﾋﾟﾝｸﾞ ﾓｰﾀ)ﾊﾟﾙｽと方向信号に従う時に正しい出力手順を与えるﾄﾞﾗｲﾊﾞを使用することで達成されます。

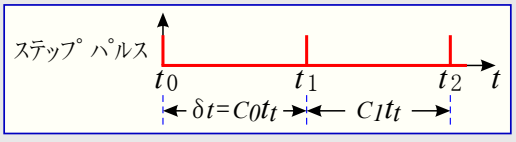
一定速度でｽﾃｯﾋﾟﾝｸﾞ ﾓｰﾀを回転するため、ﾊﾟﾙｽは図2-4.で示される安定した頻度で生成されなければなりません。

計数器が周波数ft[Hz]で走行する、これらのﾊﾟﾙｽを生成します。

計数器Cによって設定される遅延間隔δtは以下です。

$$\delta t = Ct_t = \frac{C}{ft} \text{ [s]}$$

図2-4. ｽﾃｯﾋﾟﾝｸﾞ ﾓｰﾀ ﾊﾟﾙｽ



ﾓｰﾀ段階角 α 、位置 θ 、速度 ω は次式で与えられます。

$$\alpha = \frac{2\pi}{spr} \text{ [rad]} \qquad \theta = n\alpha \text{ [rad]} \qquad \omega = \frac{\alpha}{\delta t} \text{ [rad/sec]}$$

ここでのsprは1周当たりの段階(ｽﾃｯﾌﾟ)数、nはｽﾃｯﾌﾟ数、そして1rad/secは9.55rpmです。

2.3. 直線状速度勾配

滑らかな方法でステッピング モータを始動そして停止するため、加速と減速の制御が必要とされます。図2-5.は加減速、速度、位置の関連を示します。一定の加減速を用いることが直線状の速度特性を与えます。

ステッピング モータ パルス間の遅延時間(間隔)δtが速度を制御します。これらの遅延時間はステッピング モータの速度が可能な限り速度勾配近くに沿うように計算されなければなりません。

ステッピング モータ運動の離散段階制御とそれらの段階間での遅延時間の分解能はタイマ/カウンタの周波数によって与えられます。

図2-5. 加減速($\dot{\omega}$)、速度(ω)、位置(θ)

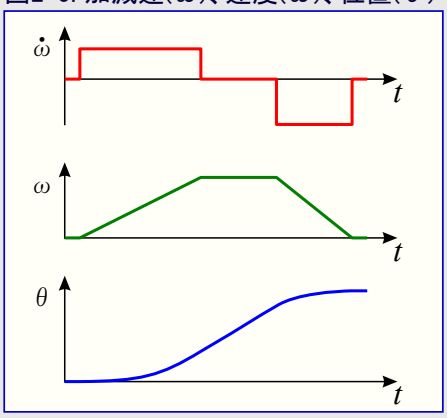
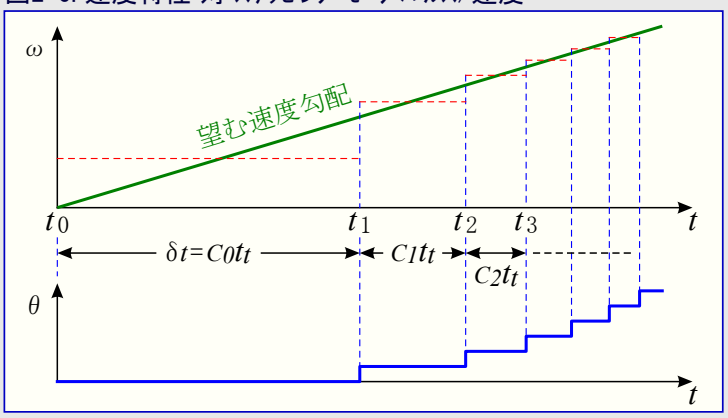


図2-6. 速度特性 対 ステッピング モータ パルス/速度



2.3.1. 段階相互間遅延の正確な計算

後続するする計数器遅延C_nだけでなく最初の計数器遅延C_0も以下によって与えられます(詳細については追補をご覧ください)。

$$C_0 = \frac{1}{t_t}\sqrt{\frac{2\alpha}{\dot{\omega}}} \qquad C_n = C_0\left(\sqrt{n+1} - \sqrt{n}\right)$$

マイクロ コントローラの計算能力は限定されており、2つの平方根計算が時間を消費します。従って複雑な計算なしでの近似が考慮されます。

段階相互間遅延(間隔)に対してテイラー級数近似を用いた時間nでの計数器値は以下によって与えられます(詳細については追補をご覧ください)。

$$C_n = C_{n-1} - \frac{2C_{n-1}}{4n+1}$$

この計算は2つの平方根よりもずっと高速ですが、n=1で0.44の誤差を生じます。この誤差を補償するための方法は0.676でC_0を乗算することによります。

2.3.2. 加速での変更

追補で示されるように、加速はC_0とnによって与えられます。加速(または減速)で変更が行われる場合、新しいnが計算されなければなりません。

モータの加速、速度、段階角としての時間t_nとnは以下によって与えられます。

$$t_n = \frac{\omega_n}{\dot{\omega}} \qquad n = \frac{\dot{\omega} t_n^2}{2\alpha}$$

これらの式の合成がこの関連を与えます。

$$n\dot{\omega} = \frac{\omega^2}{2\alpha}$$

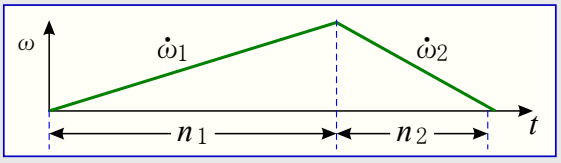
これは与えられた速度に達するのに必要とする段階(ステップ)数が加速に逆比例($n_1\dot{\omega}_1 = n_2\dot{\omega}_2$)することを示します。

これは$\dot{\omega}_1$から$\dot{\omega}_2$への加速の変更がnによって行われることを意味します。これは図2-7.で示されます。

与えられた段階数の移動に於ける減速は速度=0で終わるように正しい段階丁度で始めなければなりません。n_1を得るのに次式が用いられます。

$$n_1 = \frac{(n_2 + n_1)\dot{\omega}_2}{(\dot{\omega}_1 + \dot{\omega}_2)}$$

図2-7. 速度昇降勾配



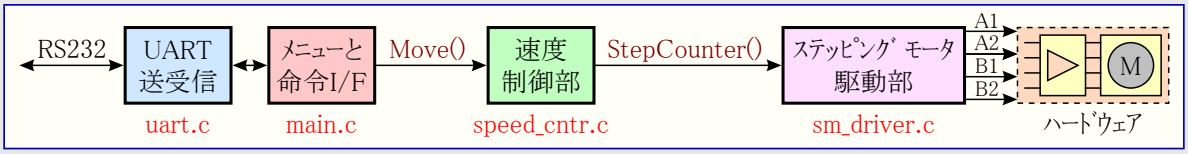
3. 実装

この応用記述と共にCで書かれた動作実装が含まれます。ソース コードの完全な資料とコンパイル情報はソース コードと共に含まれる'readme.html'ファイルを開くことによって得られます。

実演応用はステッピング モータの直線状速度制御を実演します。使用者はシリアル ポートを使用して各種命令を発行することによってステッピング モータ速度特性を制御することができ、そしてAVRは接続されたステッピング モータをそれによって駆動します。

実演応用は図3-1.の構成図で示されるように、3つの主な部分に分けられます。各部分に対して1つのファイルがあり、主ルーチンによって使用されるUARTルーチン用のファイルもあります。

図3-1. 実演応用の構成図



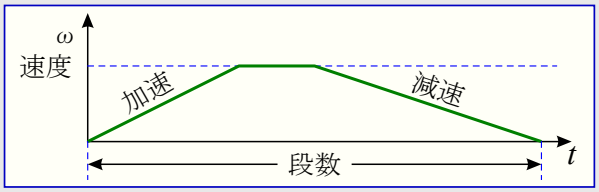
main.cはメニューと命令インターフェースを持ち、シリアル(通信)線に接続された端末により、使用者にステッピング モータの制御を与えます。

speed_cntr.cはステッピング モータを望む速度特性に従わせるのに必要なデータを計算して段階(ステップ)パルスを生成します。

sm_driver.cはステッピング モータを制御するために、段数を計数して正しい信号を出力します。

ステッピング モータを制御するために速度特性を記述する4つの項目が必要とされます。速度特性は速度0で始まり、与えられた速度まで加速します。この速度は減速が始まるまで一定に保たれます。最後に与えられた段数で速度0へ減速します。速度特性は図3-2.で示されます。

図3-2. 速度特性



速度特性を記述する項目は以下です。

段数(step)	: 移動する段階(ステップ)数
加速(accel)	: 使用する加速度
減速(decel)	: 使用する減速度
速度(speed)	: 使用する(最大)速度

3.1. メニューと命令インターフェース

実演応用を使用するのに、使用者はAVRのシリアル ポートに端末を接続しなければなりません。UART設定は19200bps、8ビット データ、パリティなし、1停止ビットです。どの端末偽装プログラムでも動作するでしょう。端末を使用して使用者はステッピング モータを制御するための各種命令を与え、そして実演応用から戻された情報を得ることができます。

(uart.cで見つかる)UART受信割り込みルーチンは受信した文字を受信緩衝部に格納し、そして後退(BS:Back Space)を処理します。[Enter](ASCII符号$0D(13))受信時、主ルーチンは緩衝部を読んで与えられた命令を実行します。

開始時と'?'命令で、以下のヘルプ画面が示されます。

```
------------------------------------------------------------
Atmel AVR446 - Linear speed control of stepper motor

?        - Show help
a [data] - Set acceleration (range: 71 - 32000)
d [data] - Set deceleration (range: 71 - 32000)
s [data] - Set speed (range: 12 - motor limit)
m [data] - Move [data] steps (range: -64000 - 64000)
move [steps] [accel] [decel] [speed]
         - Move with all parameters given
<enter>  - Repeat last move

    acc/dec data given in 0.01*rad/sec^2 (100 = 1 rad/sec^2)
    speed data given in 0.01*rad/sec (100 = 1 rad/sec)
------------------------------------------------------------
```

```
------------------------------------------------------
ATMEL AVR446 - ステッピング モータの直線状速度制御

?        - ヘルプ表示
a [data] - 加速設定(範囲:71～32000)
d [data] - 減速設定(範囲:71～32000)
s [data] - 速度設定(範囲:12～モータ限界)
m [data] - [data]段移動(範囲:-64000～64000)
move [段数] [加速] [減速] [速度]
         - 与えられた全項目で移動
<enter>  - 最後の移動繰り返し

    加減速値は0.01×rad/sec^2(100=1rad/sec^2)で与えます。
    速度値は0.01×rad/sec(100=1rad/sec)で与えます。
------------------------------------------------------
```

メニュー表示または命令実行後、次のような情報行が示されます。

```
Motor pos: 0     a:4000  d:4000  s:2000  m:400
```

実演応用は移動段(ステップ)数は勿論、現在のモータ位置、加速度、減速度、速度の設定も伝えます。

ステッピング モータを移動するには3つの異なる方法があります。

- <Enter>押下
  ステッピング モータは応用によって与えられた設定で指定されるように動きます。
- m [data]
  ステッピング モータは与えられた設定で[data]段(ステップ)移動します。
- move [段数][加速][減速][速度]
  応用は[加速][減速][速度]設定で[data]段(ステップ)移動します。

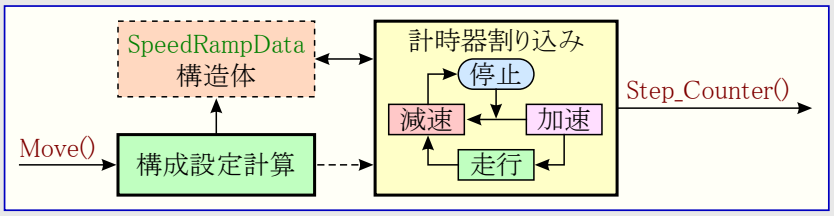
ステッピング モータが走行を開始すると、'Running...'が表示されます。モータが動いている限り、新しい命令は妨げられます。停止後に'OK'が表示され、新しい命令が受け入れられます。

3.2. 速度制御部

速度制御部は計算して速度特性を生成します。速度制御部に関する構成図は図3-3.で示されます。ステッピング モータを動かすため、Move()関数を呼ぶことによって速度制御部が構成設定されます。

Move()関数は最初に必要とする全ての項目を計算して速度勾配データ構造体にそれらを格納し、そして計時器割り込みを許可します。計時器は望んだ速度勾配に従って割り込みを生成し、ステッピング モータを移動するために各割り込みでStep_Couner()関数を呼びます。

図3-3. 速度制御部の構成図



3.2.1. 構成設定計算

実演応用では速度特性に関する項目が毎回の命令毎に計算され、ステッピング モータ移動開始に持ち込むまでに呼び出しからの小さな遅れを生じます。実際の応用に於いて、速度特性での限定された変更だけが必要とされる場合に、これは不要かもしれません。その場合、各項目を先行して計算し、構成設定計算を飛ばすことができます。

コードを高速にするために浮動小数点演算は避けられ、従って精度を保つための変数の尺度調節が重要です。演算を簡単にするために予め計算されたコンパイラ定数も使用され、これはsm_driver.hヘッダ ファイルで見ることができます。

速度は以下で得ます。

$$A\_T\_x100 = \alpha f_t \times 100 \qquad min\_delay = C = \frac{A\_T\_x100}{speed}$$

加速は以下で得ます。

$$T1\_FREQ\_148 = 0.676 \times f_t \div 100 \qquad A\_SQ = 2\alpha \times 10000000000 \qquad step\_delay = C_0 = T1\_FREQ\_148 \sqrt{\frac{A\_SQ}{accel}} \div 100$$

速度特性計算に関して2つの異なる筋書きがあります。

1. 望む速度に達するまで加速継続、または
2. 望む速度に達する前に減速開始

筋書きは速度特性を記述する4つ全ての変数に依存します。

3.2.1.1. 希望速度到達まで加速継続

図3-4.で減速開始前に望む速度に達する速度勾配が示されます。

- max_s_limは望む速度へ加速するのに必要とする段(ステップ)数です。

$$max\_s\_lim = n = \frac{speed^2}{2\alpha \times accel \times 100}$$

- accel_limは(望む速度と無関係に)減速を開始する前の段(ステップ)数です。

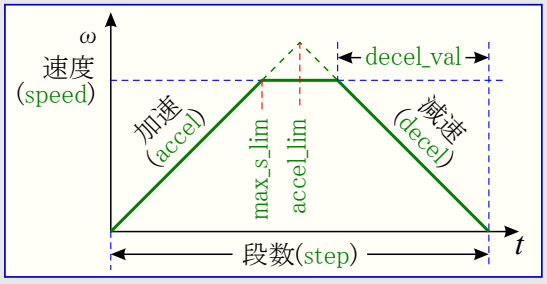
$$accel\_lim = n_1 = \frac{step \times decel}{accel + decel}$$

max_s_lim<accel_limなら、加速は望む速度到達によって制限されます。

減速はこれに依存し、この場合のdecel_valは以下で得られます。

$$decel\_val = -max\_s\_lim \times \frac{accel}{decel}$$

図3-4. 望む速度値によって制限される速度勾配



3.2.1.2. 希望速度到達前に減速開始

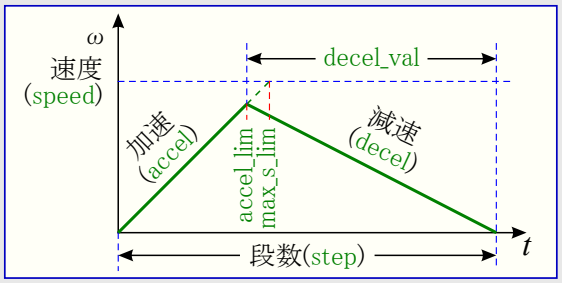
図3-5.で望む速度へ達する前に減速を始めなければならない速度勾配が示されます。

max_s_lim>accel_limなら、加速は減速開始によって制限され、そしてdecel_valは以下で得られます。

$$decel\_val = -(step - accel\_lim)$$

(訳補) 以降で記述されていますが、減速時の減速段数計数器は負数で始まり、0で終了となる上昇計数器です。従って減速開始時のこの計数器の初期値であるdecel_valは負(−)の段数として計算されます。

図3-5. 望む速度到達前に減速が始まる速度勾配



3.2.2. 計時器割り込み

計時器割り込みは'段(ステップ)パルス'を生成し(StepCounter()関数呼び出し)、そしてこれはステッピング モータが移動する時にだけ走行します。計時器割り込みは図3-6.で示されるように、速度特性に従って4つの異なる状態で動作します。

この動きは図3-7.で示される計時器割り込みに於ける状態機構で実現されます。

図3-6. 各種速度特性部分に対する動作状態

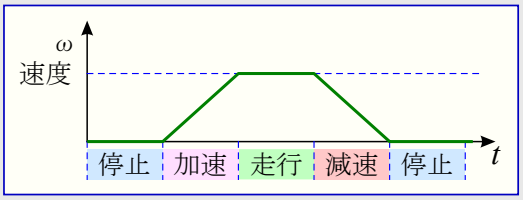
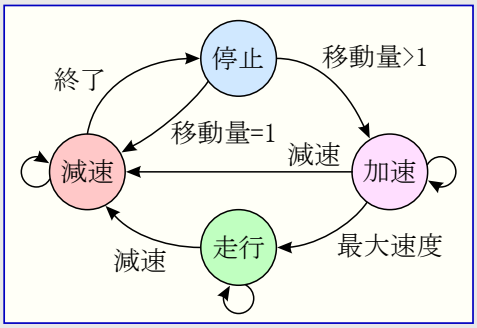


図3-7. 計時器割り込み用状態機構



応用開始時、またはステッピング モータ停止時、状態機構は停止状態に留まります。構成設定計算が行われると、新しい状態が設定され、計時器割り込みが許可されます。1段(ステップ)よりも多く移動するとき、状態機構は加速へ行きます。1段(ステップ)だけの移動なら、状態は減速に変更されます。

状態が加速に変更されると、応用は、
- 望む速度に達して状態が走行に変更されるか、または
- 状態を減速に変更して減速を始めなければならなくなるか

のどちらかまで、ステッピング モータを加速します。

状態が走行に設定されると、減速を開始しなければならなくなるまで、ステッピング モータは一定速度に保たれ、その後に状態が減速に変更されます。

そして減速に留まり、望む段(ステップ)数で速度が0に達するまで減速します。その後に状態は停止に変更されます。

3.2.2.1. 計算と計時器

加減速中の各段階(ステップ)に対しては新しい遅延時間が計算されなければなりません。この計算は剰余を生じる除算を含み、精度を改善するためにこの剰余が保持され、次の式に含められます。

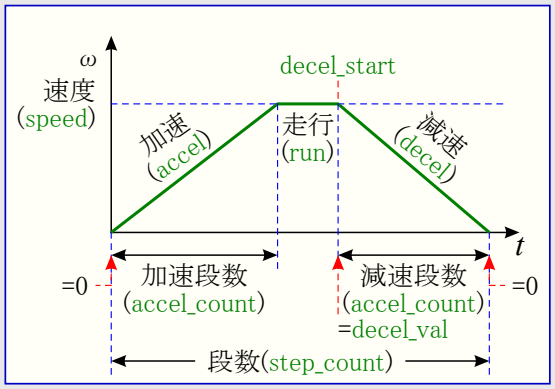
$$新step\_delay = step\_delay - \frac{2 \times step\_delay + reset}{4 \times accel\_count + 1} \qquad rest：端数分$$

$$新rest = (2 \times step\_delay + reset) \times (前式剰余(4 \times accel\_count + 1))$$

位置の経緯を保つのと状態を変更する時のためにいくつかの計数器変数が必要とされます。図3-8.ではこれらの使い方が図解されます。

- step_countは段数を数え、加速開始時に0で始まり、減速終了後に指示された段数と同じ値を持ちます。
- accel_countは加減速を制御するのに使用されます。加速では0から始まり、加速終了まで各段毎に増やされます。減速開始時に負(−)のdecel_valに設定され、各段毎に増やされます。0到達時に移動が終了され、状態を停止に設定します。
- decel_startは減速開始を知らせます。step_countがdecel_startと等しい時に状態が減速に設定されます。

図3-8. 遅延時間に於ける計数器変数

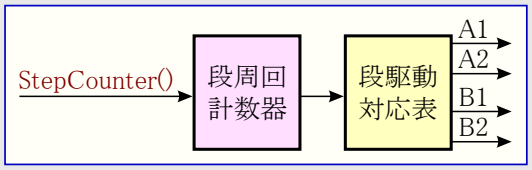


3.3. ステッピング モータ駆動部

ステッピング モータ駆動部は指示された方向でステッピング モータを移動するための正しい手順の信号を生成します。ステッピング モータ駆動部に関する構成図は図3-9.で示されます。

段周回計数器はStep_Counter()関数が呼ばれる時毎に増加または減少(±1)します。全段間隔(1相駆動)使用時はこの計数器値が0～3になり、半段間隔(1-2相駆動)使用時はこの値が0～7になります。この値はステッピング モータに於ける電気的な1周回での各位置と等価です。段周回計数器値は段参照表に対する指標として用いられ、ステッピング モータ駆動部へ正しい信号が与えられます。

図3-9. ステッピング モータ駆動部構成図



3.4. コード量と速度

完全な実演コードは4Kバイトのコード メモリを使用し、速度制御部とステッピング モータ駆動部がこの内の1.5Kバイトを使用します。構成設定計算を削除して予め計算された項目値を使用することが更にコード量を減らすでしょう。

Move()を呼ぶと、計時器割り込みが開始される前に構成設定計算が行われます。これはステッピング モータを始動させるための呼び出しから概ね1.5msの遅れを生じます。計時器割り込みは加減速中の計算を実行し、1回の計時器割り込みに於いて概ね200μsが費やされます。一定速度での走行時はより少ない時間が必要とされ、概ね35μsで充分です。そして最大出力速度は加減速計算によって制限されます。1回転当たり400段(ステップ、0.9°/ステップ)のステッピング モータに対する最大出力速度は以下です。

$$\omega_{max} = \frac{2\pi}{200[\mu s] \times 400} = 78.5[\text{rad/sec}] \ (=750[\text{rpm}])$$

このコードを他の応用に実装する時は他の割り込みに注意が祓われなければなりません。ステッピング モータに関する計時器割り込みが他の割り込み処理ルーチンによって妨げられた場合、これはステッピング モータ速度を変化させる(定加速ではなくする)でしょう。与えられた応用に於いて危機的になるかもしれないのではなく、可能な限りコードを頑強で(最終)決定的にすることは常に良いことです。

4. 参照文献

D. Austin著、2005年1月の「組み込みシステムのプログミング(Embedded Systems Programming)」内の記事、「実時間でのステッピング モータ速度特性の生成(Generate stepper-moter speed profiles in real time)」

http://www.embedded.com//showArticle.jhtml?articleID=56800129

D. W. Jones著、2001年McGraw-Hill出版、W. H. YeadonとA. W. Yeadon編集、「小電動機ハンドブック(Handbook of Small Electric Motors)」の5.2.10、10.8、10.9、10.10項のステッピング モータの制御

http://www.cs.uiowa.edu/~jones/step/

5. 追補

5.1. 計時器遅延

与えられた時間での速度は以下です。

$$\omega(t) = \int_{\tau=0}^{t} \dot{\omega}\, d\tau = \dot{\omega} t$$

位置は以下によって与えられます。

$$\theta(t) = \int_{\tau=0}^{t} \omega(\tau) d\tau = \frac{1}{2} = \dot{\omega} t^2 = n\alpha$$

軸角 $\theta = n\alpha$ に於ける第n段(ｽﾃｯﾌﾟ)ﾊﾟﾙｽは以下です。

$$t_n = \sqrt{\frac{2n\alpha}{\dot{\omega}}}$$

そして2つの段間遅延(間隔)時間は以下です。

$$C_n t_t = t_{n+1} - t_n = \sqrt{\frac{2\alpha}{\dot{\omega}}} \left(\sqrt{n+1} - \sqrt{n}\right)$$

最後に計数器遅延に関する式は以下で得られます。

$$C_n = \frac{1}{t_t} \sqrt{\frac{2\alpha}{\dot{\omega}}} \left(\sqrt{n+1} - \sqrt{n}\right)$$

これは最初と第n段(ｽﾃｯﾌﾟ)の計数器遅延に関する式を導きます。

$$C_0 = \frac{1}{t_t} \sqrt{\frac{2\alpha}{\dot{\omega}}} \qquad C_n = C_0\left(\sqrt{n+1} - \sqrt{n}\right)$$

5.2. 段相互間遅延

ﾃｲﾗｰ級数近似を用いた

$$\sqrt{1 \pm \frac{1}{n}} = 1 \pm \frac{1}{2n} - \frac{1}{8n^2} + O\left(\frac{1}{n^3}\right)$$

は以下を導きます。

$$\frac{C_n}{C_{n-1}} = \frac{C_0\left(\sqrt{n+1} - \sqrt{n}\right)}{C_0\left(\sqrt{n} - \sqrt{n+1}\right)} = \frac{1 + \frac{1}{2n} - \frac{1}{8n^2} + O\left(\frac{1}{n^3}\right) - 1}{1 - \left(1 - \frac{1}{2n} - \frac{1}{8n^2} + O\left(\frac{1}{n^3}\right)\right)} = \frac{4n-1}{4n+1}$$

最後に、計数器遅延に関する式は次のように近似することができます。

$$C_n = C_{n-1} - \frac{2C_{n-1}}{4n+1}$$

本社

Atmel Corporation
2325 Orchard Parkway
San Jose, CA 95131, USA
TEL 1(408) 441-0311
FAX 1(408) 487-2600

国外営業拠点

Atmel Asia
Unit 1-5 & 16, 19/F
BEA Tower, Millennium City 5
418 Kwun Tong Road
Kwun Tong, Kowloon
Hong Kong
TEL (852) 2245-6100
FAX (852) 2722-1369

Atmel Europe
Le Krebs
8, Rue Jean-Pierre Timbaud
BP 309
78054 Saint-Quentin-en-Yvelines
Cedex
France
TEL (33) 1-30-60-70-00
FAX (33) 1-30-60-71-11

Atmel Japan
104-0033 東京都中央区
新川1-24-8
東熱新川ﾋﾞﾙ 9F
ｱﾄﾒﾙ ｼﾞｬﾊﾟﾝ株式会社
TEL (81) 03-3523-3551
FAX (81) 03-3523-7581

製造拠点

Memory
2325 Orchard Parkway
San Jose, CA 95131, USA
TEL 1(408) 441-0311
FAX 1(408) 436-4314

Microcontrollers
2325 Orchard Parkway
San Jose, CA 95131, USA
TEL 1(408) 441-0311
FAX 1(408) 436-4314

La Chantrerie
BP 70602
44306 Nantes Cedex 3
France
TEL (33) 2-40-18-18-18
FAX (33) 2-40-18-19-60

ASIC/ASSP/Smart Cards
Zone Industrielle
13106 Rousset Cedex
France
TEL (33) 4-42-53-60-00
FAX (33) 4-42-53-60-01

1150 East Cheyenne Mtn. Blvd.
Colorado Springs, CO 80906, USA
TEL 1(719) 576-3300
FAX 1(719) 540-1759

Scottish Enterprise Technology Park
Maxwell Building
East Kilbride G75 0QR
Scotland
TEL (44) 1355-803-000
FAX (44) 1355-242-743

RF/Automotive
Theresienstrasse 2
Postfach 3535
74025 Heilbronn
Germany
TEL (49) 71-31-67-0
FAX (49) 71-31-67-2340

1150 East Cheyenne Mtn. Blvd.
Colorado Springs, CO 80906, USA
TEL 1(719) 576-3300
FAX 1(719) 540-1759

Biometrics
Avenue de Rochepleine
BP 123
38521 Saint-Egreve Cedex
France
TEL (33) 4-76-58-47-50
FAX (33) 4-76-58-47-60

文献請求
www.atmel.com/literature

本応用記述はATMELのAVR446応用記述(doc8017.pdf Rev.8017A-06/06)の翻訳日本語版です。日本語では不自然となる重複する形容表現は省略されている場合があります。日本語では難解となる表現は大幅に意訳されている部分もあります。必要に応じて一部加筆されています。頁割の変更により、原本より頁数が少なくなっています。

必要と思われる部分には( )内に英語表記や略称などを残す形で表記しています。

青字の部分はﾘﾝｸとなっています。一般的に赤字の0,1は論理0,1を表します。その他の赤字は重要な部分を表します。