

高等学校 工業

LEGO MINDSTORMS NXT を用いた二輪型倒立ロボット制御の教材開発

青森県立八戸工業高等学校 実習教諭 田中 昌勝

要 旨

本研究では、組込み型システムの中の、「NXT 走行体」という自律型ライントレース・ロボットを用いて、実際に組込みソフトウェアの研究をする。倒立振子とライントレースの制御システムを ICT 実習教材としてまとめ、即活用できる教材開発を行った。

キーワード : LEGO MINDSTORMS NXT 教材開発 問題解決能力

I 主題設定の理由

近年、組込みソフトウェア開発分野における若年層・初級エンジニアの育成が重要になってきている。この状況に対して、LEGO MINDSTORMS NXT は多くの教育機関で使用されており、特に、NXT は制御システムやロボット工学の利用が期待されている。組込みソフトウェア開発分野におけるエンジニアを目指す生徒が、ものづくりの楽しさの経験を通してソフトウェア開発が行える実践的な学習教材の開発を目指した。

II 研究目標

本研究では、NXT 走行体という自律型ライントレース・ロボットを用いて、実際に組込みソフトウェアの開発を行った。倒立振子とライントレースの制御システムを開発し、学習教材を ICT 教材としてまとめ、実習教材として即活用できる教材づくりを目的とした。

III 研究の実際とその考察

1 NXT の特徴

LEGO社がMIT（マサチューセッツ工科大学）と共同開発して1998年に発表したMINDSTORMSは、一般愛好家やロボット教育の世界に旋風を巻き起こした。この製品のコンセプトの核となるのは、ロボットの心臓部であるRCX やNXT と呼ばれるマイクロプロセッサが組み込まれたインテリジェントブロックである。これは、オフラインPCでプログラミング（フローチャートを書くときのように、機能を持ったアイコンを配置していく）し、USB 等のインターフェースを介してインテリジェントブロック内のメモリーに書き込むことで実行できる小規模な組込みシステムである。

このインテリジェントブロックと組み合わせることのできるブロックとして、アクチュエータの一つであるモータや各種センサ、モータの主軸になるものなど、数百種類のパーツがある。なお、NXT には、4つのセンサと3つのモータのブロックを同時に組み合わせることができる。また、ユーザーインターフェースとして液晶画面と4つの操作ボタンとスピーカを搭載している。

2 NXT 走行体

図1に示すNXT 走行体は、NXT を用いた倒立振子型ライントレーサであり、使用したアクチュエータは2個のモータである。また、使用したセンサは3種類で、各1個の光センサ、タッチセンサ、ジャイロセンサである。光センサは、白い床面に描かれている黒色のラインに反射し、そこから反射光を受光し、その強度を検出するものである。タッチセンサは、制御動作開始のためのスタートスイッチの役目を担っている。

ジャイロセンサは、左右の車輪（合計2輪）が接地して不安定な倒立した姿勢になっているNXT の、単位時間当たりに傾いている角速度を検出するものであり、この信号を基にして振子（NXT 本体部分）の倒立制

御を行う。

3 NXT 走行体の主たる構成要素

(1) マイクロプロセッサ (制御部)

マイクロプロセッサには、LEGO社製のインテリジェントブロック (RCX , NXT) を利用できるが、ここではRCXの後継機種であるNXT (図2) を使用することにした。

NXT は、National Instruments社のグラフィカルプログラミングソフトウェアであるLabVIEW をベースにしたプログラミング環境を備えている。これは、ハードウェアとの通信やデータ解析などを、コマンドアイコンを使ったROBOLAB での、プログラミングを可能にしている。NXT 本体内部には32ビットマイクロプロセッサが内蔵されており、アイコン操作を主体としたユーザーインターフェースを持っている。信号の入力ポート数は4，出力ポート数は3，Bluetooth 機能も搭載している。

今回は、オンラインPCにCygwin, Eclipse をインストールした開発環境で、C言語のプログラムを作成し、USB インターフェースを介して、NXT 本体に転送した。

(2) 光センサ

光センサには、LEGO MINDSTORMS NXT 付属の反射型を使用した。これは、発光素子と受光素子が同じ方向を向いた構造になっており (図3) ，発光素子からの光を対象物 (今回はコース床面) に当て、反射光を受光素子で受けて、その強度を検出する。例えば、白い面から反射してくる光は、黒い面から反射してくる光よりも強いので、その差で光センサが向いている面の白黒 (濃淡) を判断できることになる。

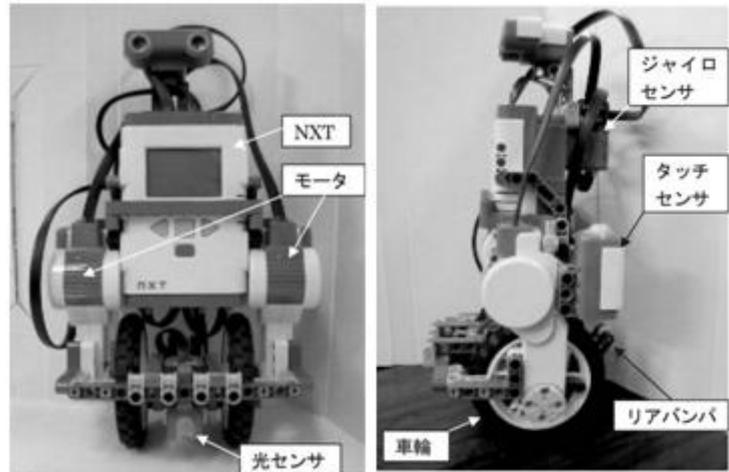
(3) ジャイロセンサ

ジャイロセンサは、回転速度 (角速度) を検出するものである。また、制御対象物の回転中心に配置する必要がないため、ビデオカメラやデジタルカメラの手ぶれ防止、ヒューマノイドロボットや模型用ヘリコプタの姿勢制御など、多くのものに利用されている。

本研究では、HiTechnic 社のジャイロセンサを使用した (図4) 。このセンサは、NXT 標準の接続ケーブルを使用してNXT の入力ポートと接続する。このことにより、単軸方向について毎秒300 回までの回転速度値を読み出すことができる。この検出信号を基にして、走行体の姿勢の維持を行う。

(4) タッチセンサ

タッチセンサ (図5) は、機械的な押しボタンスイッチであり、ボタンが押されることによって変化する信号を検出するものである。プログラム制御実行開始のためのスタートボタンとして、タッチセンサを利用した。また、経過時間を測定し、タイマ制御を行う場合のスタートスイッチとしても機能させることができる。



(a) 正面から望む (b) 横から望む

図1 NXT を利用した倒立振り型走行体



図2 NXT 本体



(a) 側面 (b) センサ部分
図3 光センサ (反射型)



図4 ジャイロセンサ

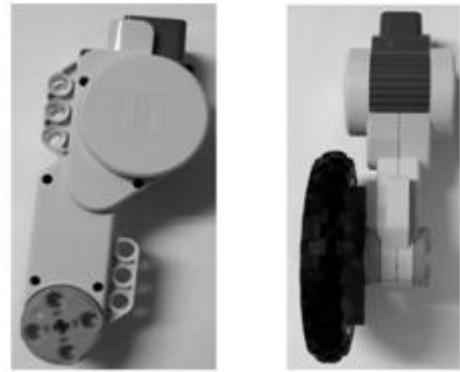


図5 タッチセンサ

(5) アクチュエータ（モータ）

LEGO MINDSTORMS NXT 付属のアクチュエータは、DCモータである。きょう体内部には歯車と軸が組み付けられ、車輪を取り付けることができる（図6）。また、内部に角度センサが組み込まれており、サーボモータとしても制御できる。

回転速度は、プログラミングの際に回転パワーを指定することにより制御することができる。また、サーボ機能を持っているので、回転数や回転角度（精度 $\pm 1^\circ$ ）を細かく指示することが可能である。



(a)側面から望む (b)タイヤ取付け状態
図6 アクチュエータ

4 倒立振り子とライトレースの制御

倒立振り子とは棒の下端を支点として、垂直に立てた状態をいう。したがって、棒が垂直な状態から少しでも変動すると倒れてしまい、制御をしないかぎり不安定な状態となる。

この倒立状態を保つ制御の仕組みは、次のようになる。図7に示すように、倒立振り子下端の支点が台車に取り付けられているものとする。台車には車輪が付いており、倒立振り子の倒れる角度や角速度をジャイロセンサで検出し、モータ（車輪）を駆動して、振り子が倒れる方向に台車を移動させると、その反動によって振り子が逆方向に回転しようとする。この制御により、垂直状態が保たれることになる。具体的な例を示すと、手のひらに載せた傘を垂直に立てる遊びの仕組みと同じである。一方、NXT 走行体に装備された一つの光センサによって、白い床面に描かれた黒い線を追従させるためには、次のように考えることにする。図8に示すように、光センサが白い面を認識している場合は、常に右方向に進む（左右二輪のうち、左側の車輪の回転速度を右側に比べて高速とする）ようにする。黒い線を認識した場合、次は左方向に進む（左右二輪のうち、右側の車輪の回転速度を左側に比べて高速とする）ようにする。これを繰り返すことで、追従すべきコースに対してジグザグ状態になるが、直線、曲線にかかわらず追従が可能になる。これらの2つの仕組みを組み合わせることで、NXT 走行体がジャイロセンサにより姿勢（倒立振り子）を制御されながら、コースをトレース（追従）できることになる。

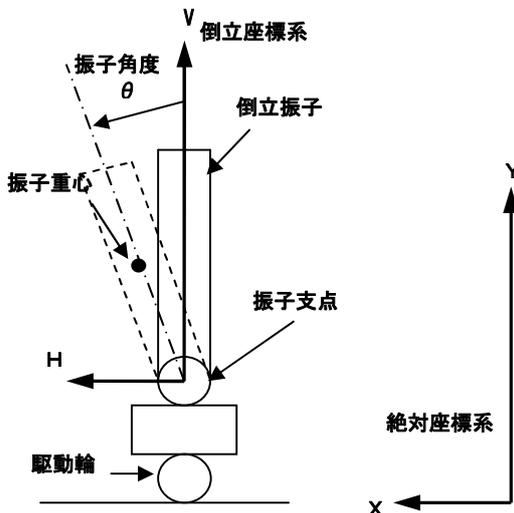


図7 倒立振り子下端の支点が台車にある場合の制御

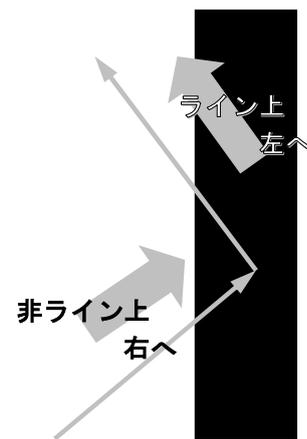


図8 一つの光センサによるライトレースの仕組み

5 ソフトウェア

最終的にはNXT（走行体）にUSB 接続して、プログラムを転送することになるが、そのプログラムを開発するためのPCについて、次に示すような環境を整備する必要がある。

(1) nxtOSEK の導入

nxtOSEK はオープンソースの LEGO MINDSTORMS NXT 用開発/実行環境であり、次のような特長を持っている。

- ア リアルタイムOSでI/O ドライバが定義済み
- イ ANSI C/C++言語環境開発
- ウ NXT のモータやセンサ等のデバイスに対するリアルタイム制御用C/C++言語API
- エ 自動車電子制御用OS標準規格であるOSEKや μ ITRON に対応したマルチタスクスケジューリング機能
- オ 無線操縦カーや今回のNXT 走行体などの制御アプリケーションを含むサンプルプログラムが豊富

(2) Cygwin の導入

nxtOSEK 使用の前段階として、Cygwinをインストールした。これは、Linux 系OSの様々なフリーソフトウェアをWindows (マイクロソフトコーポレーション) 上で稼働できるようにする。

(3) Eclipse CDT の導入

NXT 走行体で稼働するプログラムの開発に、はんだ用プログラム言語を使用するメリットの一つとして、Web 等で入手できる多数の成果物の利用が挙げられる。そこで、nxtOSEK を使用したプログラムの開発、ビルド、アップロード (NXT への転送) などという一連の作業は、Eclipse CDT から行えば簡略化できる。

6 プログラム

図9に示したように、このプログラムは、起動させると光センサが発光する。その状態で、タッチセンサが押されるまでは、走行待機となる。タッチセンサが押されると、ジャイロセンサの値をリセットし、ジャイロセンサの値が 600[deg/sec] (振子の倒立状態を維持している角速度の基準検出値) であるか検出する。検出した値が、600[deg/sec]に対してプラス側あるいはマイナス側に偏っていれば、それに応じてモータを動かし (前進あるいは後進)、ジャイロセンサの値を600[deg/sec]に近づける。再び、光センサの値を読み取る。光センサの白い面及び黒い面に対する検出値を、それぞれ「白い値」及び「黒い値」とする。ここで、光センサの検出値が、 $(\text{白い値} + \text{黒い値}) / 2$ より大きければ、黒と判断し左折をする。逆に、 $(\text{白の値} + \text{黒の値}) / 2$ より小さければ、白と判断し右折をする。リセットスイッチが押されなければ、ジャイロセンサ値をリセットして繰り返す。リセットスイッチが押されたら、プログラムを終了する。

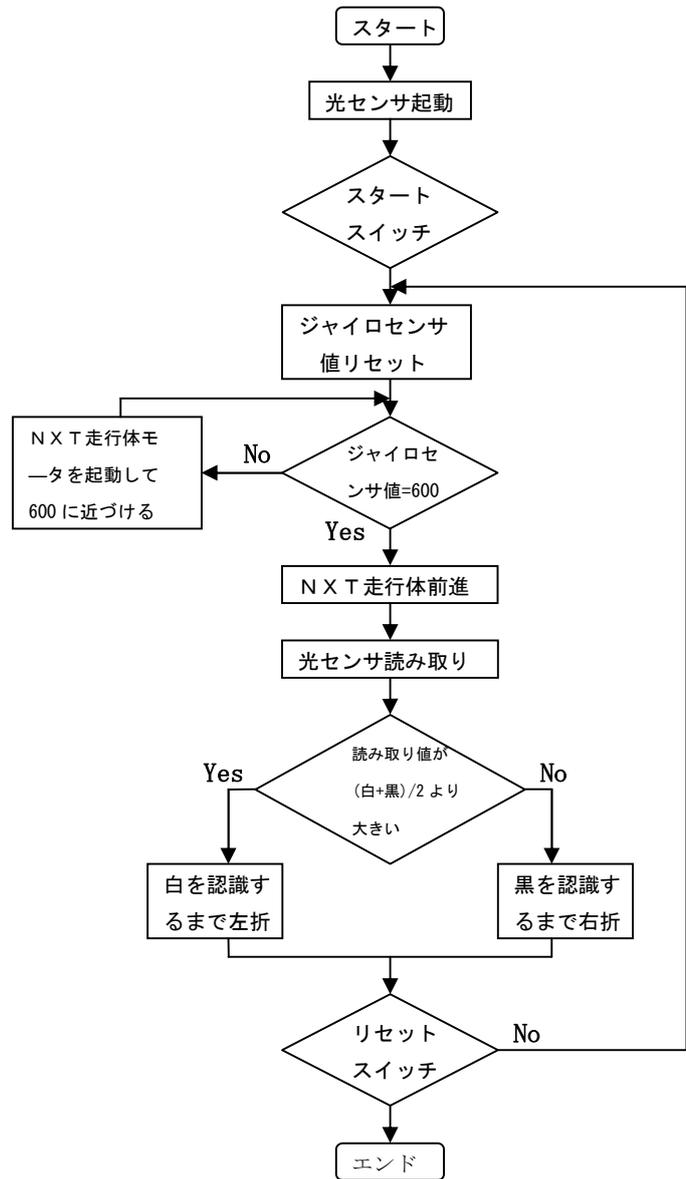


図9 プログラムのフローチャー

7 プログラム設定値の実験結果及び考察

(1) 光センサ値の調整

光センサが一つの場合、ラインレース技法の基本としてエッジ走行がある (図10)。通常は、大きく左右に振れてジグザグするような走行 (無駄な走行距離が多く、走行距離も長くなる) になってしまう。そこで、黒と白の中間の値を検出している場合の動作について定義することで、ジグザグの振れの大きさを狭めることができる。すなわち、ラインレースをしていると、黒と白の境界線上でラインの値を検知することがあるが、検出値は黒の値より低く、白の値より高い値になる。黒の値になる前の値で左に曲がることによって、小さい振れ幅で走行することができ、より短時間での走行が可能になる。

実験の結果、光センサを白い面に向けた場合の検出値は500 で、黒い面の場合には700 であった。これらの値をプログラムに設定することで、コースからの脱線確率を減少させることができた。この2つの値の中間範囲（例えば $600 \pm \alpha$ ）は、白色と黒色とが同時に存在している面の範囲にセンサが向いていることを意味しているため、この条件をプログラムに反映することで、さらなる高速化が期待できる。

(2) ジャイロセンサの調整

提供されている初期設定プログラムでは、ジャイロセンサに関する判断値は、610[deg/sec]に設定されている。この値を変化させて、倒立振り子状態で二輪走行を行い、その安定性の違いを調べた。変化させた値の範囲は、580[deg/sec]から640[deg/sec]の間で10間隔に設定した。結果的には、初期設定プログラムでの初期値610[deg/sec]にほとんど近い600[deg/sec]となった。

(3) 比例制御（P制御）

エッジ走行（オンオフ制御）ができるだけジグザグ走行にならないように灰色検出状態の数値を検出できるようにした。しかし、オーバーシュート（ON状態）とアンダーシュート（OFF状態）を繰り返す「ハンチング」を起し、制御量が一定にならないという問題が発生した。走行体の走行制御としてオンオフ制御を取り入れた場合は、走行制御に対する調整が必要となる。図11の走行軌跡を見て、以下の点に気付いた。

- ア 切替えタイミング（波形の頂点）が、目標値を超えた位置にあること。
- イ いつまでたっても、波を打ち続け安定しないこと。

切替えタイミングが目標値を超えた位置にあるのは、光センサによる光量検出の遅れとサーボモータの旋回方向切替え時間などが起因している。オンオフ制御を取り入れたときの短所であり、走行中も以下のような影響を受けている。

- ア 走行動作が不安定であり、揺らぎの影響を受けやすく、ラインから外れる可能性が高い。
- イ ライントレースの軌跡が安定しないことで、高速走行することは難しい。

オンオフ制御では操作量が一定なので、オーバーシュートの山の面積やアンダーシュートの谷の面積が操作量の累積に比例し、面積が広いほど走行体はラインに対して急角度で移動することになる。ラインに復帰する際には、できるだけ小さな角度で緩やかに戻るのが望ましいので、偏差が小さいときにも全力で旋回する必要はない。

そこで、偏差と操作量を比例させる比例制御を適応した。比例制御では、偏差×定数で操作量を決定する。偏差が大きいくほど操作量は増大し、逆に、小さいほど操作量は減少する。そして、0付近になったときの角度は、オンオフ制御に比べてずっと浅くなるためオーバーシュートを小さくできる。パラメータ（定数）が適当であれば、ハンチングは数回で収束し、制御量が安定する（図12）。結果は、0.96~1.0の数値で安定した。

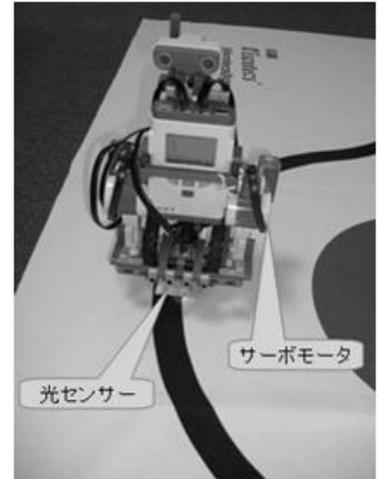
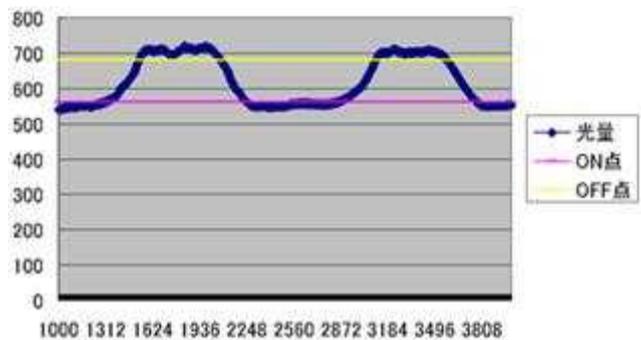


図 10 走行体のエッジ走行



(ON点:560 OFF点:680)

図 11 走行軌跡

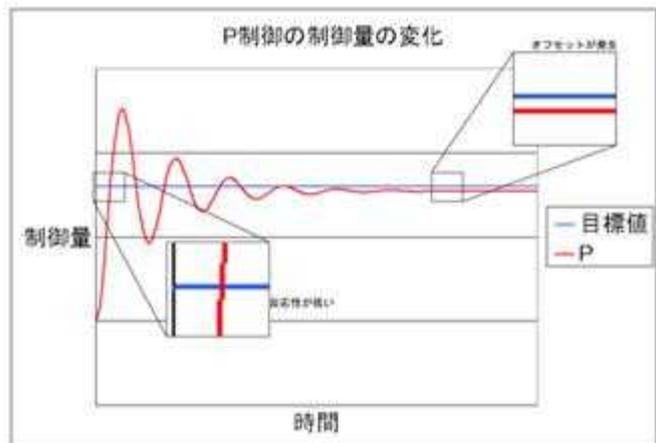


図 12 P制御の制御量の変化

比例制御を使うことで、現在の制御量に応じた操作量を決定することができた。つまり“現在の状況に応じて、力の抜き方を決める”のが比例制御といえる。

比例制御の欠点は、偏差が小さいときは操作量が小さくなり過ぎ、制御量が目標値とずれたところで安定してしまうことと、制御量が急激に変化した際の反応性が低いという点である。この制御量と目標値の差を「定常偏差」あるいは「オフセット」という。P制御では、現在の制御量だけを操作量の決定要素に使用するため、操作量は非常に小さな値になる。そのため、制御量の急激な変化に対応できず、オーバーシュートが発生し、反応性が低くなることがわかった。

IV 研究のまとめと課題

1 研究のまとめ

LEGO MINDSTORMS NXT を使ったロボット製作やプログラミングは、「制御システムを楽しむ」という点においては優れた教材である。特に、プログラムや判断値の変更など試行錯誤を繰り返し、より実体化した存在として取り入れられたプログラムが完成したときは明確な自己評価へとつながった。しかし、ロボット教材を使った学習形態は、ループの中で繰り返し進める形式「トライ アンド エラー」であり、試行錯誤を繰り返すため、生徒たちのモチベーションの高低によっては、最後まで持続しないことが予想される。

2 今後の課題

今回開発したICT 教材を活用して実験・実習を行うことで、制御システムへ興味関心を持たせつつ、課題研究などへつなげることができると考える。高度な制御技術が必要となるETロボコン大会などへの参加を想定した場合には、今回研究したNXT 走行体では、何らかの原因でトレースラインを外れ、コースアウトする現象が起きたため、これを回避する手立てを確立しなければならない。

ETロボコン大会では、競技に参加するロボットはすべて同じ仕様・形状に統一されている。バッテリーやモータ、その他パーツもすべて同じであるため、優劣を決定するのはソフトウェア次第となる。大会で設定された課題をクリアするためには、NXT 走行体がコースアウトしたと判断し、トレースを再開するリカバリプログラムの開発が必要と考える。

<参考文献・参考URL >

- ・アイティメディア株式会社『「E Tロボコンではじめるシステム制御」最新記事一覧』 2010
http://monoist.atmarkit.co.jp/mn/kw/et_robo_seigyو.html (2013.1.23)
- ・牛丸真司『nxtOSEK チュートリアル』2011
<http://www2.denshi.numazu-ct.ac.jp/lego/NXT/nxtOSEK/index.html> (2013.1.23)
- ・sourceforge.net『nxtOSEK インデックス』 2012
<http://lejos-osek.sourceforge.net/jp/index.htm> (2013.1.23)
- ・Software Developer's Think IT 『第1回ETロボコンを仕事に役立てる』 2010
<http://thinkit.co.jp/story/2010/08/02/1680?page=0> (2013.1.23)
- ・高津裕人, 佐藤祐哉, 茂木拓也, 金田 徹, 大貫雅和 2010 技術論文『試行錯誤によるE Tロボコン出場のためのN X T走行体の開発』関東学院大学工学部研究報告第54-1巻

<使用エディタ>

- ・Cygwin (フリーソフト)
- ・GNU ARM (フリーソフト)
- ・LEGO MINDSTORMS NXT Driver (フリーソフト)
- ・nxtOSEK (フリーソフト)
- ・Eclipse CDT (フリーソフト)

<商標>

- ・Windows は、マイクロソフトコーポレーションの登録商標である。