

# チーム「走れっ!! GB-TANK」

中村将太 本庄悠

(サレジオ工業高等専門学校 電子工学科)

## 1.はじめに

前回、2006年度に行われたコンテストでは、私たちの先輩はプレゼンテーションの部門で見事奨励賞頂くことができたが、ロボットカーの走行競技においては良い結果を残すことはできなかった。そこで今回は、その反省を踏まえて、私たちが新たにGPSロボットカー製作に取り組むことになった。

## 2.コンセプト

今回は、ずばり「上位入賞できるロボットカー」である。そのためにも前回の走行競技で果たすことができなかった「完走」も目標に掲げ、製作を進めてきた。具体的には、なるべくハードウェアの製作に時間を掛けず、ソフトウェア（アルゴリズム、プログラミング）に関する製作に重点を置いて製作するということである。

去年と今年との大きな変更点は、駆動部にラジコン戦車を利用し、GPS受信機からの測位データ処理などを行う中枢部には、ノートPCではなくH8マイコンを採用した点である。これにより、前回のコンテストにエントリーしたロボットカーとはまったくことなる構成となっている。

以上の仕様変更による利点は以下の通りである。

- ・ 既製品の駆動部を用いることで、安定した走行が可能。
- ・ 旋回や直進などの制御の簡易化。
- ・ 駆動部の製作時間の短縮。
- ・ ノートPC撤去による軽量化。
- ・ 外見のデザイン性。
- ・ ラジコン戦車の機能として大砲からのBB弾発射も可能。

## 3. ハードウェアについての概要

### 3.1. ロボットカー

今回ロボットカーに使用したラジコンカーの外観と内部を示す。以下に示す。



図1. ラジコンカー 外観

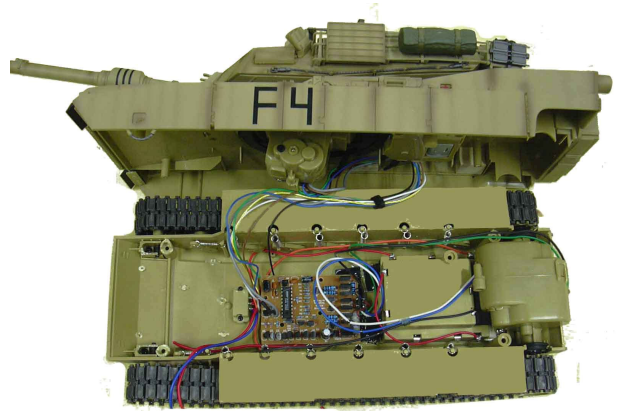


図2. ラジコンカー 内部

また、ロボットカーのパーツ構成を図1に示す。

表1. パーツ構成

構成要素	パーツ名	数量
GPS受信機	小型GPSモジュール	1台
駆動部	東京マルイ社製 M1A2_ABRAMS 1/24 RC BATTLE TANK	1台
データ処理	H8 3052F	1個
駆動部制御1	Microchip社製 PIC 16F873A	1個
駆動部制御2	東芝製 TA7267P	2個
キャタピラ回転数 検出センサ	秋月電子通商 P-00613	2個

### 3.2. ハードウェア構成

以下に回路構成を示す。

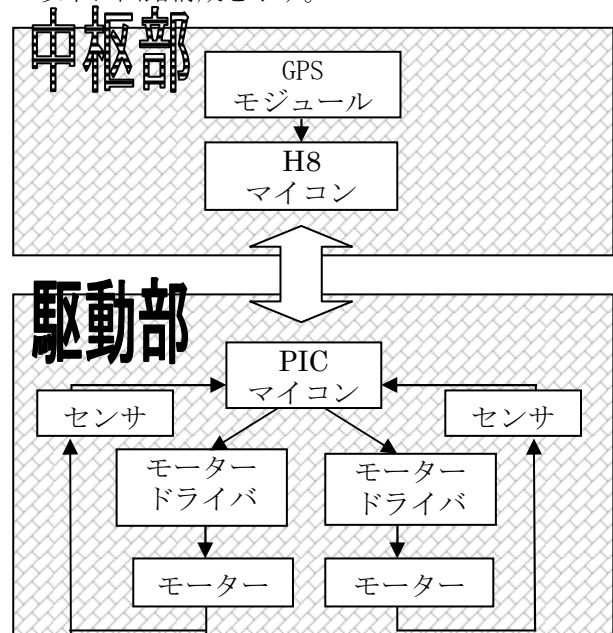


図3. ハードウェア構成

以下に図 3 で示した各ハードウェアの用途を示す。

#### GPS モジュール

GPS の電波を受信し、緯度や経度などの情報を H8 マイコンへ送信する。

#### H8 マイコン

GPS モジュールから送られてきた緯度、経度を 1 つ前の単位時間における緯度、経度と比較し、今どの方角に進んでいるのかを判断する。更に、目的地の緯度、経度も比較することで、現在地から目的地がどの方角にあるのかも判断する。この 2 つの情報により、マシンが次にどんな動作をすればよいのかを判断し、制御信号を PIC マイコンに送信する。

#### PIC マイコン

H8 マイコンから送られてきた制御信号に応じて、ロボットカーの進行動作パターン（直進、回転、停止）をここで判断し、モータードライブにその進行パターン信号を出力することで駆動部を制御する。また、正確な直進をするための軌道修正制御も行う。具体的には駆動部の車輪の回転数を、センサ（パルス検出センサ）を用いて、そこから送られてくるパルスをカウントし、それにより、左右のモータを PWM 制御して修正する。また、一定回数のパルスが送られてくるまでの回転動作信号の出力などを行う。

#### モータードライブ

ここでは、微弱な PIC マイコンからの出力信号を増幅してモータに送り、モータを動かす。また、モータからの逆起電力を遮断し、PIC マイコンの破損を防ぐ役目もある。

#### モータ

モータードライブからの出力により正転、逆転、停止をする。

#### センサ

ロボットカーのキャタピラを回転させている車輪の凹凸部分にフォトリフレクタの赤外線を当て、この状態で車輪を回転させることで、凸でリフレクタが隠れる時は光が反射し、1 (+5V) を出力する。また、リフレクタが凹で隠れない時、つまり反射なしの時は 0 (0V) を出力し、0、1 のパルス波形を得る。そして、このパルスは PIC マイコンに送られる。

### 4. ソフトウェアについての概要

#### 4.1. 使用言語

今回使用した言語は

H8 マイコン：C 言語

PIC マイコン：アセンブラ言語である。

これらの言語を採用した理由は、H8 マイコンでは、三角関数や配列処理、ポインタなど複雑な処理が必要なためである。また、PIC マイコンは、特にどんな言語でも良かったのだが、単純な処理プログラムを容易に構築できるアセンブラ言語を採用した。

#### 4.2. 基本動作

ここで、マシンの基本動作を説明する。基本動作は、直進、回転、停止のみである。具体的には、H8

からデータが送られてくるまでは直進し、データが送られてきたら停止して、そのデータどおりの方向を向くまでその場で回転し、その後また直進、といった動作の繰り返しになっている。図 4 に基本動作のパターンを示す。

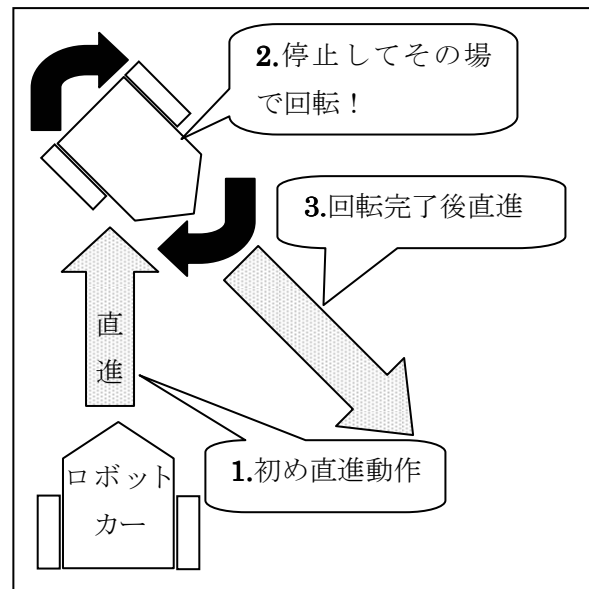


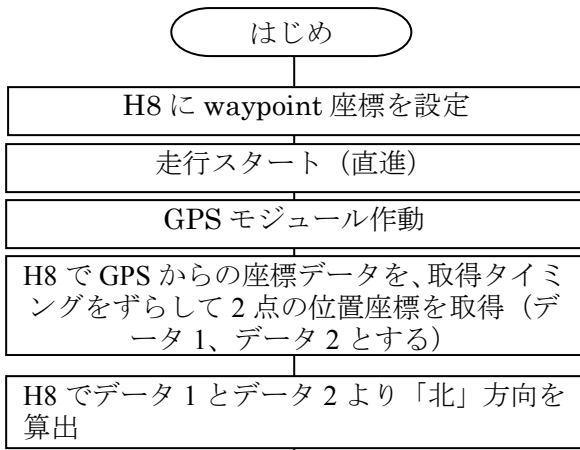
図 4. マシンの基本動作

#### 4.3. 全体の流れ

ここで、ロボットカーの動き、航法アルゴリズムのフローチャートを示す。なお、ページ数の都合上ここでは簡単に航法アルゴリズムを図示する。また、図 5～図 10 のアルゴリズムの構成を下記で説明する。

表 2. アルゴリズム解説一覧

図番号	図の解説
5	走行スタート時にマシンの基準となる方角（北）を GPS 受信機からの座標より算出する。
6	マシンは、基準となる方角（北）に進行方向を向ける。
7	マシンの現在地から目標の waypoint へ到達するための進行方向を算出する処理。
8	マシンを目標の waypoint の方向に向かせる処理。
9	目標の waypoint に到達したらマシンを一時停止させ、まだ到達していない waypoint がないか確認し、まだ、waypoint があれば、再度図 7、図 8 を到達するまで繰り返す。また、waypoint を全て通過したなら、図 10 の処理へ進む。
10	図 9 の処理で全ての waypoint を通過したことが確認できたなら、マシンがスタート地点へ戻るよう処理する。



PIC 制御 1 へ  
図 5. 航法アルゴリズム 1

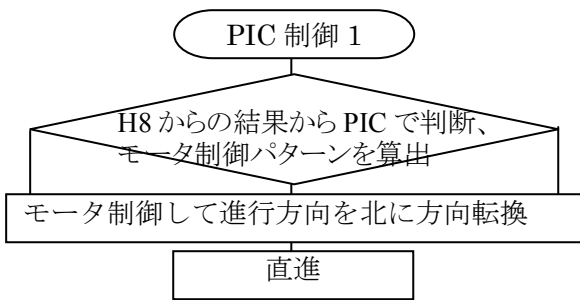


図 6. 航法アルゴリズム 2

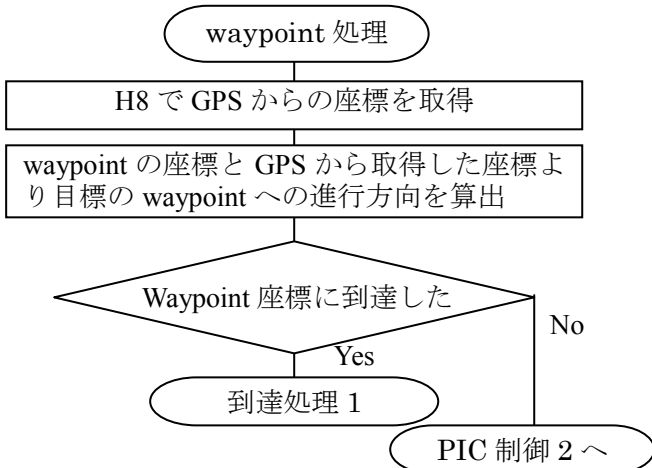


図 7. 航法アルゴリズム 3

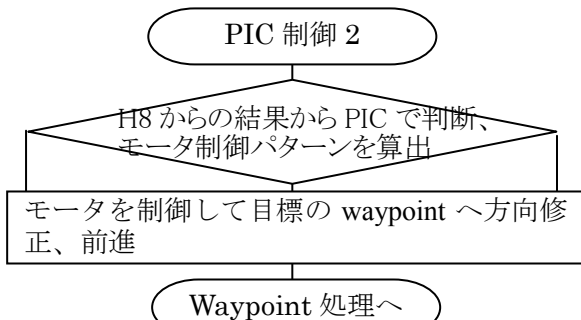


図 8. 航法アルゴリズム 4

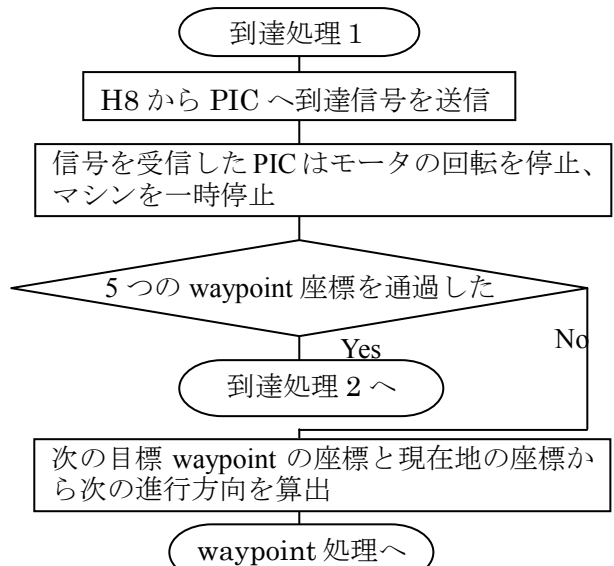


図 9. 航法アルゴリズム 5

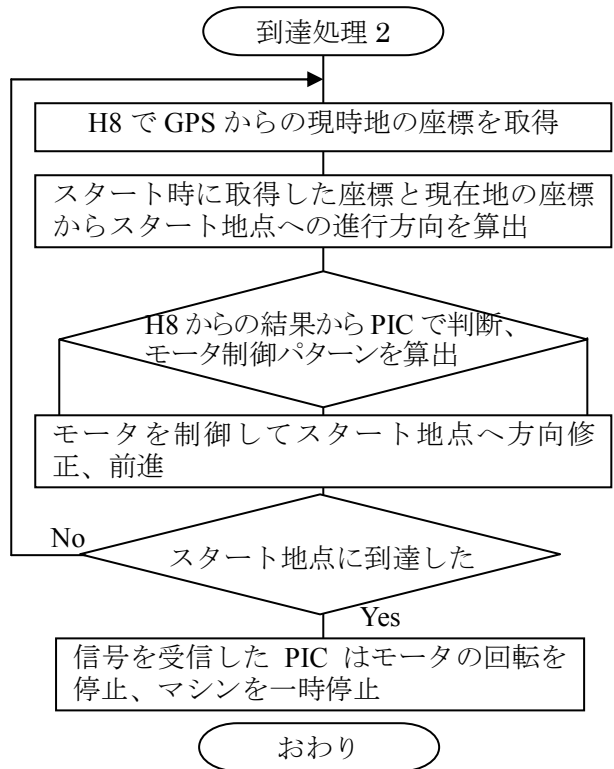


図 10. 航法アルゴリズム 6

これらのフローチャートより、先の解説一覧表からも述べたように、走行しはじめてから最初に北を向かせることで、その方向が基準となり、そこから waypoint の方向へ何度かの角度でマシンを回転させるかが計算から導ける。その具其他的な説明は後に示す。

#### 4.4. 北を向くまでの比較方法

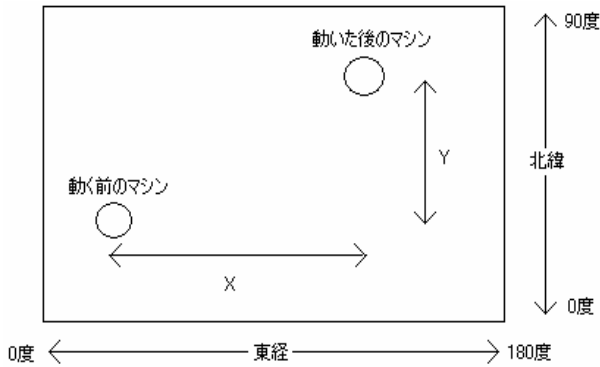


図 11. 北を向くまでの動き

ここで、目的地までたどり着くためのアルゴリズムを説明する。マシンは基準の方角、つまり北の方角を基準にして、基準と GPS のデータとを条件にあてはめて、回転を決める。従って、基準とする方角（絶対方位）を決めなければならない。絶対方位は東西南北のどの方角にしてもよいが、今回は分かりやすく北にすることにした。そうすると目的地を探す前にマシンは必ず北を向いていなければならない。

この北を決定するには、まずマシンを数秒間直進させ、直進する前の緯度、経度から直進後（数秒後）の緯度、経度を減算し、その差からマシンがどの方向を向いているのかで北を決めることができる。その後、北を向くように回転する。詳しい比較方法とその動作は次のようになっている。記号は図 11 を参照。

はじめに先ほど述べたとおり、2 点の緯度、軽度の差を下式より求める。

$$\begin{aligned} (\text{動く前の経度}) - (\text{動いた後の経度}) &= X \cdots (1) \\ (\text{動く前の緯度}) - (\text{動いた後の緯度}) &= Y \cdots (2) \end{aligned}$$

次に求めた値が以下の①～⑥までの条件式に当てはめる。

$$\begin{array}{lll} X > 0 \cdots \textcircled{1} & X = 0 \cdots \textcircled{2} & X < 0 \cdots \textcircled{3} \\ Y > 0 \cdots \textcircled{4} & Y = 0 \cdots \textcircled{5} & Y < 0 \cdots \textcircled{6} \end{array}$$

以下の組み合わせにより、北に向くためのマシンの動作パターンが上記の①～⑥の条件式にあてはまった組み合わせにより、次のマシンの動作が決まる。

- ①④の時…角度  $\phi = \{90 - \tan^{-1}(Y/X)\}$  度分左回転。北向き完了。
- ①⑤の時…東に進んでいるので 90 度左回転。北向き完了。
- ①⑥の時…90 度左回転。まだ北を向いていないので直進し次のデータを待つ。
- ②④の時…北に進んでいるので完了。
- ②⑤の時…止まっているので直進し次のデータを待つ。
- ②⑥の時…南に進んでいるので 180 度右か左に回転。北向き完了。
- ③④の時…角度  $\phi = \{90 - \tan^{-1}(Y/X)\}$  度右回転。北向き完了。
- ③⑤の時…西へ進んでいるので 90 度右回転。北向き完了。
- ③⑥の時…90 度右回転。まだ北を向いていないので直進し次のデータを待つ。

#### 4.5. 目的地に着くまでの比較方法

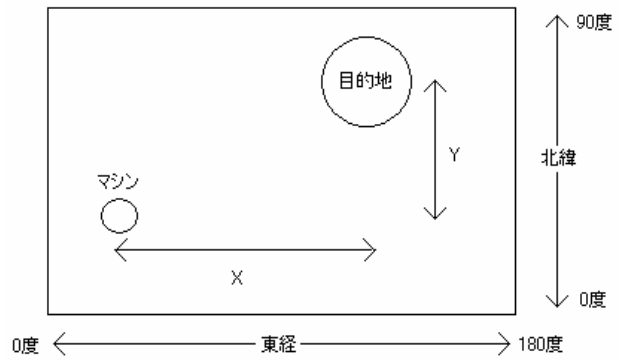


図 12. 目的地に着くまでの動き

マシンと目的地の関係を図にすると図 12 のようになる。マシンが目的地に辿り着くには、マシンから目的地がどの方角にあるのかわからなければならない。そのため、目的地の経度、緯度からマシンの経度、緯度を引き、その差から方角を求める。比較方法は先ほどの北を向くまでのものとほとんど同じだが、多少違うので次に示す。記号は図 12 を参照。

$$\begin{aligned} (\text{目的地の経度}) - (\text{マシンの経度}) &= X \cdots (3) \\ (\text{目的地の緯度}) - (\text{マシンの緯度}) &= Y \cdots (4) \end{aligned}$$

$$\begin{array}{lll} X > 0 \cdots \textcircled{1} & X = 0 \cdots \textcircled{2} & X < 0 \cdots \textcircled{3} \\ Y > 0 \cdots \textcircled{4} & Y = 0 \cdots \textcircled{5} & Y < 0 \cdots \textcircled{6} \end{array}$$

①④の時…

- $X > Y$  ⇒ 東北東を向くまで回転して直進。
- $X = Y$  ⇒ 北東を向くまで回転して直進。
- $X < Y$  ⇒ 北北東を向くまで回転して直進。

①⑤の時…東を向くまで回転して直進。

①⑥の時…

- $X > |Y|$  ⇒ 東南東を向くまで回転して直進。
- $X = |Y|$  ⇒ 南東を向くまで回転して直進。
- $X < |Y|$  ⇒ 南南東を向くまで回転して直進。

②④の時…北へ向くまで回転して直進。

②⑤の時…停止

②⑥の時…南を向くまで回転して直進。

③④の時…

- $|X| > Y$  ⇒ 西北西を向くまで回転して直進。
- $|X| = Y$  ⇒ 北西を向くまで回転して直進。
- $|X| < Y$  ⇒ 北北西を向くまで回転して直進。

③⑤の時…西を向くまで回転して直進。

③⑥の時…

- $|X| > |Y|$  ⇒ 西南西を向くまで回転して直進。
- $|X| = |Y|$  ⇒ 南西を向くまで回転して直進。
- $|X| < |Y|$  ⇒ 南南西を向くまで回転して直進。

#### 5. 現状と今後の課題

10 月現時点では、マシンの製作に問題があり、H8 による GPS データ処理のプログラムが未完成である。また、PIC マイコンも PWM の組み込み方法が未解決である。今後は、コンテストまでにそれらが解決できるよう製作活動に取り組んでいこうと思う。