

マイコンカーをさらに速くする

100%には、まだ上がある

～「真の100%」を求めて～

栃木県立栃木工業高等学校
電気科 小林 幸雄

1. はじめに

2000年にマイコンカーの研究を始め、もう20年が過ぎようとしている。研究を始めた当初はAdvancedクラスのマイコンカー中心に活動を進めてきたが、栃木工業高校に赴任してからの5年間は諸般の事情によりBasicクラスのマイコンカーのみを扱うようになった。この5年間、車体やプログラムに対して、考えられる様々なアイデアと工夫を盛り込み、全国のトップクラスのレベルに少しずつ近づいている感がある。



図1 今年試作したFFマイコンカー

図1は、2019年マイコンカーラリー全国大会 BASICクラスの優勝マシンのスペックを参考にして組み上げたマシンで、これから走行データの検証をおこなう予定のものである。FFマイコンカーの走行試験は初めてなので、どのような結果が得られるか非常に楽しみである。

昨年度、生徒にマイコンカーのPWMに関する基本的な内容をオシロスコープを用いて解説しているとき、偶然「ストレートの走行スピードを上げることのできる技術」を見つけることができた。本稿は、この技術を課題研究などで生徒が簡単に応用できるように、BASICクラスのサンプルプログラムをベースに説明をする。プログラムの一部を変更するだけで、簡単に実現できるものなので、その効果を是非とも体感してほしい。

2. 軽量化だけが工夫ではない

部活動や課題研究でBASICマイコンカーの研究活動をする、はじめに「車体の軽量化」の作業にとりかかることが多い。車体が少しでも軽くなれば性能向上につながるが、そのうち軽量化にも限界があることに気が付く。また、「FF車が速い、FR車が速い」と駆動方式の検討にも取り組むが、全国大会の結果を吟味すると、これが速いと断定することも難しい。

全国大会の映像がHP上にあり、画像解析すると直線やカーブでのスピードを求めることもできる。「全国トップクラスのBASICマイコンカーのR600カーブ走行速度が、自分のマイコンカーの直線部分のトップスピードとほぼ同じ」というデータをみたときは、残念で絶望的な気持ちになった時もあった。「他校にできることは、自分たちもできるはず」という思いで、研究活動を進めてきた。

そして、「どうやったらあの驚異的なスピードが実現できるのか」「なぜ、自分たちは、あのスピードを再現できないのか」という現実を突き付けられ、マイコンカー指導者としての限界を感じる事となる。この5年間で、間違いなく技術的な前進はしてはいるが、飛躍的な進歩に至る解決策を見つけることができなかった。

3. 判断基準「平均速度」

大会結果を客観的な尺度で評価するため、コース全長を走行タイムで割り算した、「平均速度」というデータで過去の創意工夫を判断している。栃木工業高校で生徒の記録したデータを1年ごとにまとめると以下のようなになる。

大会名	コース全長 m	走行タイム 秒	平均速度 m/秒
2017全国大会	64.52	30.98	2.08
2018全国大会	50.43	23.54	2.14
2019北関東地区予選	50.00	23.00	2.17
2020北関東地区予選	50.55	21.91	2.31

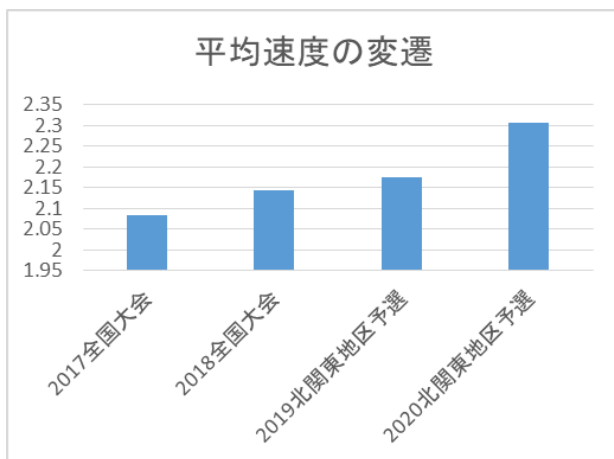


表1 栃木工業高校のBASICマシンの速度変遷

当然、大会コースの難易度や電池の充電状態等によって、走行タイムにばらつきが生じる。表1の全国大会や北関東地区予選のコースレイアウトは難易度の高いものばかりなので、平均速度を客観的な尺度として用いても十分意味のあるものと考えている。

表1のデータから、2020年北関東地区予選で、劇的に平均速度が向上していることがわかる。実はこの大会のコースレイアウトは、「プログラムで対応していない走行パターン」を含むものであり、完走をするために電池を消耗させ、スピードを抑えてなんとか走行したときのものであった。それでも、過去最高の平均速度を記録することができた。

JMCR2020全国大会 Basic Class 予選 結果			
2020/1/11(土)・12(日) 西日本総合展示場 新館(A展示場)			
順位	学校名	走行タイム	平均速度
1	福岡県立苅田工業高等学校	22.97	2.61
2	秋田県立大曲工業高等学校	24.13	2.49
3	愛媛県立松山工業高等学校	24.37	2.46
4	福岡県立戸畑工業高等学校	24.89	2.41
5	岡山県立笠岡工業高等学校	25.01	2.40
6	福井工業大学附属福井高等学校	25.18	2.38
7	宮崎県立佐土原高等学校	25.28	2.37
8	岡山県立水島工業高等学校	25.33	2.37
9	宮崎県立日南振徳高等学校	25.34	2.37
10	広島県立宮島工業高等学校	25.68	2.34
11	神戸市立科学技術高等学校	25.77	2.33
12	岡山県立東岡山工業高等学校	25.84	2.32
13	熊本県立熊本工業高等学校	26.05	2.30
14	群馬県立太田工業高等学校	26.35	2.28

表2 JMCR2020全国大会BASICクラスの結果

表2は北九州の小倉で開催された2020年JMCR全国大会の予選結果である。残念ながら栃木工業高校の

BASICマシンは、2019年全国大会と同じハーフラインで落車という結果になったため、全国大会の記録を残すことができなかった。

しかし、現在の平均速度と全国大会の平均速度を比較することで、全国でのポジションを確認することができ、全国13位程度の平均速度をだしていることがわかる。

本稿は、走行速度を劇的に向上させた方法がソフト記述に依存しており、それが簡単に実現できるもので、今後のBASICクラスやAdvancedクラスのマイコンカー全体の向上につながると考え、ここに紹介するものである。

4. よき車体がよきタイムにつながる

走行用電池4本(1.2V 2次電池4本)という限られたエネルギーで最高のパフォーマンスを実現するためには、タイヤの走行抵抗をギリギリまで削減する必要がある。走行抵抗を減らすためには、タイヤに貼るシリコンシート幅を減らす必要がある。全国大会では、シート幅が15mm程度のものがスタンダードになってきている。

次に、ギアボックスのギア比が固定されてしまうため、タイヤ直径が重要なポイントになる。全国大会上位のタイヤ直径データが公開されているため、これを参考にすると検討時間の短縮につながる。そして、ホイールを可能な限り軽く作ることで、加減速のパフォーマンスを著しく向上させることができる。

また、指定モータのばらつきが大きいいため、その選定に対しても注意が必要である。特にBASICマシンは、1.2V×4本の電圧でモータを駆動するため、特性のそろったモータを2個選別する必要がある。選別をするためには、2.4V程度の電圧で数秒間無負荷回転させる。モータの様子を観測し、「振動が少なく、軽い音で回転するもの」かつ「無負荷電流の値がそろっているもの」を選別して採用している。

BASICマイコンカーには、「フリーモード」というオプションが用意されている。是非ともこのフリーモードを活用してほしい。ストレート部分などで、ステアリングを左右に大きく振りながら蛇行走行していたマシンが、このフリーモードを活用すると、素直な走行が可能となる。そのとき、ハンドリングのセンター値を正確に求めることが大切である。

フリーモードを用いてストレートを走行させると、「モータのPWMを100%にすること」が簡単に実現できる。そして、「100%のPWM状態をいかに長時間継続できるか」が、走行タイム削減への近道となる。

5. PWM出力端子

図2のマシンは、構造上CPU基板が完全に隠れてしまい、ドライバー基板も半分程度隠れてしまうため、

オシロスコープのプローブをあてる場所に工夫が必要となる。

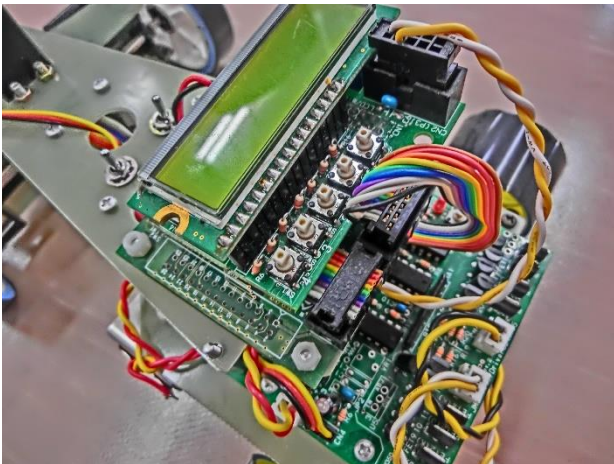


図2 各種基板の様子

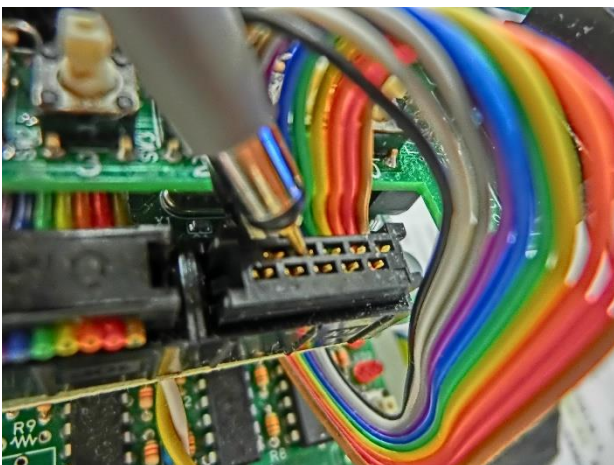


図3 オシロスコープ測定端子 (P2 7ピン)

CPU 基板の P2 の 7 ピンが、「左モータの PWM 出力端子」であることから、図3に示すように、この部分で PWM 波形を観測する。

6. タイマ RD リセット同期 PWM モード

サンプルプログラムに記述される motor 関数は、±100%の範囲で速度のパーセント値を引数にして PWM 動作が実現できるようになっている。たとえば motor(90, 90)と記述すれば、左右二つのモータを90%のパワーで動作させることができる。また、図4から PWM の制御周期は、16m 秒であることが確認できる。

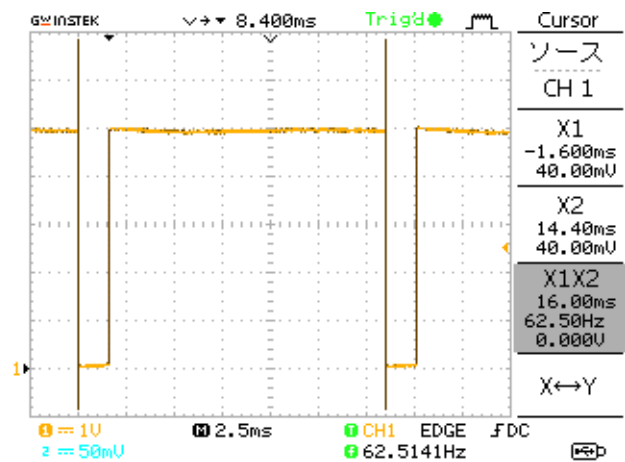


図4 90%PWM 出力波形

図5は99%PWM出力の観測波形、図6は100%PWM出力の観測波形である。

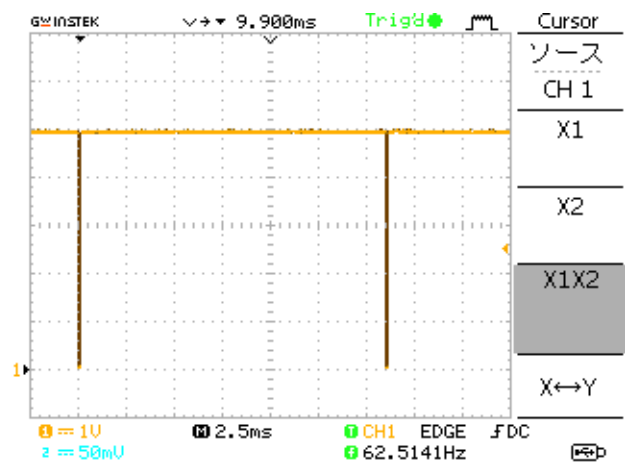


図5 99%PWM 出力波形

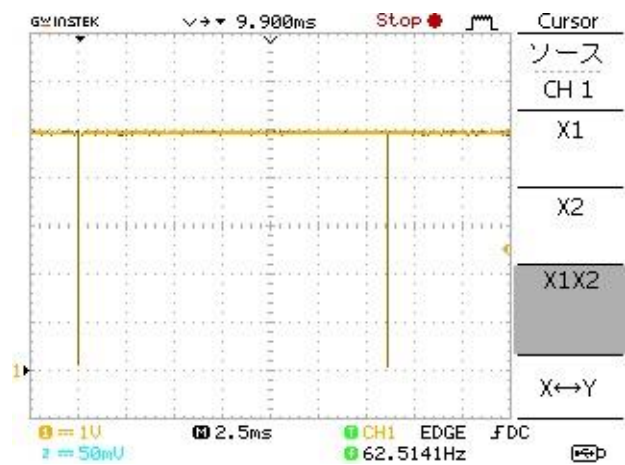


図6 100%PWM 出力波形

図6の100%PWMの場合でも、わずかに負のパルス波形が観測される。この部分を拡大したものが図7になる。約400n秒のパルス幅をもつ負のパルスであることが確認できる。

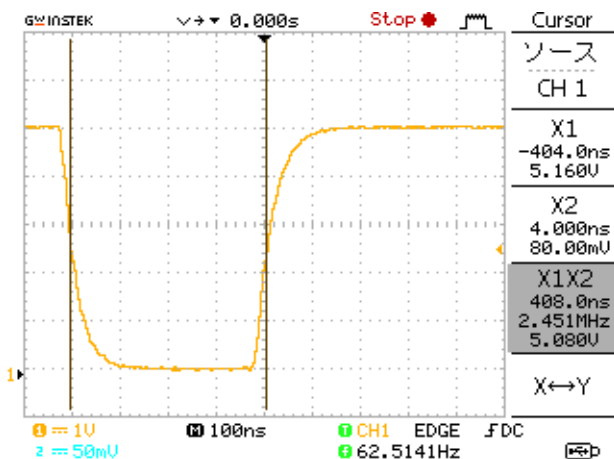


図7 100%PWM出力波形状拡大図

つまり、最高速度でマイコンカーを走行させるように motor(100, 100)と記述しても、実際には約 400n 秒だけフリーモードであればフリー走行期間が、フリーモードなしであればブレーキ走行期間が存在することになる。

この PWM 動作は、R8C38A マイコンを「タイマ RD リセット同期 PWM モード」に設定することで実現している。リセット同期モードの設定等は、参考資料に記載したプログラム解説マニュアル等に詳しく記述されているので、基本的な説明は省略する。

7. リセット同期モードを「リセット」させない

図 7 に示す約 400n 秒の負パルス無くすことができれば、マイコンカーのトップスピードはさらに上がることになる。

motor 関数の内部の trddr0 と trdcr0 に設定する変数は、100%のとき(式1)のようになる。(式1)の PWM_CYCLE に加算する数について、一つずつ増やしながらか実験をおこなった。実験の結果、(式1)の部分(式2)のように変更することで、約 400n 秒の負パルスの出現を無くすことができることがわかった

変更前： $(PWM_CYCLE - 1) \dots$ (式1)

変更後： $(PWM_CYCLE + 2) \dots$ (式2)

ここで、(式2)の意味を考えてみる。

PWM のパルス幅は、16m 秒という基本周期を設定する PWM_CYCLE の値と、リセットタイミングを設定するレジスタ trddr0 と trdcr0 の値の関係で定まる。つまり、基本周期の値より大きくリセットタイミングの値を設定することで、リセット動作を発生させなくすることが可能になる。ユーザーズマニュアル・ハードウェア編にも 100%PWM に関して触れてはいるが、ここまで具体的には記述されていない。このことに気が付けば、400n 秒の負パルスの発生を防止することが可能と

なる。そして、「真の 100%」を実現することができる。表現を変えれば、「リセットタイミングをカウンターの加算領域の外に設定する」ことで、「リセットの発生をおこさない」ということになる。

8. プログラムの変更点

①事前準備

motor 関数の中の先頭部分に、ディップスイッチの設定によって走行速度を設定できるプログラムが記述されている。まず、この部分をコメント化する。

②プログラムの変更

左モータ制御及び右モータ制御の先頭部分に、「100%の場合のときに、(式2)が設定できる」ように if 文を追加する。(最終ページ参照)

9. まとめ

本稿では、リセット同期 PWM モードをリセットさせないことで、「真の 100%」走行を実現できることを説明した。プログラムの変更で、100%走行部分の性能向上ができるとは思いつかなかった。これによって、平均速度 2.4m/秒の世界がすぐそこにきている。平均速度が上がると、様々な障害で大きな困難が出現し、その対応に追われることになる。

今までの経験から、マイコンカーの本質は「ポテンシャルの高いマシンをいかに制御するか」ということであり、車体が完成してからのプログラミング作業やデータ検証作業が、最も時間を必要とするものである。

しかし、栃木工業高校での 5 年間は、2 回の水害とその復興活動で多くの時間を費やすことになってしまった。ほとんどの研究活動がストップし、災害復興によって失われた 5 年間の中で、「真の 100%」走行の発見は偶然と奇跡によるものと思われる。現在、水没で失ったマイコンカーコースの製作を課題研究の中でおこなっているため、実際に走行・調整ができるのはしばらく先になる。

全国レベルのスピードを再現するための裏技と考えられる答えの一つが「真の 100%」走行だと思う。車体・モータ・電池といったハード的な条件を最適化することで、ストレート部分に対して全国クラスの走行再現が可能になってくると考えている。

本稿を読んだ皆さんに、マイコンカーの大会や調整会等で、実験結果や感想を教えてもらえれば非常にありがたいと思います。

【参考資料】

- ①R8C/38C グループ ユーザーズマニュアル ハードウェア編
- ②マイコン実習マニュアル R8C/38A 版
- ③マイコンカーキット Ver.5.1 kit12_38a プログラム解説マニュアル R8C/38A 版

変更後：真の100%走行

太字部分が変更・追加部分

変更前：オリジナルプログラム

```
*****  
/* モータ速度制御  
/* 引数 左モータ:-100~100、右モータ:-100~100  
/* 0で停止、100で正転100%、-100で逆転100%  
/* 戻り値 なし  
*****  
void motor( int accele_l, int accele_r )  
{  
    int sw_data;  
  
    sw_data = dipsw_get() + 5;  
    accele_l = accele_l * sw_data / 20;  
    accele_r = accele_r * sw_data / 20;  
  
    /* 左モータ制御 */  
    if( accele_l >= 0 ) {  
        p2 &= 0xfd;  
        trdgrd0 = (long)( PWM_CYCLE - 1 ) * accele_l / 100;  
    } else {  
        p2 |= 0x02;  
        trdgrd0 = (long)( PWM_CYCLE - 1 ) * ( -accele_l ) / 100;  
    }  
  
    /* 右モータ制御 */  
    if( accele_r >= 0 ) {  
        p2 &= 0xf7;  
        trdgrc1 = (long)( PWM_CYCLE - 1 ) * accele_r / 100;  
    } else {  
        p2 |= 0x08;  
        trdgrc1 = (long)( PWM_CYCLE - 1 ) * ( -accele_r ) / 100;  
    }  
}
```

```
*****  
/* モータ速度制御  
/* 引数 左モータ:-100~100、右モータ:-100~100  
/* 0で停止、100で正転100%、-100で逆転100%  
/* 戻り値 なし  
*****  
void motor( int accele_l, int accele_r )  
{  
    // int sw_data;  
  
    // sw_data = dipsw_get() + 5;  
    // accele_l = accele_l * sw_data / 20;  
    // accele_r = accele_r * sw_data / 20;  
  
    /* 左モータ制御 */  
    if( accele_l == 100 ) {  
        p2 &= 0xfd;  
        trdgrd0 = (long)( PWM_CYCLE + 2 );  
    } else if( accele_l >= 0 ) {  
        p2 &= 0xfd;  
        trdgrd0 = (long)( PWM_CYCLE - 1 ) * accele_l / 100;  
    } else {  
        p2 |= 0x02;  
        trdgrd0 = (long)( PWM_CYCLE - 1 ) * ( -accele_l ) / 100;  
    }  
  
    /* 右モータ制御 */  
    if( accele_r == 100 ) {  
        p2 &= 0xf7;  
        trdgrc1 = (long)( PWM_CYCLE + 2 );  
    } else if( accele_r >= 0 ) {  
        p2 &= 0xf7;  
        trdgrc1 = (long)( PWM_CYCLE - 1 ) * accele_r / 100;  
    } else {  
        p2 |= 0x08;  
        trdgrc1 = (long)( PWM_CYCLE - 1 ) * ( -accele_r ) / 100;  
    }  
}
```