

自転車ロボットの製作

三野 航太

1. 研究概要

私は自転車ロボットの製作を行っている。このロボットの目標は、2輪だけで走行、停止を倒れずに行うことである。

2. 研究の具体的内容

(1) 姿勢制御の仕組み

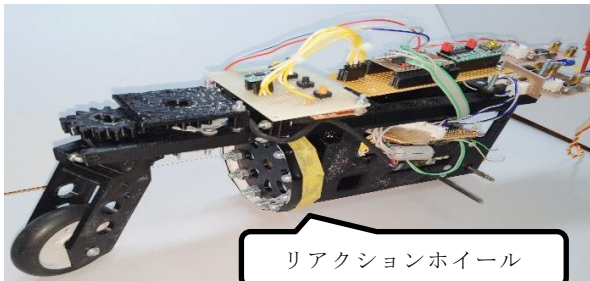


図1 車体横

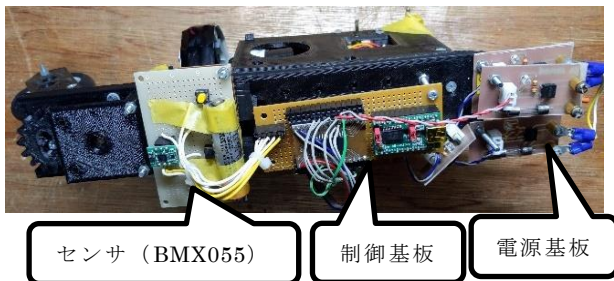


図2 車体上

図1内のリアクションホイールと呼ばれる円盤を回転させ姿勢制御を行う。

リアクションホイールは、3Dプリンタで制作した。重量が重いほうがよいため、ボルトを用いて重量を稼いでいる。

リアクションホイールを、時計回りに回転させたとする。そうすると車体全体は作用反作用の法則により、逆向きの力が働く。これを利用することで車体の傾きを制御することができる。

姿勢制御には、このほかにもハンドルを切

って倒れないように制御する方法もあるが、前進しながらでしか、姿勢制御ができないため今回はリアクションホイールを利用した。

(2) 車体の製作

車体は3Dプリンタで制作した。拡張性、メンテナンス性を重視し、接着剤などを極力使用せず、ねじ止めで固定するように工夫している。ハンドルなども、ステッピングモータを利用し、ハンドルを切ることができるようにしている。前輪には、ボールベアリングを搭載し、滑らかな動作が可能にした。

(3) 電子基板の製作

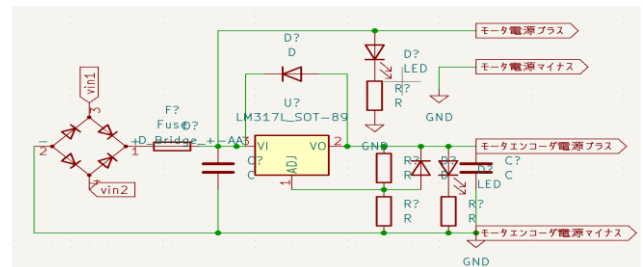


図3 モーター電源回路図 (kicad)

電子基板は、安全性、保守性、信頼性などを重視し製作を行った。

図3のように電源回路部では、ヒューズなどの過電流保護用の素子、ブリッジダイオードなどの素子などを内蔵し電源のプラスマイナスのミスなどの際にもマイコンの破壊を防止している。そのほかにも図3内の3端子レギュレータ (LM317) と呼ばれる、一定の電圧を安定して供給するICも搭載し、電圧変動によるICの破壊などが起こらないようにし、信頼性を向上させた。LED電源インジケータなども内蔵しており、モーターが回転

しないなどのトラブルの際、簡単に電源が来ているかなどの確認もできるようにしている。

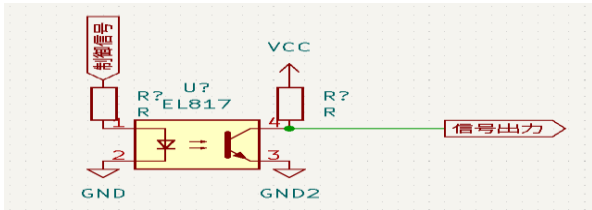


図4 フォトカプラ回路図

モーターのノイズがマイコン基板に入らないように、フォトカプラという電子部品を利用し、電氣的に絶縁を行っている。通常モータードライバなどを利用する際には、マイコン電源とモーター電源のGNDを接続することが必要である。しかし、モーターで発生したノイズがGNDを通してマイコン側電源に影響し加速度センサなどにノイズが乗ることがある。これを対策するためにGNDを切り離してもモーターの制御を行えるようにフォトカプラ(図4)を利用した。

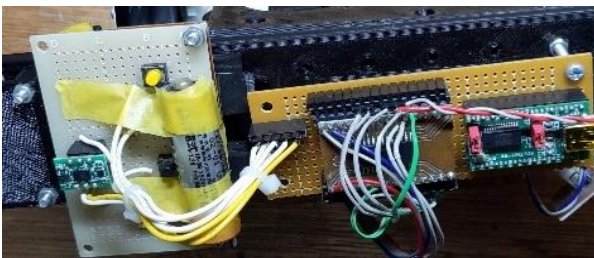


図5 マイコン基板部

マイコン基板部では、メインマイコンであるPIC16F18857を搭載している。このマイコンは、I/Oピンが多く、プログラムメモリと実際の実行プログラムを格納するメモリが多く(56KB)、大規模なプログラムも実行することができるマイコンである。そのほかにも内蔵モジュールとして様々な用途に使えるハードウェアタイマーが多くあり、タイマー割り込みのほか、方形波の周波数測定や、スリープからのウェイクアップなどにも利用できる。

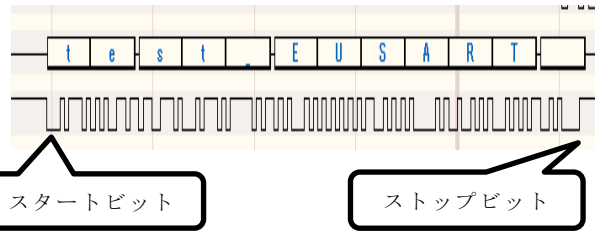


図6 USARTの波形

内蔵モジュールであるUSARTモジュールを調歩同期式で使用している。これは、図6を見てもわかるように最初にスタートビットが送られ今まで1だった信号が0になっている。これを利用して同期をとっている。そして最後の文字の送信後には、ストップビットと呼ばれる信号を送り伝送路を1にする。この信号をパソコンなどと容易に接続するために調歩同期式通信方式(TTL、CMOSレベル)の信号をUSBの信号に変換するためのICであるFT232RLが搭載されたモジュールも、ロボットに搭載している。これによりマイコン内でA/D変換された、アナログ値などを、Tera Termなどに表示させデバックを行いやすくしている。そのほかにも別のソフトを使うことで値を直接グラフにすることもでき、傾きなどの変化をわかりやすくすることもできる。

(4) 制御プログラムの製作

(ア)ソフトウェアは、MPLAB X IDE というPICマイコンの開発環境を利用した。

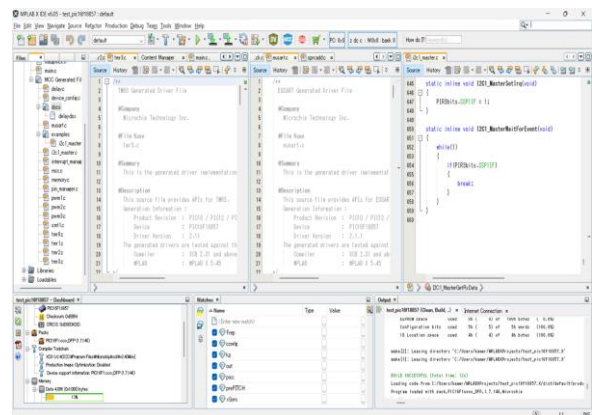


図7 MPLAB X IDE

この開発環境には周辺モジュールのレジスタを GUI で設定できる機能 (MCC) や、高度なデバッガなどが内蔵されている IDE である。今回は様々な内蔵モジュールを利用しているためこの MCC を利用した。

(イ) 車体の傾きの検出

ジャイロセンサ、加速度センサ、磁気センサの値の取得を行い、傾きを求めた。今回用いたセンサ (BMX055) は加速度センサ 3 軸 (X 軸、Y 軸、Z 軸)、ジャイロセンサ 3 軸、磁気センサ 3 軸がパッケージ内に入っている。このセンサは、I2C を用いてデータを受け渡しする。

I2C とは、ホットスワップ可能な通信規格で、一つの基板内のマイコン同士や、マイコンとセンサなどの通信など比較的、短距離の通信に利用される。アドレスさえ違えば複数のデバイスとの通信も行えるため、マイコンなどに接続するセンサなどにも多く使われている。

(ウ) カルマンフィルタ

加速度センサとジャイロセンサには得意不得意があり、どちらかのセンサ単体だと正確な車体の傾きをマイコンが認識することができない。その対策のためカルマンフィルタと呼ばれる、ソフトウェア処理を行った。

これは各センサの得意なところのみを利用するようなものである。

このフィルターは、時間変化を利用するため、マイコン内で正確な時間をカウントする必要がある。このカウント時間は、数マイクロ秒単位でカウントする必要がある、PIC マイコンに内蔵されているタイマーを複数組み合わせ、利用するなど様々な工夫が必要となった。

```
float Kalman_getAngle(float newAngle, float
newRate, float dt) {
    Rate = newRate - Bias;
    Angle += dt * Rate;
    P[0][0] += dt * (dt * P[1][1] - P[0][1] -
P[1][0] + Q_Angle);
    P[0][1] -= dt * P[1][1];
    P[1][0] -= dt * P[1][1];
    P[1][1] += Q_Bias * dt;

    float S = P[0][0] + R_Measure;
    float K[2];
    K[0] = P[0][0] / S;
    K[1] = P[1][0] / S;

    float y = newAngle - Angle;
    Angle += K[0] * y;
    Bias += K[1] * y;
    float P00_temp = P[0][0];
    float P01_temp = P[0][1];

    P[0][0] -= K[0] * P00_temp;
    P[0][1] -= K[0] * P01_temp;
    P[1][0] -= K[1] * P00_temp;
    P[1][1] -= K[1] * P01_temp;
    return Angle;
}
```

図 8 カルマンフィルタプログラム

(エ) モーターPWM制御

モーターの制御にはマイコンに内蔵されている、CCP モジュールを利用している。このモジュールは、3 つの機能があり、モーターの制御には PWM 機能を利用している。PIC マイコンの PWM 機能では、周期とデューティサイクルを調整することでモーターの電力を制御することができる。PWM はモーターの制御以外にもインバーターなどにも利用されている。ちなみに周期は 20kHz、分解能 10Bit で制御している。

(オ) モーター回転速度の計測

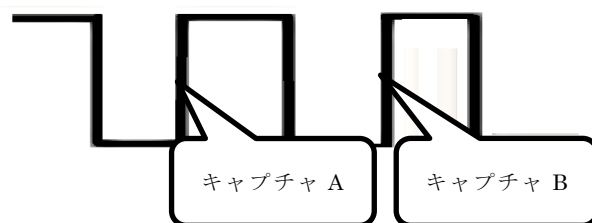


図 9 CCPモジュール

モーターの回転速度の計測には、CCP モジュールのキャプチャ機能を利用した。これは

波形のエッジごとに専用レジスタに、指定したタイマーの値を書き込むものである。今回利用したモーターにはエンコーダが内蔵されている。そのためそのエンコーダのオンオフタイミングを測定すると周波数が求まる。

具体的な動作はタイマーを同期モードで動作させている。図9のように波形が立ち上がった時のタイマーの値がまずキャプチャAに保存される。そのあと次のエッジでキャプチャBが保存される。キャプチャAからキャプチャBが立ち上がるまでの時間はタイマーで記録されているため、波形のエッジ間の時間を測定でき、周波数を求めることができる。今回使用した制御にはリアクションホイールの回転速度が必要になるためここから求めている。

(カ) PID 制御

姿勢制御ではPID制御を使用している。今回は、

- 積分ゲインにホイール回転速度をかけたもの
- 微分ゲインに角度の微分値をかけたもの
- 比例ゲインに角度をかけたもの

この3種類の式を利用して姿勢制御を行っている。この3種類の式が適切に影響しあうことで正確な制御ができる。

3. 研究のまとめ

今回、この自転車ロボットの作製を通して、PICマイコンの内蔵モジュールなどの知識を深めることができた。今まで使ってきたマイコンは、レジスタなどが少ないものが多く、簡単に設定し利用できていたが、今回利用したマイコンは非常に多くの機能があり、モジュールの設定に苦労した。ジャイロセンサ、加速度センサを正確に動作させるノウハウなどについても学んだ。電源周りの安定度の大切さや、外乱による加速度センサの値の乱れなどをいかに取り除き、正確な傾きを取得す

ることが大変だった。PID制御自体は情報が多く比較的作りやすかったが、カルマンフィルタなどは内部構造が複雑なためか解説がほぼなく製作にはほかのマイコン向けに作られたライブラリを改造するなどの方法も有効であることを学んだ。

参考文献

PIC16F18857 Data Sheet MICROCHIP

C言語によるPICプログラミング大全
後閑 哲也 技術評論社

電子工作のためのPIC16F1ファミリ活用ガイドブック
後閑 哲也 技術評論社

トランジスタ技術 2020年6月号 CQ出版