

本



ist

もくじ

表紙「好物は干し芋です！」

keko

何か～テトリスのやうなもの～

P.2 naru

エアコンコントローラの製作

P.6 naru

倒立振り子開発の途中経過レポート

P.10 a504to

何か～マトリスのやうなもの～

naru

小ピンのマイコンで、手軽に、能動部品なしに多くのLEDを制御することを考えました。手軽にということで回路が単純で鉛メッキ線とジャンパ線によるユニバーサル基板への片面配線が可能ということを目指しました。また、秋葉原のあるお店でシフトレジスタ74HC164の価格は160円です。これは昨今の安価なマイコンの価格に匹敵します。これを部品表から排除し、さらに安く作る工夫をしました。

一般的なドットマトリクスLEDの駆動回路

ドットマトリクスLEDモジュールは、複数のLEDが格子状に配線されてモジュールになっている部品です。

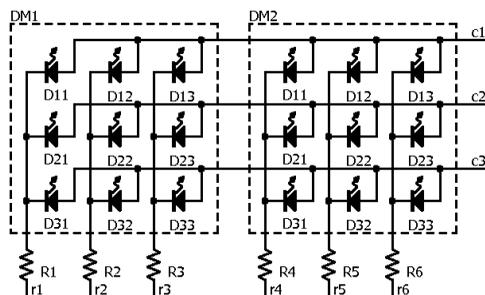


図 1:一般的な回路構成

図1のように、一般的なドットマトリクスLEDモジュールの駆動回路構成では $m \times n$ のドットマトリクスLEDモジュールを駆動するために $m+n$ 本の出力端子が必要となります。たとえばこの図1のように、 6×3 のドットマトリクスLEDの駆動に必要な端子の数は $6+3=9$ [本]となります。 m と n の値が大きくなり、貴重なマイコンのI/Oを割けない場合はシフトレジスタやラッチなどの能動素子を用いて出力端子を増やすことになります。

「何か」の秘密：LEDを逆に接続する

あるコミュニティの掲示板でこのような部品を知りました。(図2)

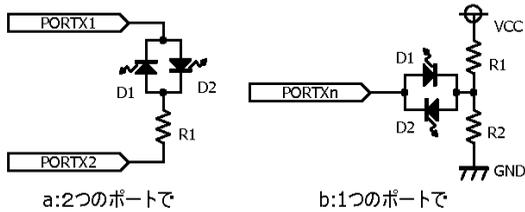


図 2:逆に接続された LED

図 2a のように2つのポートでドライブする場合、各ポートの状態と LED の状態の関係は表 1a になります。また、図 2b のように1つのポートでドライブする場合は表 1b になります。Hi-Z は 3 ステートポートを入力に設定したハイインピーダンス状態を表します。

| PORTx1 | PORTx2 | D1 | D2 |
|--------|--------|----|----|
| L | H | 点灯 | 消灯 |
| H | L | 消灯 | 点灯 |
| H | H | 消灯 | 消灯 |
| L | L | 消灯 | 消灯 |

a

| PORTxn | D1 | D2 |
|--------|----|----|
| H | 点灯 | 消灯 |
| L | 消灯 | 点灯 |
| Hi-Z | 消灯 | 消灯 |

b

表 1:図 2 のポート状態と LED 状態の対応

これをドットマトリクス LED に応用できないでしょうか。図 3 は単純にマトリクス状に並べてみました。こんなドットマトリクス LED モジュールは入手できません。図 3 の回路は図 4 のように変形することができます。

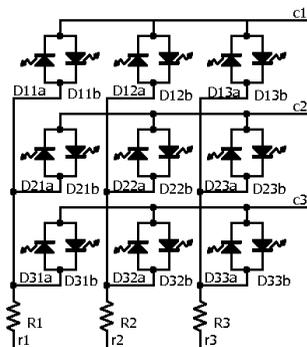


図 3:逆接続 LED を単純にマトリクス状にしてみる

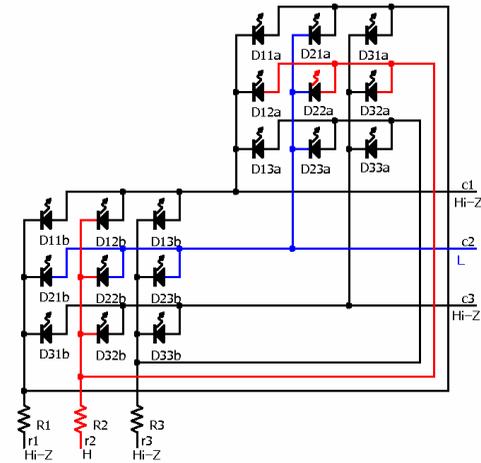


図 4:逆接続 LED のマトリクスを変形

これなら一般のドットマトリクス LED モジュールが使いそうです。

例えば D22a だけ点灯したいときは(r1,r2,r3)を (Hi-Z, H, Hi-Z)、(c1, c2, c3)を (Hi-Z,L, Hi-Z)にします。すると D22a だけが点灯して他は消灯します。r1, r2, r3 側で複数同時に H にすると行単位で表示できます。これでダイナミックスキャンすればいいわけです。

この問題について考えてみます。

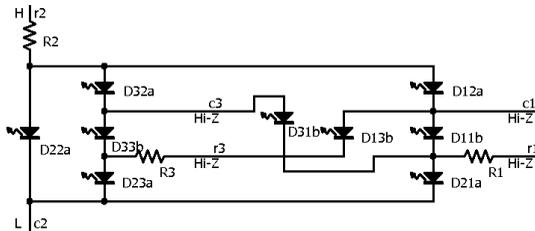


図 5:問題

Hi-Z の端子が余計な LED らの電流によって H と L の間の電位になり、余計な LED が点灯すると考えられます。余計な LED らは直列に接続されていることに注目し、目的の LED の順電圧が余計な LED らによって分圧されていると考えます。図 6 は赤色 LED の電圧電流特性です(地味に測りました:-)。これによれば、目的の LED の電圧が 1700[mV]のとき流れる電流は 12[mA]です。これに対して余計な LED らにかかる電圧は 1/3 の 566[mV]なので流れる電流は 0.1[mA]以下というこ

とになります。このコントラストにより、余計な LED らは点灯しているように見えません。

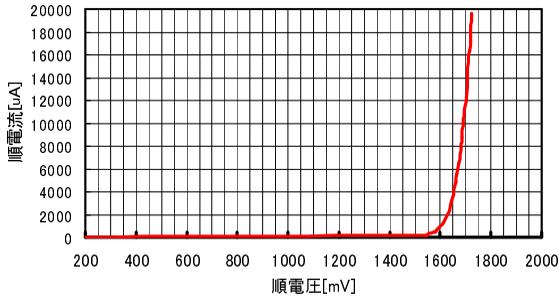


図 6:赤色 LED の電圧電流特性

さらに簡略化(ロウとカラムの番号対応は必要なく任意のロウとカラムを繋げればよい)して 8*8 のドットマトリクス LED モジュール 2 個をユニバーサル基板に実装してみました図 7 に配線例を示します。

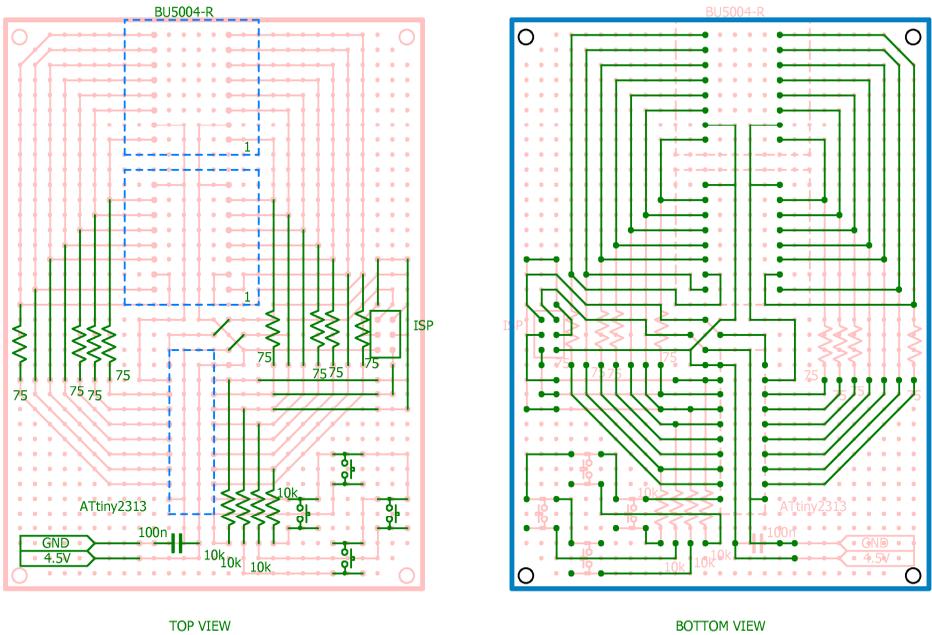


図 7:ユニバーサル基板への配線例

クーラーコントロールの製作

naru

立秋とは名ばかりのこの暑さなお厳しい暑さが続きます。残暑お見舞い申し上げます。

さて、学校の教室ではエアコンのコントロールが原始的で、エアコンを動かすと寒い、止めると暑いという状況になっています。その操作板を図 1 に示します。

L(low),M(mid),H(high)の3段階から出力を選択できるようになっているのですが、どの設定にしても部屋が寒くなるので、正しい解釈は

L:large(大げさ)

M:max(最大)

H:high(極端に高い)

などと言われています。そんなわけで、このクーラーにコントローラを付加してエアコンにすることを考えます。



図 1:クーラー操作板

RC サーボモータ

操作板を分解して配線を引き出したりすればリレーなどで簡単にコントロールできますが、公共施設の設備なのでそうもいきません。そこでここでは RC サーボモータの機械的につまみを回してコントロールします。

RC サーボモータといっても Radio で Control するわけではありません。Radio Control Car に使用されているサーボモータなので、一般に RC サーボモータと呼ばれています。

今回は千石で入手したミニスタジオの PR90 という RC サーボモータを使用します。¥1500 円と非常に安価で小型な RC サーボモータですが 1.6kg-cm のトルクを出せます。エアコンのつまみを回すには十分なトルクです。

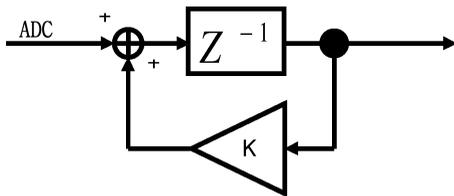


図 4:1 次 IIR フィルタ

室温の制御

図5に示すように入力と出力を対応させました。一般の on/off 制御を4段階に拡張したものです。Th はヒステリシス。Taim は目標温度です。

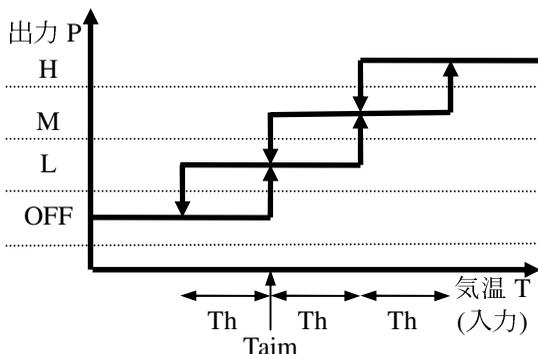


図 5 : 入出力の対応

on/off 制御なので気温は滅多に目標温度に収束しません。正確に温度

を収束させるには4値PWMして積分制御を行う必要があります。でもPWMするとRCサーボモータがガシヨガシヨうるさい(笑)ので今回はそこまでやっていません。

設置

図7に設置して動作している様子を示します。表示は左から室温、目標温度、ヒステリシス、です。下の段は0~3をoff, L, M, Hに対応させています。とりあえず両面テープで接着してみました。サーボの軸はつまみともちゃんとかみ合って働いてくれました。

5分程度の周期でガシヨンという音がしてクーラーが切れたり入ったりします。目論見どおりの動作です。ヒステリシスは部屋の容量にもよりますが、設置した教室では0.4度程度が丁度いいみたいです。秋までに暖房に対応したファームウェアを用意しようと思います。

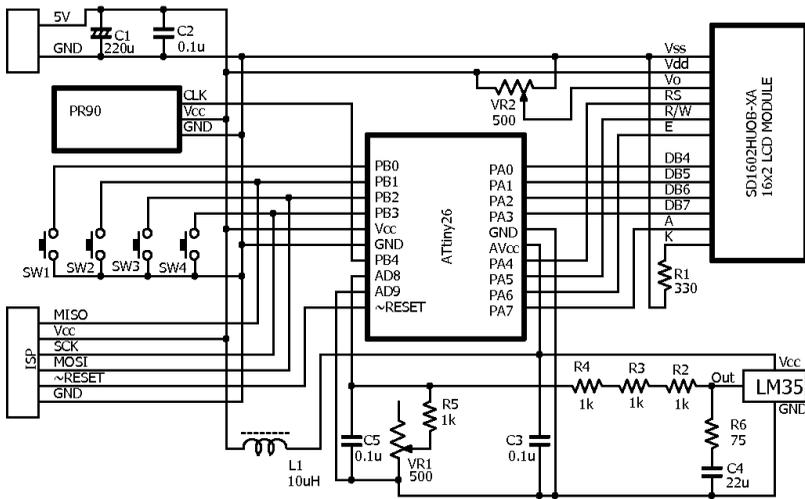


図 6:全体回路図

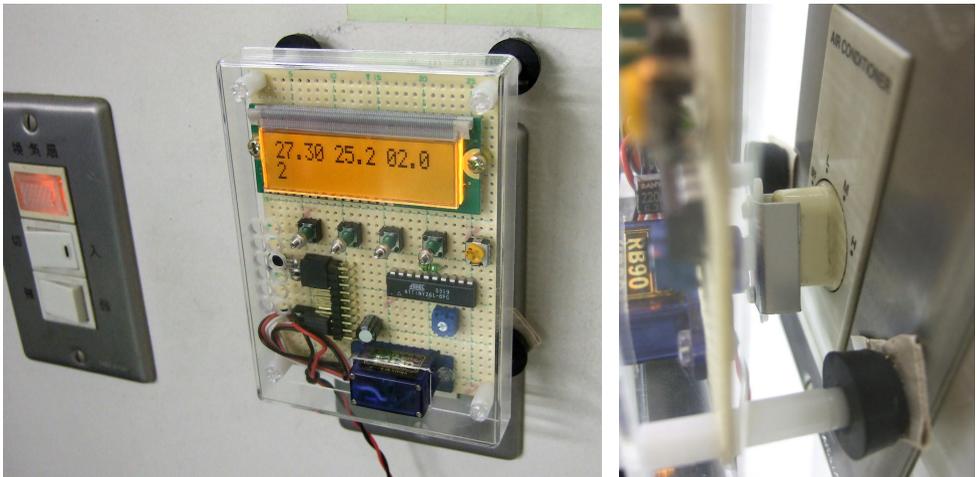
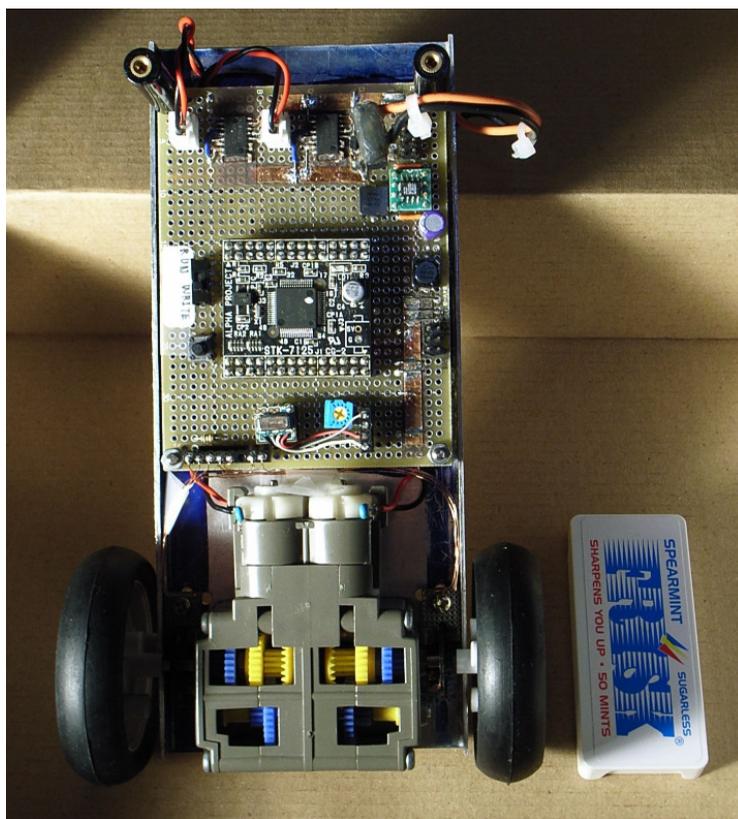


図 7:設置・動作している様子

倒立振り子開発の途中経過レポート

2007/08/11 hello, WORLD 開発部 a504to



1. 開発動機

マイコン性能の向上により、従来行えなかった高度な制御を小さなコンピュータで高速に行えるようになってきています。また、各種センサも安くなり、ゲーム機のコントローラなどにジャイロセンサ、加速度センサなどの従来高価だったセンサが使われています。これらをうまく使うことで安い倒立振り子を作りたいと考えました。本当は[1]が秋葉原の店に展示されていて、格好いいと思ったけど高くて買えないから作ることにしました。

2. システム概要

倒立振り子は現在の自らの傾きを検出してモータなどで姿勢を正します。このプロジェクトでは角速度センサ（レートジャイロ）から得られる角速度をオブザーバに入力することにより角度を推定します。

オブザーバとは状態フィードバック制御のときに使う手法で、制御対象をモデル化したものに入力を入れることで、センサで直接観測できない値などを推定します。詳しくは制御関係の本を見てください。これを使うと角速度から角度が出てくることがわかればいいと思います。倒立振り子は一般に慣性モーメントが大きいほど機械的な時定数が下がり、制御の周期を長くできます。しかし今回は高速なマイコンを使って制御の周期に関する条件を緩和できるので、小さくて軽いロボットに挑戦してみます。電池も単三電池二本でやってみます。

3. 部品

制御で使う角速度を得るためにジャイロセンサが必要になります。ジャイロセンサはプレステ3のSixAxisコントローラ(調べてみると4軸みたいですが)から取り出したジャイロセンサモジュールを使います。このモジュールには村田製作所のENC-03Rというジャイロ[2]と増幅・フィルタ回路がすでに実装されているので都合がいいです。

さらに制御を行うマイクロコンピュータが必要です。このプロジェクトでは(株)アルファプロジェクトのSTK-7125を使いました。STK-7125はルネサステクノロジ社製のSH7125と発信子、リセット回路が乗ったCPUボードで¥2940という安価な値段と小さいサイズでこのようなロボットを作るときに最適です。

トルクを発生するためにモータが必要です。将来自由自在に倒立を立たせたまま動かしたいので、左右のタイヤは独立で駆動します。今回は入手性がよくて安いタミヤのダブルギアボックスを選択しました。さらにタイヤもタミヤのナロータイヤセットを選びました。図らずもジャイロとタイヤは[1]と同じになってしまいました…。

さらにモータドライバとしては小型でFET構成のNECエレクトロニクス

の μ PD16805 を選びました。FET で構成されたモータドライバは今回のように 3V 程度の低電圧での動作のときには、トランジスタで構成されたものより有利に働くはずです。

電源を考えると単三電池二本ではマイコンが動きませんので昇圧することになります。今回は秋月で容易に入手できる MAX879 を使いました。最後にプログラムをマイコン内蔵 flash に書き込むときにパソコンからシリアル通信する必要があります。ここでは RS232 のレベル変換のために ADM3202 を使います。部品などは秋葉原をひとまわりすれば入手できるはずです。 μ PD16805 はツクモロボット王国で売っています。

4. 回路と実装

回路は 3. の部品をすべてつなぐだけです…。一応工夫したところは SH-7125 の書き込みの切り替えを単純なインバータを構成してスイッチ一つでできたことです。実装はモータ系と制御系の電源を分け銅箔で配線しました。二つの電源系は電池ボックスの端子で交わっています。一応安定して動作しているので、この配線方法は間違っていないかたみたいですが。マイコン基板の電源 LED は青にしておきました。ジャイロモジュールの使い方は電気通信大学ロボメカ工房[3]の方のレポートを参考にしました。ただ、今回は試作なので増幅率などはそのままいじっていません。最新版の回路図を載せません。間違いがあるかとは思いますがご自由にお使いください。

5. プログラミング環境

制御プログラムは C 言語で書きました。コンパイラは KPIT Cummins GCC[4]です。ルネサスの IDE である HEW の評価版に組み込めるので楽に開発できます。さらにサポートページが用意されているので疑問にぶつかったときなどに質問を KPIT Cummins 社の人にするのができ、まさにいたせりつくせりです。プログラムは文末に指定した場所で公開していきたいと思っています。突っ込みどころが満載のはずですか、よろしければごらんください。

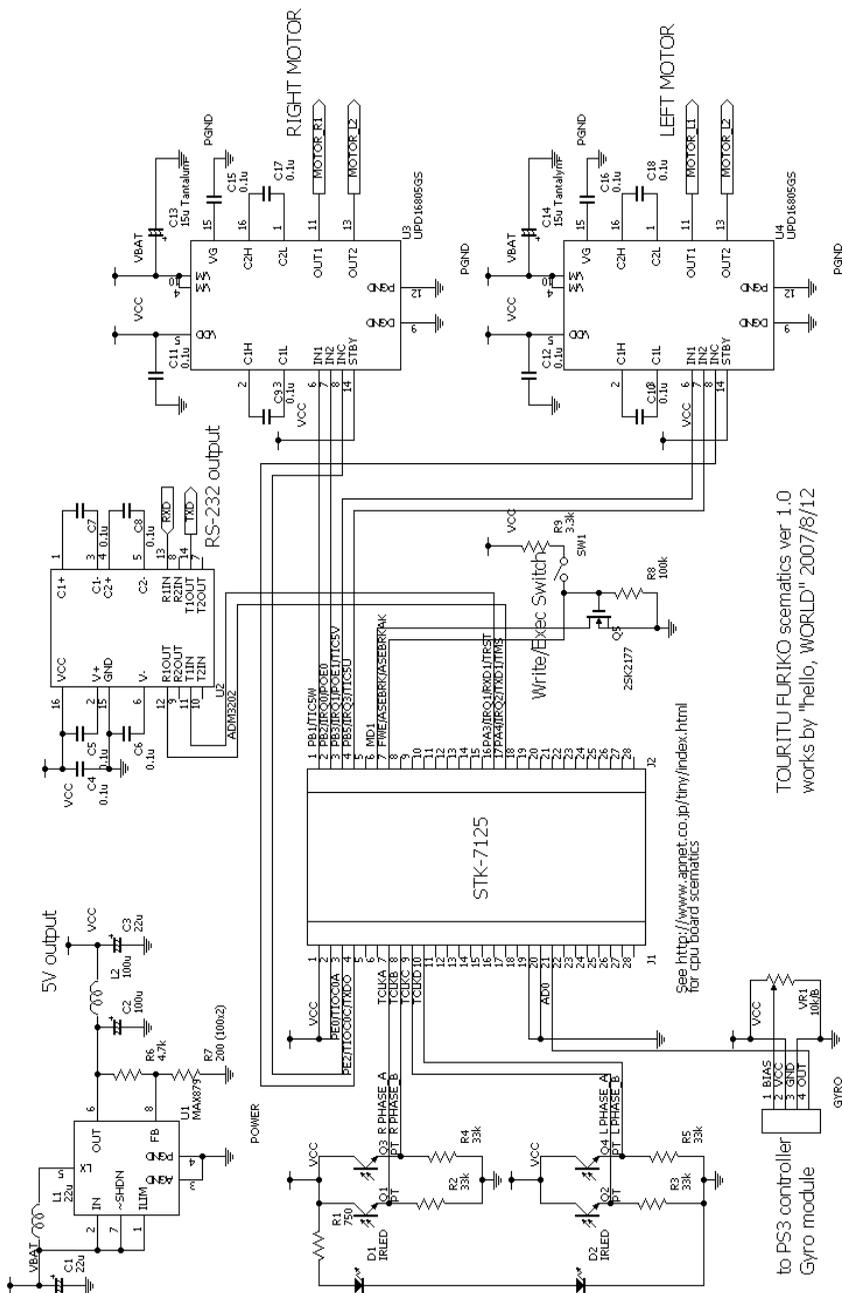


图 1 回路图

TOURITU FURIKO schematics ver 1.0
works by "hello, WORLD" 2007/8/12

6. 制御

制御は図2のブロック線図を使っています。ロボットの慣性モーメントなど測るのが難しいパラメータが出ますが、適当に定数をつけてみてもオブザーバの収束が多少遅れるだけなので問題ないみたいです。ただオブザーバの係数などが広い範囲（一つが1でもう一つが10万とか）にばらつくので、広い範囲をあらわせる型で変数を取ってください。私は `long long int (64bit)` でやりましたが、CPU パワーとの兼ね合いしいでは浮動小数点型を使ってもいいかも知れません。

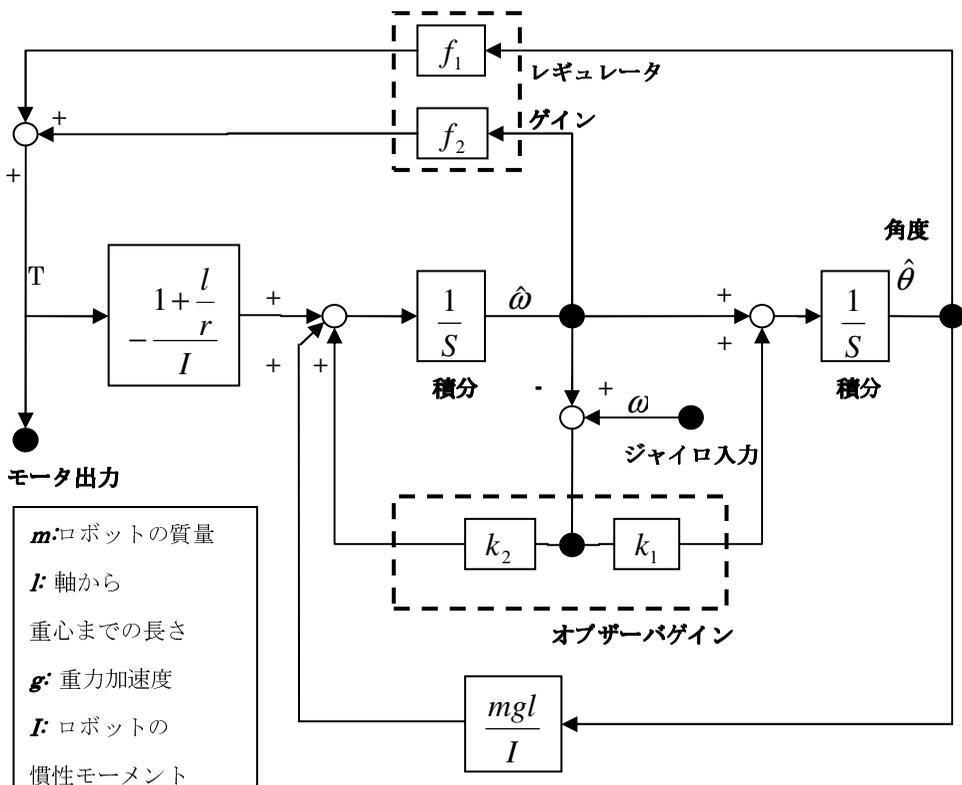


図2 ブロック線図

7. 結果 (途中経過)

一応立ちました！しかし、いまいち不安定な動作しかしません。たとえば机の端から端までもものすごい速度で駆け抜けていきます。どうやらロボットは自分の角度を正すことはするみたいですが、同じ位置にじっとしていることは無いみたいです。これではしょうがないので、ロータリエンコーダをつけて制御にロボットの位置情報を入れることを考えています。[1]では確かにロータリエンコーダらしきものが着いていました。このプロジェクトでも古い98のマウスから引っpegがしたロータリエンコーダをつけることにしました。ちょうどSH7125の内蔵カウンタユニットには位相計数モード(ロータリエンコーダの回転方向がわかるカウンタモード)が付いています。ロータリエンコーダをギアボックスからの出力軸につけていますが、現在いまいちカウントが安定しません。どうやらタミヤのギアボックスの軸がカタついて、フォトセンサとスリット円板の位置がずれてしまうようです。軸受けをつけるなどの対策が必要そうです。

8. まとめと今後の予定

このプロジェクトは安価な倒立振り子ロボットを開発することを目標として、以下のことがわかりました

- ゲーム機などのジャイロセンサは十分倒立振り子に使える。
- モータは[1]ではRE-260タイプを使っているが、二つであればFA-130(ミニ四駆のモータ)でも十分。
- バッテリーは二本で十分そうである。
- CPUはSH2@48MHzで十分あるようである。
- オブザーバのプログラムでは表せる値の範囲の大きい変数型を使わないとオーバーフローする。
- 角速度以外に、ロボットの位置の情報が無いと、ロボット自体が平行移動していつてしまう。対策が必要である。
- デバッグ方法を考えないといけない。値をシリアルで転送、表示すると制御周期がずれてしまう。表示値を間引く、空き時間に転送する、CPU内

蔵のデバッグ機能を活用するなどの対策がある。

今後さらにロボットを改良していく必要があります。具体的にはロータリエンコーダのカウントをちゃんとしてから、制御にそれを組み込む必要があります。さらに、現在角速度は適当な移動平均を取って使っているだけです。もう少し高性能なデジタルフィルタを構成できるとさらにより精度の制御ができそうです。一番有効そうなのがオブザーバの各パラメータの最適化です。今は適当に走らせてみてよさげな値を入れているだけですがもしかしたらシミュレーションなどで最適なパラメータを得られるのかも知れません。制御理論はよく解らないところがありますが今後勉強していきたいです。

倒立振り子の顛末は皆様に追って報告していきたいと思っています。つまらない文になってしまいましたが、少しでも皆様の興味を引くことができたのであれば幸いです。

開発ブログ(あまり更新していません):<http://a504to.blogspot.com/>

参考文献

[1] 北斗電子倒立制御学習キット PUPPY

<http://www.hokutodenshi.co.jp/7/PUPPY.htm>

[2] 圧電振動ジャイロ “ジャイロスター”

<http://www.murata.co.jp/catalog/s42j.pdf>

[3]電通大マイクロマウスぶろぐ 内レポート「PS3 ジャイロ、使ってみる？」

<http://www.rmkoebou.mce.uec.ac.jp/contents/Report/mouse/blogmouse2/diary.cgi>

[4]KPIT GNU Tools

<http://www.kpitgnutools.com/>

本 1st

初版 2007年8月17日

PDF版 2008年1月5日

発行 なる研／Hello,WORLD!!

<http://naruken.cweb.tk/c72>

編集・発行責任者 naru

表紙 keko

<http://pipipiclub.hp.infoseek.co.jp/>

