

Arduino を使用した鉄道模型の自動運転

太田 圭祐

指導教員 谷野 一忠

要約

近年、マイクロコンピューターが急速に普及し、より簡単に複雑な電子工作ができるようになった。今回は初心者でも簡単に扱えるマイコンボードとして有名な Arduino と言われるものを用いて鉄道模型の自動運転をしようと試みた。

Abstract

In recent years, micro computers have spread rapidly, and we could have done electronic work more easily. So, I tried to do automatic operation of model railroad using Arduino.

キーワード

Arduino (アルデュイーノ), 鉄道模型

Keywords

Arduino, Model railroad

1.背景・研究目的

近年は身の回りにある家電製品や、プリンター等といった周辺機器のようなインターフェイスや汎用性をあまり必要としない物にはマイコンと呼ばれる小型のコンピュータが採用されるようになった。このマイコンの利点として、プログラムの改変によって、機能の変更が可能であるということが挙げられる。これにより、開発コストが削減され、多用される要因のひとつとなった。このワンチップマイコンの一種である Arduino は、非常に安価で、またコンパクトであるという特徴を有している。したがって、あまり複雑な制御をおこなわない場合であれば、鉄道模型の制御装置などにも有効に利用できる可能性があると考えられる。そこで本研究では、Arduino を鉄道模型の自動運転装置に用いることを試み、性能を検証する。

2.開発環境

今回、私が研究の対象を Arduino とした。

現在流通しているマイコンには Microchip Technology

社の PIC, 日立製作所の H8 シリーズ, Atmel 社の AVR マイコンなどがあり、それぞれ仕組み、開発環境などが異なる。その中でなぜ Arduino を選択したのかというと、マイコン(ATmega328P)を最初から搭載した完成品の基盤が市販されており、ライター(マイコンにプログラムを書き込む装置)が不要、開発環境は無料という手軽さで始められることが大きな要因となった。

Arduino の開発に”Arduino 1.0.5”というソフトを利用した。

Arduino のプログラミングには C/C++風の “Arduino 言語”を使用する。プラットフォームに依存しない開発と配備を行うことが可能である。

マイコンのプログラムは、テキストエディタにプログラムを入力し、それをコンパイルしてから Arduino のマイコン本体に書き込む。バグや不具合を確認したらその都度元のテキストエディタを編集する。

しかし、この作業は一つ一つの手順が分かれており、バグがあった場合、最初からの手順を行うこととなる。その効率を向上させるために開発環境として一連の作業

の流れが一つのソフトで行われる総合開発環境が多く使われている。その中でこの”Arduino 1.0.5”というソフトはコンパイル時にバグ・不具合を発見したらその部分をテキストエディタ上で指摘するのでバグやミスの確認から修正への流れがスムーズに行える。そのため、今回プログラムを開発するソフトとして使用した。

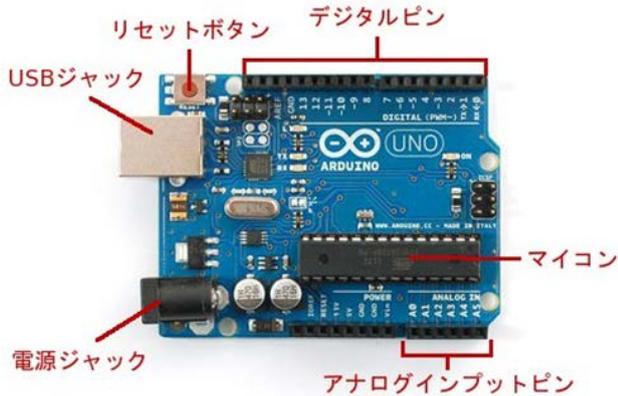


図1 Arduino 本体

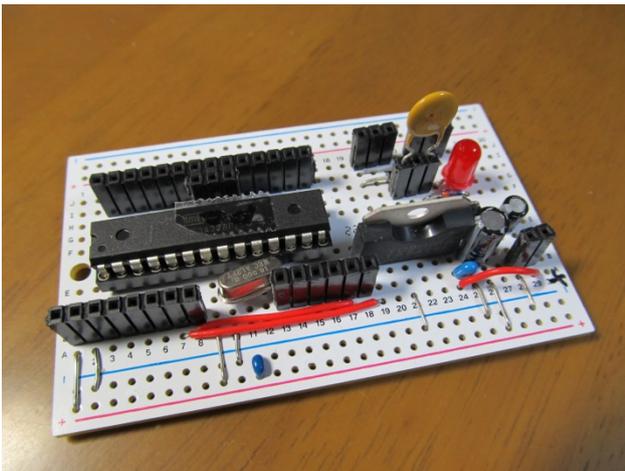


図2 今回製作した実験用基板。

3.鉄道模型

今回の研究では鉄道模型を走らせる線路に TOMIX 社の N ゲージ、車両を同社製のキハ 22(M 車)を使用した。

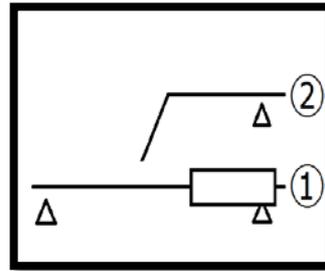
4.線形

線形は電動ポイントを筆頭に片側へ二本の引き込み線、もう一方は一本の引き上げ線とした。

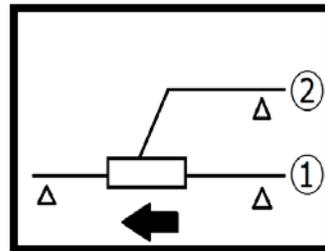


図3 上から見た実験用路線

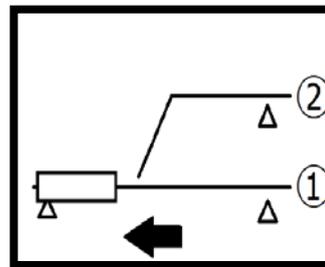
5.運転の手順



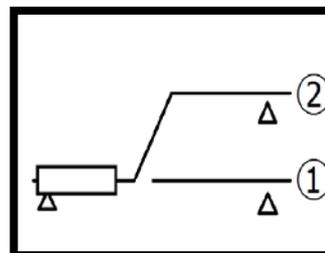
①△はセンサーである。まず、電車が一番線に待機している。



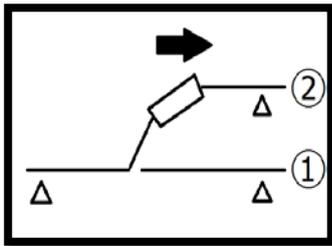
②一番線から引き上げ線へ電車が進行する。引き上げ線のセンサーが感知するまで電車は進行する。



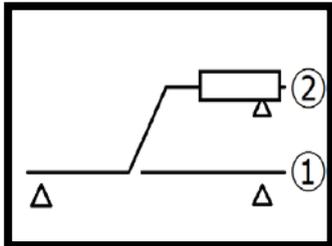
③引き上げ線のセンサーが電車を感知すると電車は停止する。



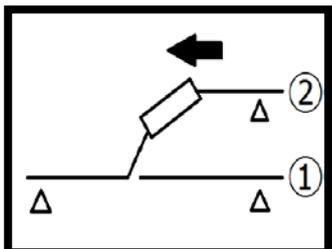
④電動ポイントを2番線に切り替える。



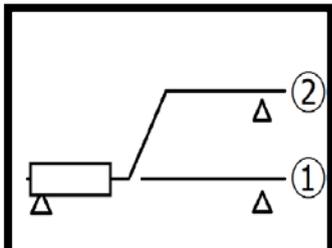
⑤二番線へ電車が進行する。二番線のセンサーが感知するまで電車は動き続ける。



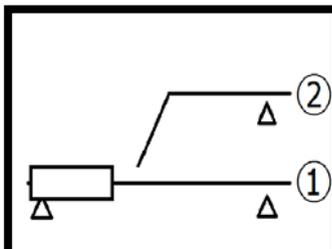
⑥二番線のセンサーが電車を感知すると電車は停止する。



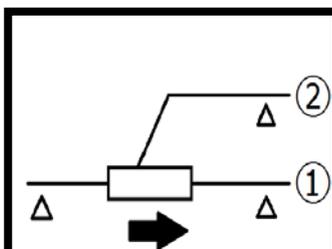
⑦数秒後に再び引き上げ線へ進行する。引き上げ線のセンサーが電車を感知するまで電車は進行する。



⑧引き上げ線のセンサーが電車を感知すると電車は停止する。

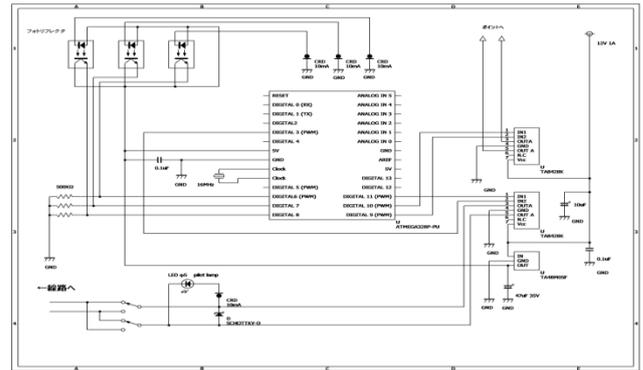


⑨ポイントを直線側に切り替える。



⑩一番線へ進行する。センサーが感知するまで電車は動き続ける。そして①へ戻って同じ処理を続ける。

5.回路図



6.鉄道模型感知用のセンサー

今回の研究では、車両感知用センサーにフォトリフレクタ(反射タイプ) TRP-105Fを使用した。フォトリフレクタは赤外線 LED から放出された光を物体に反射させ、フォトトランジスタで受光することで出力電流が変化するセンサーである。寸法が 3mm×3mm、検出距離が 1mm から 10mm 程度で今回の研究に用いるセンサーとしては理想的だと判断してこれを使用する運びとなった。

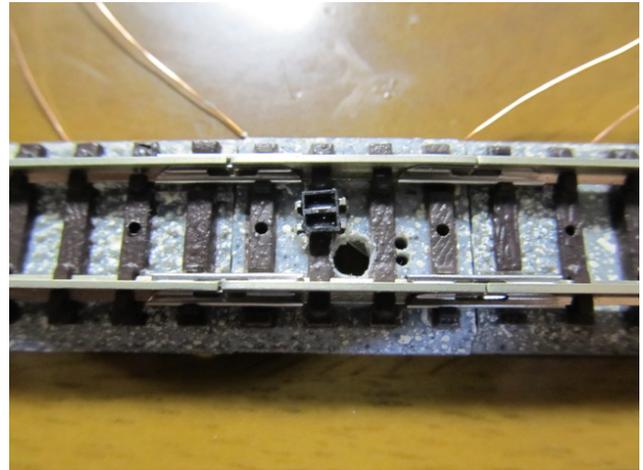


図4 線路上に設置したフォトリフレクタ

7.鉄道模型の駆動方法

今回の回路では、直流 12V から鉄道模型駆動用の電力(12V)と“レギュレーター TA48M05F”を使用し Arduino 駆動用の 5V を取り出す。

Arduino で鉄道模型を駆動するためにはアナログ出力関数である“analogWrite(pin, value)”を使用する。ATmega328P ではデジタルピン 3,5,6,9,10,11 で analogWrite 関数がサポートされており、次にそのピン

に他の関数が実行されるまで、アナログ値(PWM 波)が出力される。PWM 信号の周波数は 490Hz である。

【パラメータ】

Pin: 出力に使うピンの番号

Value: デューティ比(0 から 255)

0 で 0V, 255 で 5V が出力される。

【車両を走らせるスケッチ】

```
{
  analogWrite(11,127); //2.5V 出力
  digitalWrite(3,LOW); //3 番ピン入力
}
```

しかし、この ATmega328P では出力される電圧、電流は最大 5V 200mA である。それに対して鉄道模型の駆動には電圧 12V、電流は最低 500mA 程度必要であり、マイコン単体の出力では鉄道模型の駆動には力不足である。そこで“モータードライバ TA8428K”を使用し、0~5V を 0~12V に変換して駆動させた。

【鉄道模型駆動の流れ】

ATmega328P からの出力(0~5V)



モータードライバ(TA8428K)で変換



鉄道模型用電力 出力(0~12V)

8.ポイントの切り替え

ポイントの切り替えにはデジタル入出力関数である“digitalWrite(pin, value)”を使用した。

この関数で指定したピンに HIGH(5V)又は LOW(0V)に設定が可能である。しかし、今回使用した TOMIX 製の電動ポイントは 12V 150mA(メーカー公称値)で作動するため、やはり力不足である。そこで前者で鉄道模型の駆動に使用したのと同様のモータードライバを使用し、ポイントの切り替えを行う。

【パラメータ】

pin: ピン番号

value: HIGH か LOW

【ポイント切り替えのスケッチ(9 と 10 ピンにポイントを接続した場合)】

```
{
  digitalWrite(9,HIGH); //ポイントへ 12V 供給
  delay(200);           //0.2 秒待つ
  digitalWrite(9,LOW); //ポイント 0V 供給
  digitalWrite(10,LOW); //ポイント 0V 供給
}

digitalWrite(9,HIGH);の 9 を 10 にすると逆方向へ切り替わる。
```

9.結論と課題

今回の実験では Arduino を使用して鉄道模型の自動運転が可能であることが明らかになった。Arduino はパソコンとも簡単に連携が可能なのでパソコンからの運転も可能であることが期待される。

しかし、今回センサーとして用いたフォトリフレクタが他の光(蛍光灯やリモコンの赤外線、自分の体などに反応して誤作動してしまうことが多々発生した。また、レギュレーターやモータードライバの発熱が著しかった。

10.今後の課題・目標

- ・フォトリフレクタに代わるセンサーの調査
- ・レギュレーターとモータードライバの放熱対策

11.謝辞

今回の研究でお世話になった谷野先生と C 言語を御教授頂いた森先生に深く感謝を申し上げます。

12.参考文献

- ・楽しい電子工作 Arduino で電子工作をはじめよう!
著 高橋 隆雄
- ・Arduino Wiki 武蔵野電波氏

<http://www.musashinodenpa.com/wiki/>