

# DCC デコーダーの製作

中田 宏

## 趣旨

今回の記事は、トラ技通信 No2 で製作した DCC コントローラーの続きです。DCC とは、電力線伝送でデジタル信号を送り、一つの電源から複数の鉄道模型を独立して制御する方法でした。今回は、電源すなわちエンコーダーを製作して、デコーダーは鉄道模型会社の製作するものを実際の模型に組み込んで使用しました。単に鉄道模型を制御するだけならば、前回で完結しているのですが、実際にレールを流れる信号を捕まえてみたくて、デコーダーも製作しました。

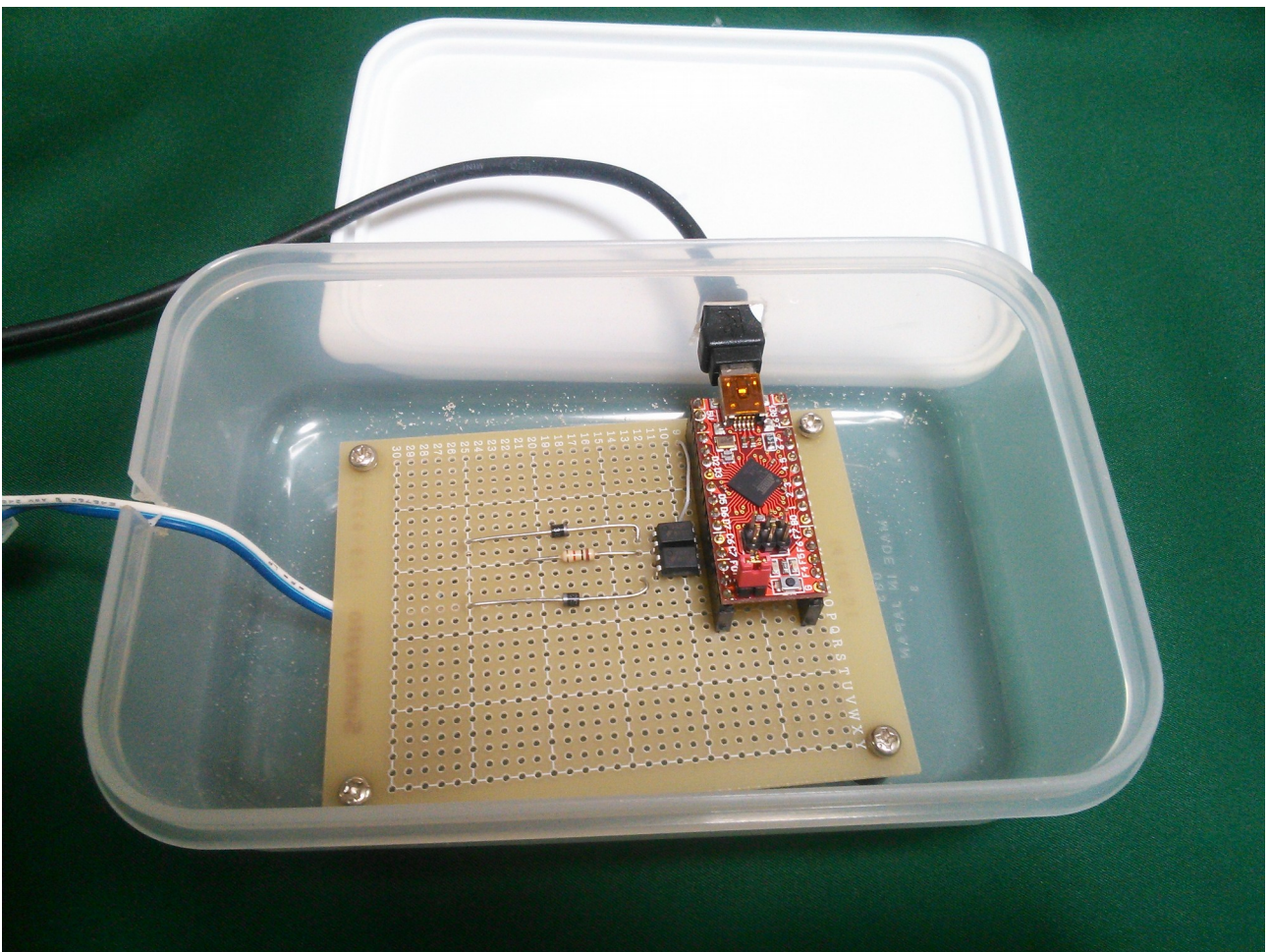


写真1 製作例

## 設計

今回は、信号をデコードすることを目標にします。レールから電力を取り出してモーターを駆動することはありません。実際に電力を取り出すには、ダイオードブリッジを使えば DC12V が簡単に取り出せます。あくまで信号制御の一貫として信号のデコードをメインに考えました。

表示には、パソコン画面を使用します。デバイスとしては、USB 接続のマイコンの GPIO

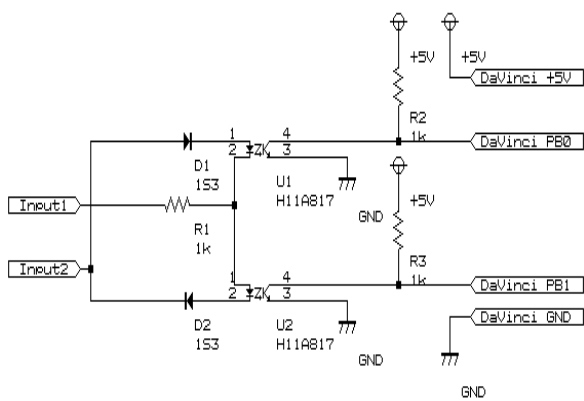
を使って、信号を取り出すことにします。信号の取り出し方は、簡易的には、+または-のどちらか片方のパルス長を調べれば充分です。今回は実験の意味もあり、単純なデコーダーというよりは、プロトコルアナライザに近いものが作れないかと考えました。よって、±のパルスとスイッチング時の 0V 時間を測れるハードウェアを企画しました。

レールに流れる±の信号を、フォトカプラで USB マイコンの GPIO に取り込みます。USB マイコンは、+のパルスが出ている時間、-のパルスが出ている時間、どちらのパルスも検出できない時間をタイマでカウントします。

カウント結果は DCC 信号にデコードせず、なるべく生に近いデータとして、USB 経由でパソコンに送ります。受け取ったパソコン側で DCC のデコードを行っています。

## 機器構成

回路図を示します。



レールに接続した入力端子から、+検出用のフォトカプラと-検出用のフォトカプラに接続します。フォトカプラの受け側は、12Vでは電圧が高すぎて直接受けられないので、電圧を下げるための抵抗が直列に入っています。フォトカプラと抵抗のほかにダイオードが入っている理由は、誤って高電圧を端子にかけても、フォトカプラにダメージを与えないた

めです。

USB マイコンには、またまたストロベリーリナックス社の DaVinci を使用しました。フォトカプラの出力は、USB マイコンの GPIO に直接つながり、USB マイコン外部で電源に抵抗経由でプルアップされています。USB マイコン内部にも GPIO プルアップ用の抵抗があるのですが、抵抗値が高すぎてフォトカプラの反応が鈍くなったため、外付けの 1kΩ 抵抗を使用しています。

## プロトコル

このデコーダーで一番苦労した点が、USB に流すプロトコルの決定でした。12Mbps のインタラプト通信でパソコンにデータを送っています。インタラプト通信なので、1/1000 秒に 1 回 64 バイトの packets を送ることができます。

DCC 信号は、最高速で 9kHz 出せますので、64 バイトの packets 内に 9 回の±のパルス時間が入る可能性があります。1 回の±のパルスについて、以下の 4 種類の時間情報を送りたかったのです。

- +のパルス時間
- +から-に移行する間の 0V 時間
- -のパルス時間
- -から+に移行する間の 0V 時間

64 を 9 で割ると上記 4 種類のデータを送るのに 7 バイトしか使えません。時間精度は 16bit を保証したかったので、圧縮しないと全データを送ることができません。ここは妥

協して二つの 0V 時間をパソコンに報告しないことに決めました。

### デバイスドライバ

試作品は、まず Windows で動かすことにしました。ここで気になるのが、1/1000 秒毎に送られてくるデータを、アプリケーションがポーリングすることで取りこぼさないだろうかということでした。特に OS の負荷が高くなってくると、ポーリング周期が長くなって、データを取りこぼす心配があります。そこで、簡単なデバイスドライバを書いて、ドライバの中に数 k バイトバッファリングすることにしました。

### アプリケーション

今回作成した Windows アプリケーションは、ドライバからデータを受け取って、DCC をデコードします。さすがにパソコンの CPU は充分早く、デコードに失敗したことはありません。細かい制御が必要なわけではないので、コマンドプロンプトから起動しテキストデータを出力する仕様としました。

### 結果

前回作成した電源からレールに信号を流し、鉄道模型が走っている状態で信号をモニタしてみました。模型のモーターからノイズが出て一部デコードに失敗するかもしれないと予測していたのですが、予想を裏切って信号を正確にデコードできています。