〔研究ノート〕

電子工学実験におけるライントレーサのデジタル化検討*

伊藤　順治*2，原田　恒迪*3

An Examination of Digitization of Line Tracers in Electronic Engineering Experiments*

Junji ITO*2, Hisamichi HARADA*3

*2, *3Department of Mechanical and Electrical Engineering, School of Engineering,
Nippon Bunri University

１．はじめに

近年，義務教育へのプログラム教科導入などプログラミングスキルを早期に身に付ける事が重要な課題となっている。本学，機械電気工学科においては１年次後期に「C プログラミング入門」で基本的なプログラミングのコーディング演習を行うが，その後の講義，演習，実験において，本格的にプログラミングスキルを向上させるプログラムはほとんど無いと言わざるを得ない状況である。一方で就職先の企業の業務内容は従来の物理的なもの作りだけではなくその機器を制御するプログラムまで作成し，ハードが分かるプログラミング技術が強く求められている。本研究ノートでは電子工学実験で広く使用されているアナログ制御型ライントレーサを用いて，その回路動作をシミュレーターにより再現し，原理を理解すると同時に，フィードバック制御を行う回路部をマイコンに置き換えることによりデジタル化する。実験および作成したプログラムにより，ライントレーサのデジタル化を実現したのでその実験内容および結果について報告する。

２．実験方法

・実験１－１

まず初めに従来のアナログライントレーサ（図１）について動作原理を確認するために keysight 社の Ad-

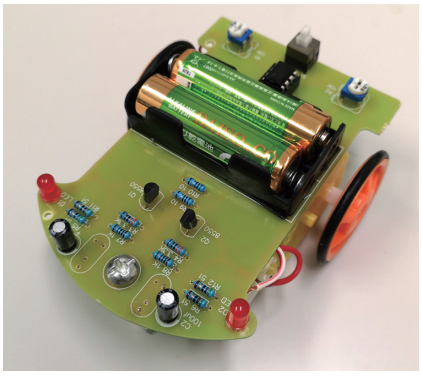

図１　アナログライントレーサ

*2022年６月15日受理
*2日本文理大学工学部機械電気工学科 教授
*3日本文理大学工学部機械電気工学科 学部生

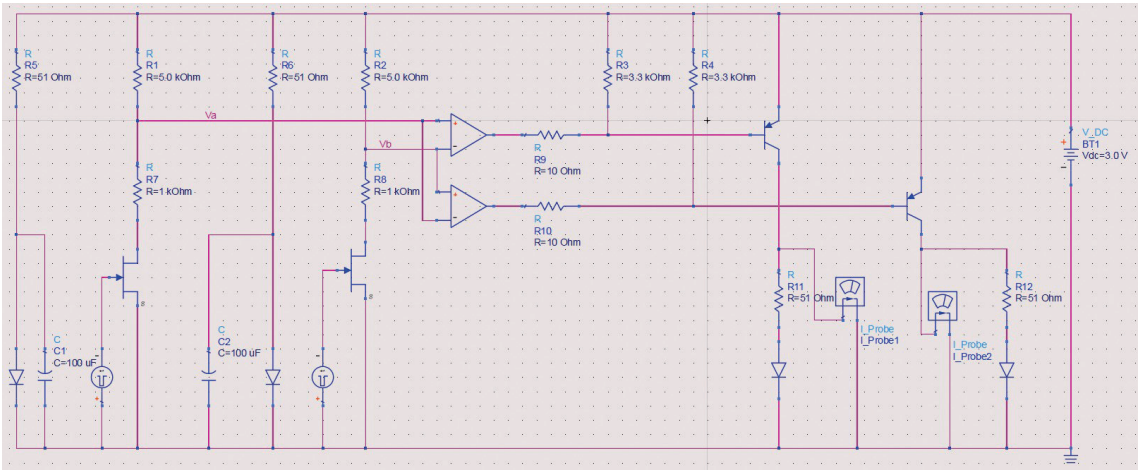

図２　ライントレーサシミュレーション回路

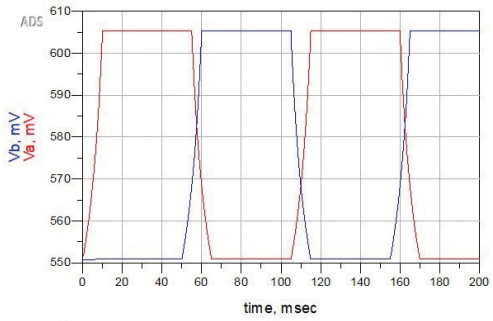

図３　コンパレータ入力波形

vanced Design System2022を使用してシミュレーションを行った。作成した回路を図２に示す。この回路の動作原理を簡単に説明する。

左右のLEDの反射光を光電素子フォトレジスタCdSにより抵抗値の変化に変換し，それを電圧変化量としてデュアルコンパレータによって比較する。図２の回路ではCdSの代替としてJ-FETを使用して抵抗値を可変した。ラインは幅18mmの黒線であることからセンサがライン上にある時には反射光は減少し，ラインから外れると反射光は増加することから，左右のセンサの位置がライン上部かそうでないかを判別する。その結果からライン上にある反対側のモーターを動作させることによりラインをトレースすることが出来る。図３にエミュレートしたコンパレータに入力される電圧波形を示している。具体的には方形波における Rise Time/Fall Time 等のパラメータによって設定した。この入力を理想オペアンプで構成したコンパレータに入力し，２つのモーターを駆動する電流をシミュレーションした。結果を図４に示す。

図４に示すように駆動電流が交互に ON/OFF している事が確認できた。

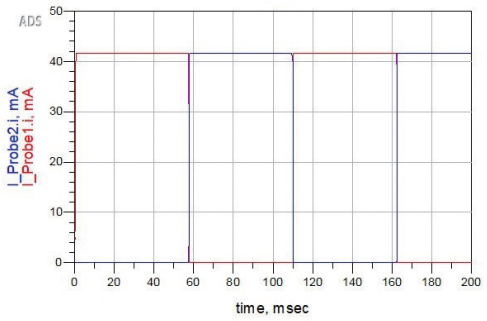

図４　モーター駆動電流シミュレーション結果

・実験１－２

次に本実験の目標であるデジタル化を行うが，実際にはコンパレータを Raspberry Pi Pico[1]（以降 RP２）に置き換える。作成した実験機を図５に示す。

図５では電池ボックスの前方に RP２を追加配置し，他の回路と干渉しないように設置した。

次にデジタル化した場合のシミュレーションを行った。具体的には回路のコンパレータの代わりに3.3V の方形波信号源２台を接続した。シミュレーション回路を図６に示す。方形波の設定では一定時間クロスする設定を行った。これは直進制御信号を想定したためである。

想定した入力方形波とモーター駆動電流のシミュレー

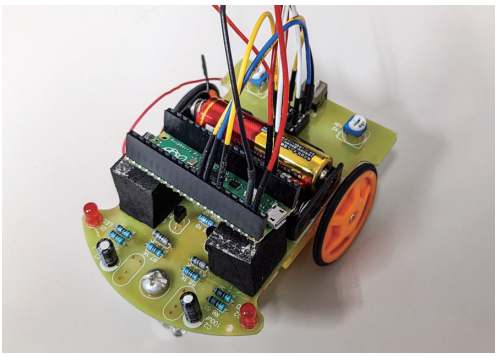

図５　RP２を搭載したライントレーサ

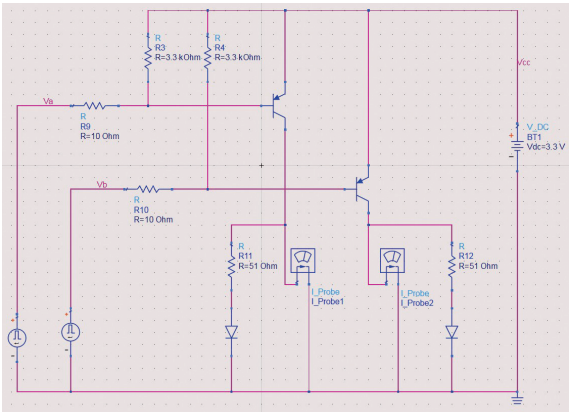

図６　方形波信号源でのシミュレーション回路

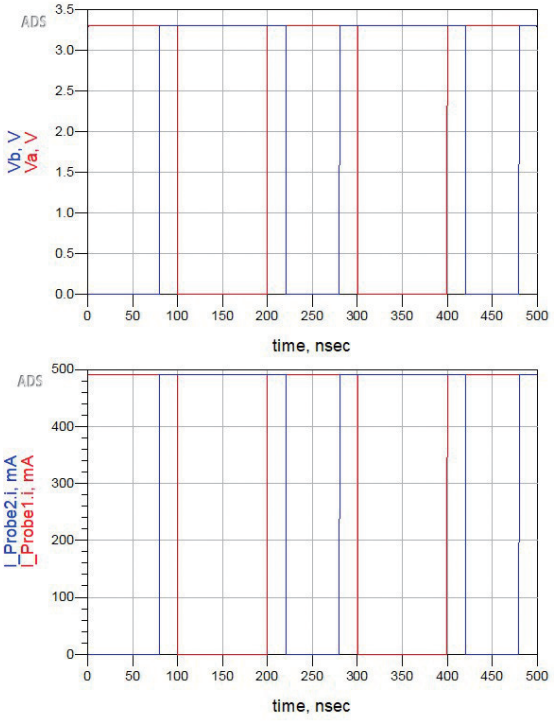

図７　デジタル化した時のシミュレーション結果

ション結果を図７に示す。入力信号V_a，V_bに同期してモーター駆動電流が出力されていることが分かる。

・実験１－３

次に，光電素子を制御する際の指標とするためガイドライン中心らの距離によるCdSの抵抗値変化を測定した。具体的には白紙上に幅18mmの黒ビニールテープを貼り（図８），テープ中心を０mmとしてCdSを左右17mmまで１mmずつ動かし，DMMを用いてV_{CC}とV_R，V_Lを測定する。得られた値と$R_{1,4}=5k\Omega$，$R_{2,5}=1k\Omega$として式（１）に代入しフォトレジスタ抵抗値$R_{3,6}$を算出した。

$$R_{3,6}=\frac{V_{R,L}}{V_{CC}-V_{R,L}}\times R_{1,4}-R_{2,5}[\Omega]\ \cdots(1)$$

右CdSを０mm～17mmスライドさせた場合の左右CdS抵抗値変化を図９に示す。

赤の折れ線グラフは右CdS抵抗値を示し，青の折れ線グラフは同じ位置での左CdS抵抗値を示している。

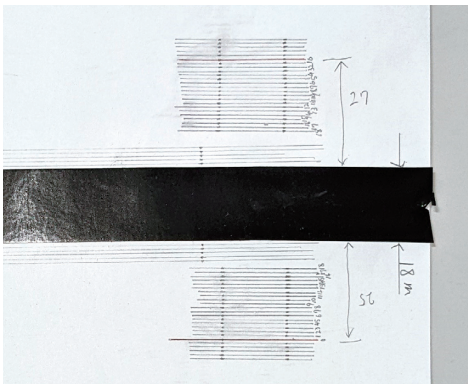

図８　試験用ライン台紙

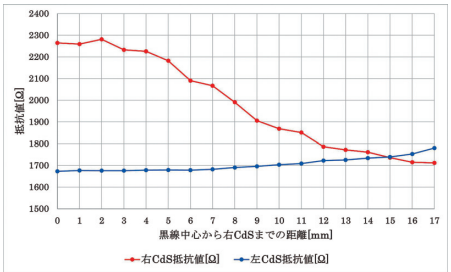

図９　右CdS基準でスライドした場合の抵抗値変化

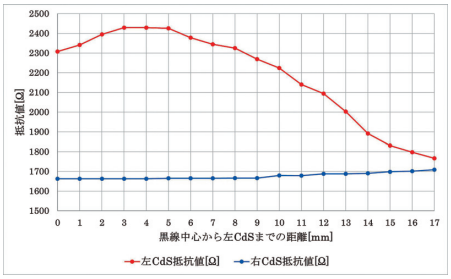

図10　左 CdS 基準でスライドした場合の抵抗値変化

赤の折れ線グラフは左 CdS 抵抗値を示し，青の折れ線グラフは同じ位置での右 CdS 抵抗値を示している。

3．プログラム作成

実験１－３で得られた測定値を元に作成したプログラムを図11, 図12に示す。

本プログラムは，CdS の入射光が弱くなる（黒線に近づく）と抵抗値が大きくなる性質を利用した。初めに，V_R, V_L, V_{CC} を RP２の ADC を用いて読み取る。次に，$R_{1,4} = 5k\Omega, R_{2,5} = 1k\Omega$ として式（１）によって CdS 抵抗値 $R_{3,6}$ を求め，その大小を比較する。$R_3 > R_6$ の場合は R_3 を基準 R_b として $R_{3,6}$ の差の割合 Δf を式（２）によって求める。反対に，$R_3 < R_6$ の場合は R_6 を基準値 R_b として Δf を算出する。$R_3 > R_6$ かつ $R_3 < R_6$ でない場合は処理を終了する。

$$\Delta f = \frac{R_a - R_b}{R_b} \times 100[\%] \quad \cdots(2)$$

次に，図９から右 CdS の位置が11mm の場合に右 CdS が左 CdS より大きい事が明確に判断できるため，その位置での $\Delta f = 8[\%]$ を基準に右左折・直進を判断する。例えば，$R_3 > R_6$ の時 Δf が８％未満の場合はラインに対して車体が大方平行であると判断し直進する。８％以上の場合は左方向に向かっていると判断し右折する。また，$R_3 < R_6$ の時 Δf が８％未満の場合はラインに対して車体が大方平行であると判断し直進する。８％以上の場合は右方向に向かっていると判断し左折する。

4．実験結果

プログラムを実行したところ，ライントレーサは直線区間では直進し，カーブ区間は小刻みに右左折することなく走行できる事を確認した。しかしながら，直線区間において安定して直進するまで発振がみられたため，パラメータの調節や PID 制御の導入を検討している。

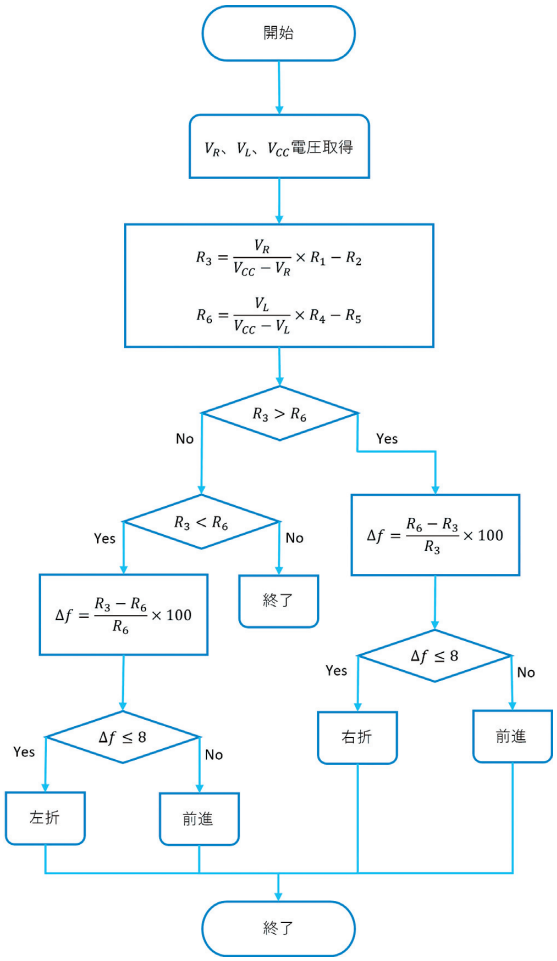

図11　プログラムフローチャート図

参考資料

[１]Gareth Halfacree, Ben Everard, “Get Started with MicroPython on Raspberry Pi Pico” Raspberry Pi Press, January25, 2021

```
while True:
    RVout = cds_r.read_u16() * (3.0 / 65535)
    LVout = cds_l.read_u16() * (3.0 / 65535)
    Vcc = vcc.read_u16() * (3.0 / 65535)
    R3 = (RVout/(Vcc-RVout))*5000-1000
    R6 = (LVout/(Vcc-LVout))*5000-1000

    if (R3 > R6):
        df = abs((R6-R3)/abs(R3))*100
        if (df <= 8):
            motor_R.value(0)
            motor_L.value(0)
        else:
            motor_R.value(1)
            motor_L.value(0)
    else:
        if (R3 < R6):
            df = abs((R3-R6)/abs(R6))*100
            if (df <= 8):
                motor_R.value(0)
                motor_L.value(0)
            else:
                motor_R.value(0)
                motor_L.value(1)
        else:
            pass
```

図12　Python を用いた制御プログラム