

# 琉球大学学術リポジトリ

カードを用いたプログラミングに関する研究：  
カードによるライントレーサ制御

メタデータ	言語: 出版者: 琉球大学教育学部 公開日: 2015-12-16 キーワード (Ja): キーワード (En): 作成者: 新垣, 学, Arakaki, Manabu メールアドレス: 所属:
URL	<a href="http://hdl.handle.net/20.500.12000/32927">http://hdl.handle.net/20.500.12000/32927</a>

# カードを用いたプログラミングに関する研究

## —カードによるライントレーサ制御—

新垣 学

A study of card programming for line tracer control

Manabu Arakaki

### I. はじめに

平成 20 年の中学校学習指導要領の改訂により、情報の領域にプログラムによる計測・制御の項目が加わった。これに伴い、筆者は教員免許状更新講習を通して現場の教員に「マイコンを用いたおもちゃ作り」として、図 1 に示す PIC マイコン制御によるライントレーサの製作方法を教えてきた。16 から 20 名の受講者の多くは工業高校の教員で、次に中学校技術の教員となっている。毎年、数名は特別支援学校や小学校の教員が受講している。

ライントレーサは白地に黒で引かれた線をなぞるように走るロボットカーである。モータやギヤボックス、制御回路、マイコンで構成されており、製作を通して中学校技術以上の様々な知識を学習できる。各パーツをユニバーサル基板にネジで固定しており、電気回路もブレッドボード上に配線しているため、講習後もよりよく走らすための調整が可能だが、マイコンのプログラム部分は特別

な書き込み装置が必要となる。価格が 5 千円程度するので、プログラムまでは手を付けにくいと考えられる。現場の教員が生徒や児童の前で簡単かつ自由にロボットカーを動かして見せることができるようにするため、ここでは、カードによるプログラム入力装置を安価にライントレーサに付加する方法およびその応用を示す。

### II. カードの読み取り装置

カードで情報を伝える方法としては、磁気カードや IC カードが一般的だが、これらはカードの製作や読み取り装置が高価となる。ここでは、ライントレーサがラインを検出する方法と同様に図 2 に示すバーコードによる入力を検討する。バーコードの白黒は高さ 10mm、幅 5mm を単位としてパソコンで作図してプリントするか、方眼紙に正確に記録する。白黒情報を読み取るだけなら、光センサを用いて容易かつ安価に実現できる。読み取る方法はバーコードの白黒情報の数だけセンサを

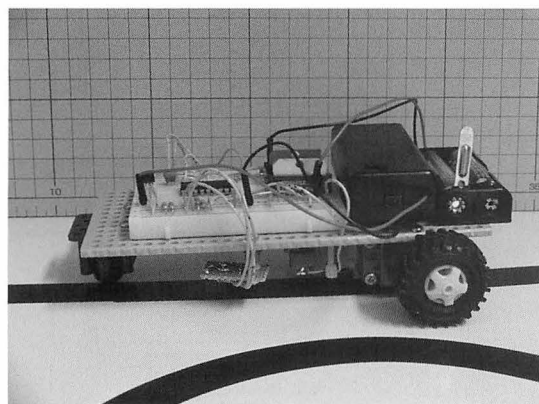


図 1. ライントレーサ

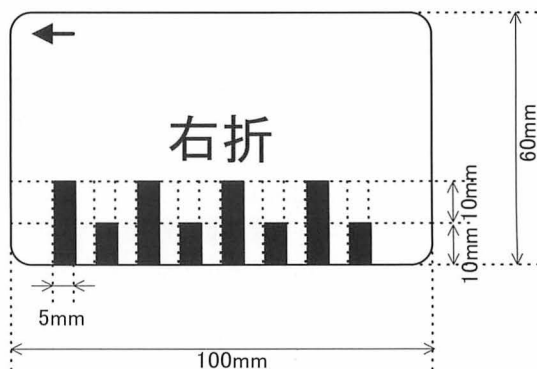


図 2. 命令情報をバーコード化したカード

並べる方法と、1つのセンサで白黒をスキャンしていく方法が考えられるが、前者はコスト、製作、拡張性の面で後者に劣るため、ここでは後者の方法で実現する。

店頭の商品に付されているバーコードは一定の速度でスキャンされるため、1列でも安定して読み取ることができるが、製作する装置では人がカードを動かすことでバーコードをスキャンするため、正確に情報を読み取ることは難しい。ここでは、読み取るタイミングを正確に得るため、タイミング用のバーコードを1本追加する。

図2に示すカードにおいて、下の列がタイミング用で上の列が命令情報を表している。タイミング用バーコードが黒の部分で命令情報を読み取る。カードは左方向に挿入されるので、黒を1、白を0とすると、この命令情報は10101010と認識される。

図3にカード読み取り装置の全体図を示す。ガイドは横10cm、高さ6cm、厚さ2.5mmのプラスチックダンボールの壁と、スリットの底となる高さ1cmのプラスチックダンボールで構成する。バーコードの読み取りにはフォトリフレクタを用いた。バーコードは2列あるのでフォトリフレクタはガイドの上下に1個ずつ取り付ける。フォトリフレクタと対面するガイドの壁は白紙を貼っておく。フォトリフレクタは赤外線LEDとフォトトランジスタで構成されており、LEDより照射された赤外光はバーコードで反射され、フォトトランジスタで受けとられる。バーコードの白の部分で反射光が大きくなり、黒の部分で反射光が最少となる。フォトリフレクタの対面の壁は白なので、カードをスキャンしていないときは反射光が最大となる。

図4にバーコード1列分の読み取りセンサ回路を示す。受光量に応じてフォトトランジスタの電流は増減し、センサの出力電圧を変化させる。ここでは、抵抗R2を120kΩとしたが、フォトリフレクタの特性やカード読み取り装置への取り付け具合によって出力電圧が変化する。マイコンでの白黒の判断は1.1V程度を閾値として行われるので、バーコードの白黒におけるセンサ出力がこれを境に大きく変化するようにR2の値を調整する必要がある。白の場合1.1Vより十分高く、黒

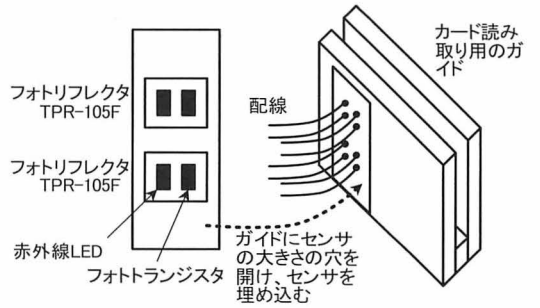


図 3. カード読み取り装置

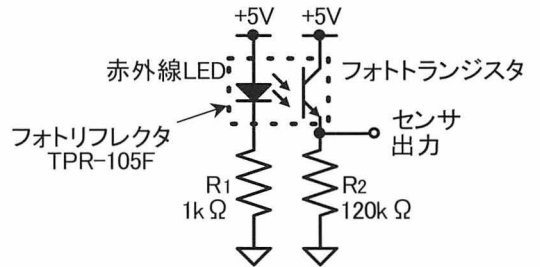


図 4. バーコード読み取りセンサ回路

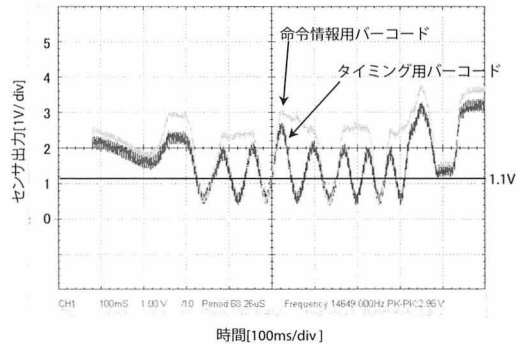


図 5. 命令情報 10101010 を普通にスキャン

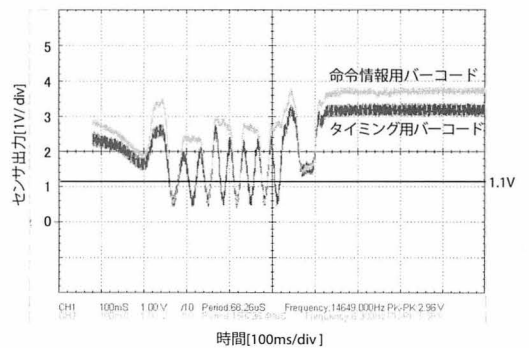


図 6. 命令情報 10101010 を高速にスキャン

の場合 1.1V より十分低くする。

図5、図6に命令情報 10101010 をスキャンした場合のオシロスコープ波形を示す。バーコードの黒が命令情報の1、白が命令情報の0を表すことにしており、タイミング用バーコード波形が左から右に向かって8回 1.1V を下回るのに対し、命令情報 (10101010) のバーコード波形はタイミング用バーコードの奇数番目で下がっていることがわかる。

また、図5、図6より、スキャンの速度によって8ケタ分の電圧の変化に270～500msを要していることがわかる。波形に0.5V程度の幅があるのは、容易かつ安価に製作するためにノイズ対策回路を全く付けていないためである。この波形から正しく命令情報を読み取るためにはタイミング用バーコードが1(1.1V以下)を示す期間の中間で命令情報を読み取ればよい。図6において1.1Vの期間が最も短い所で15msであった。よって、タイミング用バーコードが1になってから、7.5ms後に命令情報を読み取ればよいと言える。

### Ⅲ. 制御プログラムの入力について

ライントレーサは通常、黒のラインにそって移動するが、交差点やT字路など、ライン検出用のセンサが共に黒を検出した場合停止する。ここでは、その場合どうするかということカードでプログラムできるようにする。表1に命令情報とバーコードの関係を示す。

もっとも単純な命令は「前進」「右折」「左折」「停止」「後進」である。例えば方眼紙のように縦横に黒い線を引いたコースにライントレーサを置き、右折のカードを4回通して実行すれば、交差点において右折を4回繰り返し、隣接するブロックを一周することができる。

「FOR 開始」と「FOR 終了」は一般的なプログラム言語における繰り返し命令を実現するためのもので「FOR 開始」「2～15」「実行したい複数の命令」「FOR 終了」の順にカードを通すことで、2～15回の間で「実行したい複数の命令」を実行することができる。例えば、「FOR 開始」「4」「右折」「左折」「FOR 終了」の順にカードを通すことで右斜め方向に4ブロック進むことができる。

表1. 命令情報とバーコードの関係

命令情報	バーコード
前進	11111111
右折	10101010
左折	1010101
停止	11110000
後進	11111100
FOR開始	10010001
FOR終了	10010010
IFRA0(RA0が1の場合)	10001110
IFEL0(RA0が1でない場合)	10001100
IFE0(RA0に対するIFの終了)	10001000
IFRA1(RA1が1の場合)	10011110
IFEL1(RA1が1でない場合)	10011100
IFE1(RA1に対するIFの終了)	10011000
IFRA2(RA2が1の場合)	10101110
IFEL2(RA2が1でない場合)	10101100
IFE2(RA2に対するIFの終了)	10101000
IFRA3(RA3が1の場合)	10111110
IFEL3(RA3が1でない場合)	10111100
IFE3(RA3に対するIFの終了)	10111000
終了	00000000
2	00000010
3	00000011
4	00000100
5	00000101
6	00000110
7	00000111
8	00001000
9	00001001
10	00001010
11	00001011
12	00001100
13	00001101
14	00001110
15	00001111

「IFRA0」「IFEL0」「IFE0」は条件分岐を実現するもので、ここでは条件としてマイコンの入力端子 RA0 に論理 1 (1.1V を超える電圧) に相当する信号が入っているかをチェックする。RA0 が 1 の場合、「IFRA0」と「IFEL0」の間に入力

した命令が実行される。そうでない (0) 場合は「IFEL0」と「IFE0」の間に入力した命令が実行される。この他、条件分岐命令はRA1、RA2、RA3、RA4のどの入力端子をチェックするかによって4種類ある。

#### IV. マイコンのプログラム

マイコンのプログラムはバーコードから命令情報を読み込む部分とそれを実行する部分に分けられる。

図7にバーコードを読み込むプログラムのフローチャートを示す。各処理の横に太字で付してある文字は実際のプログラムにおけるラベルである。

はじめに各変数を初期化後、プログラムモードのスイッチがONであればラベルMAIN2に移ってバーコードの読み込みを開始する。II章の結果より、タイミング用バーコードが1のとき、7.5ms待ってから命令情報用バーコードを読み込む。読み込んだ01情報はデータ一時格納用変数DATAINを1ケタ左にずらしてから1ケタ目にセットする。その後、タイミング用バーコードが0になるのを待つ。0になった後、いきなり次の1を待つのではなく、ノイズ対策のためにここでも7.5ms待ってから次の動作に移る。命令情報用バーコードの何ケタ目を読んだのか数えている変数BITCT（最初に8がセットされている）を1減らし、0でない間バーコードを読み続ける。8回の読み込み後、DATAINには命令情報用バーコードの01情報が転写される。そのデータをFSRが示す番地に格納し、FSRを1増やした後、ラベルMAINにもどって次のカードがスキャンされるのを待つ。

もし、スキャンに失敗し、8ケタの命令情報が読み込めなかった場合、タイミング用のセンサが白を検出し続けることになり、ラベルCNTWの処理にて欠損データがクリアされる。

図8に読み込んだプログラムを実行するプログラムのフローチャートを示す。ラベルPRNTから始まる。はじめに、カードから読み込んだプログラムが格納されている先頭アドレス20HをFSRにセットする。その後、黒線にそって走るライトレースのためのプログラムを実行する。

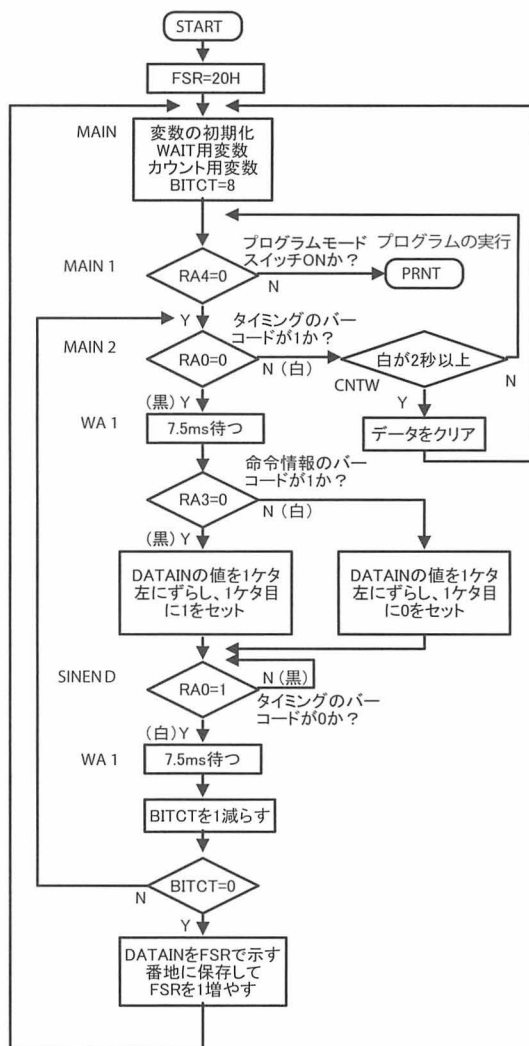


図7. バーコードを読み込むプログラムのフローチャート

ライン検出用センサが共に黒を検出した場合、FSR番地で示される命令情報を解釈しながら動くことになる。解釈する内容は表1に示したように、前進、右折、左折、停止、後進のほか、繰り返し命令や条件分岐命令を解釈する。

解釈により、命令情報が前進であった場合、マイコンの出力ポートPORTBに10010101を出力し、左右のモータを前進させ、動作確認用のLEDを光らせる。

繰り返し命令の「FOR開始」の場合、次のFSR番地のデータを繰り返し回数としてCNT0にセットし、さらに次のFSR番地を「繰り返

すべき命令」のアドレスとしてCNT1にセットする。「繰り返すべき命令」を実行後、「FOR終了」にてCNT0を1減らし、0でなければFSRにCNT1の値をセットし、「繰り返すべき命令」の最初まで実行順を戻す。これによりCNT0が0になるまで「繰り返すべき命令」が繰り返される。

条件分岐命令の「IFRA0」の場合、マイコンの入力端子RA0をチェックし、論理1に相当する入力(1.1Vより大きい電圧)なら次の命令を実行していき、「IFEL0」が現れたら「IFE0」が現れるまで命令を読み飛ばし、条件分岐を終了する。つまり「IFRA0」と「IFEL0」の間の命令を実行する。RA0が1でない場合は、「IFEL0」が現れるまで命令を読み飛ばし、その次の命令から実行するか、「IFE0」で終了する。つまり「IFEL0」と「IFE0」の間の命令を実行するか、何もしない。

フローチャートにはRA1、RA2、RA3を条件とする条件分岐命令を省略してある。

各命令を実行する度、ラベルPRNT2でFSRを1増やし、次の命令を読む準備をする。ただし、「終了」命令の場合はラベルSTARTに戻ってプログラムモードになる。

### V. カードプログラム型ライントレーサの製作

カードによるプログラム入力装置を搭載したライントレーサの全回路図を図9に示す。

マイコンPIC16F84AとモータードライバTA7291Pを使うよく知られた構成に、タイミング用バーコードを読み取るためのセンサと命令情報用バーコードを読み取るためのセンサを追加する。

追加に要する回路の費用はフォトリフレクタ2つと抵抗4つ、ユニバーサル基板、少量の配線なので、ライントレーサの製作と同時にすれば200円程度である。

タイミング用と命令情報用のセンサは同一のものである。また、ライン検出用の左センサと右センサも同一のものである。バーコード用センサとライン検出センサはフォトリフレクタが異なるため、負荷抵抗を一部変えてある。高密度でバーコードを認識するため、バーコード用のセンサには小型のものを用いている。製作のしやすさからライン検出用は大きめのものを利用しているが、バー

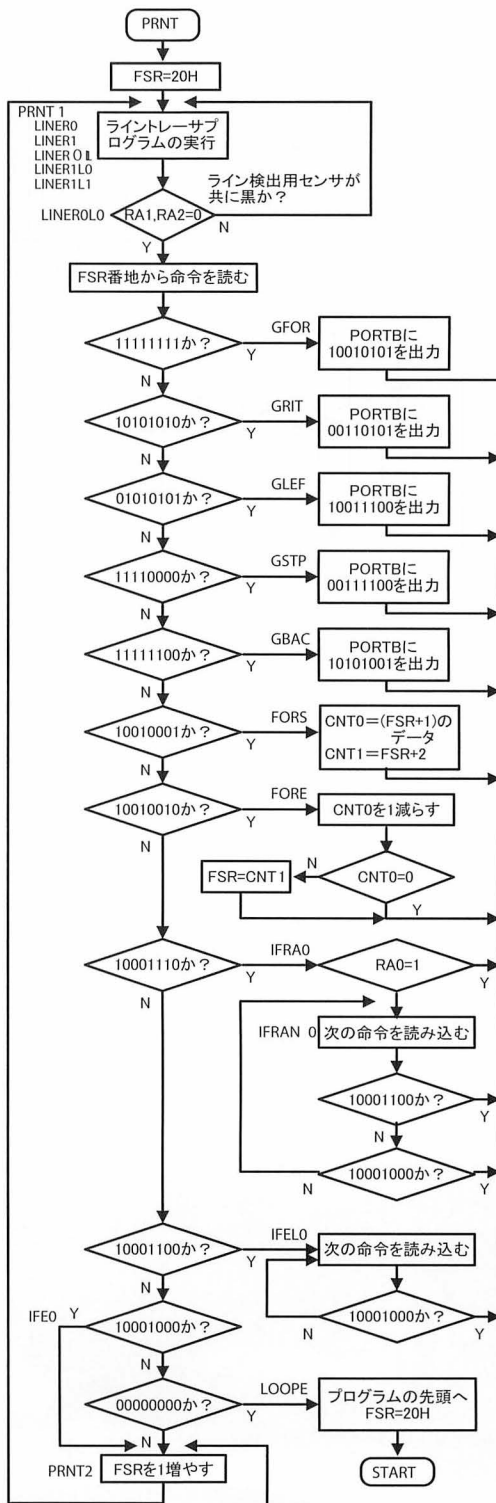


図 8. 読み込んだプログラムを実行するプログラムのフローチャート

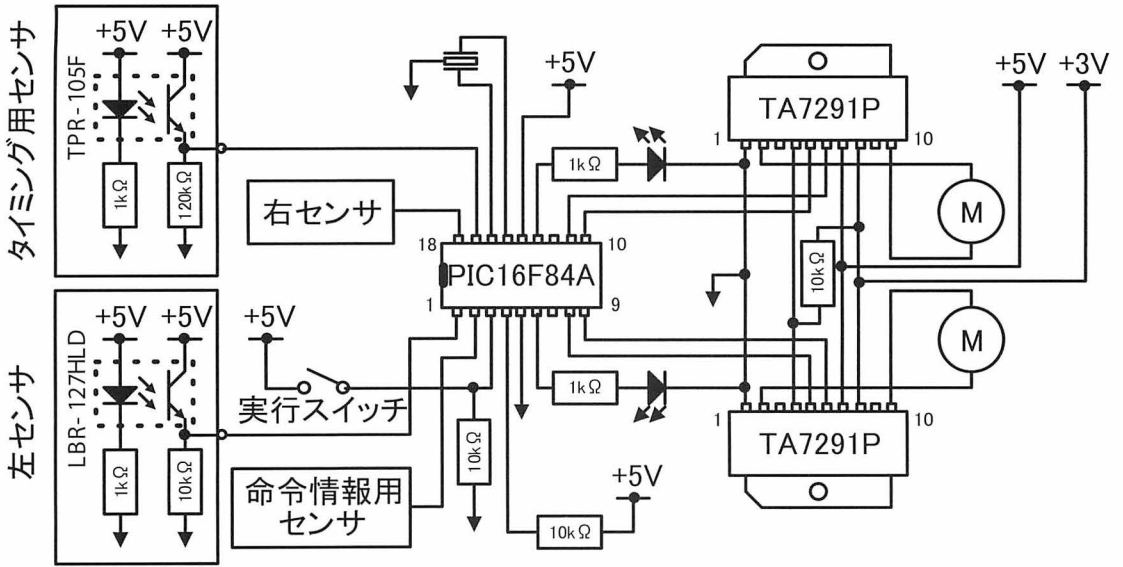


図 9. カードによるプログラム入力装置を搭載したライトレーサの全回路図

コード用と同じものを使ってもよい。

容易かつ安価に構成するためにノイズ対策の回路は追加せず、マイコンのプログラムでノイズを処理する。

実行スイッチは通常開き、プログラムモードにする。

図 10 にカード読み取り装置を搭載したライトレーサを示す。左の黒い部分が取り付けしたカード読み取り装置である。

## VI. カードによるプログラミングの学習例

図 11 にサンプルコースを示す。START の位置に置かれたライトレーサをプログラムして

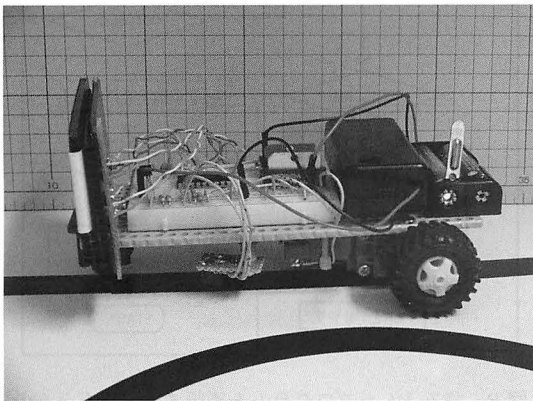


図 10. カードプログラミング型ライトレーサ

GOAL に導くことを目的とする。低学年の児童でも前進、左折、右折カードを交差点ごとに指示すれば 15 手で GOAL できる。

繰り返し命令を理解できれば「FOR 開始」、「5」、

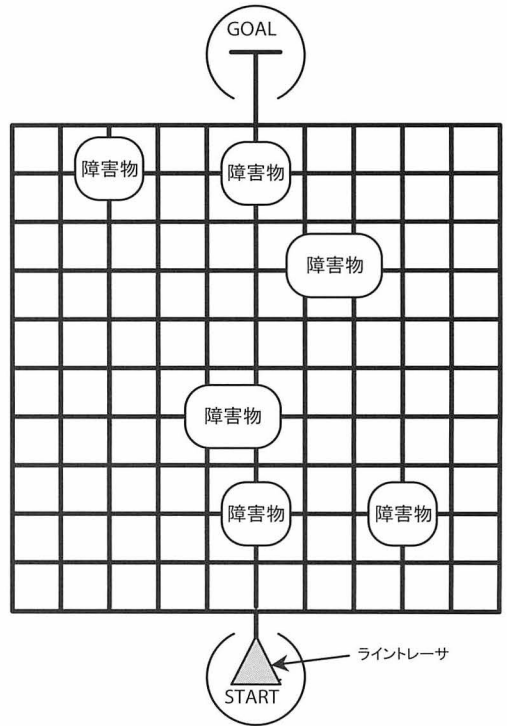


図 11. サンプルコース

「左折」、「右折」、「FOR 終了」、「前進」、「FOR 開始」、「5」、「右折」、「左折」、「FOR 終了」と、11 手で GOAL できる。

コースに障害物を置くことや、手持ちのカード数の制限、ランダムなカードの配給、プログラム時間制限つきの対戦ゲームにする等、これらを組み合わせることでプログラムを創意工夫する力が養えると考えられる。

## VII. まとめ

いつでもどこでもライントレーサをプログラムできるように、ライントレーサを制御するための命令をカードにし、簡単かつ安価な読み込み装置

と、それで読み込ませた命令を解釈・実行できるマイコンのためのプログラムを作成した。

また、ゲームをしながらプログラミングを学ぶ方法を提案した。

さらなる応用として、1つのカードを複数の制御命令の発動キーとすることでロボットの関節を複数同時に制御し、ロボットに様々なポーズをとらせることが考えられる。そして、いくつかのカードを組み合わせることでロボットを躍らせることで、児童・生徒にプログラミングへの興味を持ってもらうことが期待できる。

今後、ここでの成果を教員免許状更新講習に活用する予定である。



付録. マイコンのプログラム

```

;
; card読み取り後RA4をHIで再生
;

```

```

LIST P=PIC16F84A
INCLUDE "P16F84A.INC"
_CONFIG_HS_OSC & _WDT_OFF

```

```

TIM1 EQU 0CH
TIM2 EQU 0DH
TIM3 EQU 0EH
CNT0 EQU 0FH
CNT1 EQU 10H
CNT2 EQU 11H
BITCT EQU 13H
DATIN EQU 14H
DATCT EQU 15H
ORG 0
GOTO START

```

START

```

BSF STATUS, RP0
MOVLW B'00011111'
MOVWF TRISA
CLRF TRISB
BCF STATUS, RP0
bcf STATUS,IRP
movlw 20H
movwf FSR
clrf DATCT
clrf BITCT

```

```

movlw b'10010001'
movwf 20H
movlw b'00000100'
movwf 21H
movlw b'10101010'
movwf 22H
movlw b'10010010'
movwf 23H
movlw b'10010001'
movwf 24H
movlw b'00000100'
movwf 25H
movlw b'01010101'
movwf 26H
movlw b'10010010'
movwf 27H
movlw b'00000000'
movwf 28H

```

MAIN

```

MOVLW 0H
MOVWF CNT0
MOVLW 0H
MOVWF CNT1
MOVLW 9H
MOVWF CNT2
movlw 8H

```

```

movwf BITCT
MAIN1
BTFSZ PORTA,4
goto PRNT
MAIN2
BTFSZ PORTA,0
goto CNTW
call WA1
btfsc PORTA,3
goto SINW
rlf PORTB
rlf DATIN
bsf PORTB,0
bsf DATIN,0
goto SINEND
SINW
rlf PORTB
rlf DATIN
bcf PORTB,0
bcf DATIN,0
SINEND
btfsz PORTA,0
goto SINEND
call WA1
decfsz BITCT
goto MAIN2
movf DATIN,W
movwf INDF
incf FSR,F
clrf INDF
call WAIT1
goto MAIN
CNTW
DECFSZ CNT0, F
GOTO CNTWEND
DECFSZ CNT1, F
GOTO CNTWEND
DECFSZ CNT2, F
GOTO CNTWEND
clrf PORTB
goto MAIN
CNTWEND
goto MAIN1
PRNTV
movlw 20H
movwf FSR
PRNTV1
movf INDF,W
movwf PORTB
sublw B'00000000'
btfsc STATUS,Z
goto PRNTV1E
call WAIT1
call WAIT1
incf FSR,f
goto PRNTV1
PRNTV1E

```

```

        movlw    20H
        movwf   FSR
        call    WAIT1
        clrf   PORTB
        call    WAIT1
        goto    PRNT
PRNT
PRNT0
        movlw    20H
        movwf   FSR
PRNT1
; ライントレースプログラム
        btfss   PORTA,1
        goto    LINER0
        goto    LINER1
LINER0
        btfss   PORTA,2
        goto    LINER0L0
        goto    LINER0L1
LINER1
        btfss   PORTA,2
        goto    LINER1L0
        goto    LINER1L1
LINER0L1
        movlw   B'10110100'
        movwf   PORTB
        goto    PRNT1
LINER1L0
        movlw   B'00011101'
        movwf   PORTB
        goto    PRNT1
LINER1L1
        movlw   B'10010101'
        movwf   PORTB
        goto    PRNT1
LINER0L0
;実行するプログラムの解釈>
        movlw   B'11111111'
        subwf   INDF,w
        btfsc   STATUS,Z
        goto    GFOR
        movlw   B'10101010'
        subwf   INDF,w
        btfsc   STATUS,Z
        goto    GRIT
        movlw   B'01010101'
        subwf   INDF,w
        btfsc   STATUS,Z
        goto    GLEF
        movlw   B'11110000'
        subwf   INDF,w
        btfsc   STATUS,Z
        goto    GSTP
        movlw   B'11111100'
        subwf   INDF,w
        btfsc   STATUS,Z
        goto    GBAC
        movlw   B'10010001'
        subwf   INDF,w

```

```

        btfsc   STATUS,Z
        goto    FORS
        movlw   B'10010010'
        subwf   INDF,w
        btfsc   STATUS,Z
        goto    FORE
        movlw   B'10001110'
        subwf   INDF,w
        btfsc   STATUS,Z
        goto    IFRA0
        movlw   B'10011110'
        subwf   INDF,w
        btfsc   STATUS,Z
        goto    IFRA1
        movlw   B'10101110'
        subwf   INDF,w
        btfsc   STATUS,Z
        goto    IFRA2
        movlw   B'10111110'
        subwf   INDF,w
        btfsc   STATUS,Z
        goto    IFRA3
        movlw   B'10001100'
        subwf   INDF,w
        btfsc   STATUS,Z
        goto    IFEL0
        movlw   B'10011100'
        subwf   INDF,w
        btfsc   STATUS,Z
        goto    IFEL1
        movlw   B'10101100'
        subwf   INDF,w
        btfsc   STATUS,Z
        goto    IFEL2
        movlw   B'10111100'
        subwf   INDF,w
        btfsc   STATUS,Z
        goto    IFEL3
        movlw   B'10001000'
        subwf   INDF,w
        btfsc   STATUS,Z
        goto    IFE0
        movlw   B'10011000'
        subwf   INDF,w
        btfsc   STATUS,Z
        goto    IFE1
        movlw   B'10101000'
        subwf   INDF,w
        btfsc   STATUS,Z
        goto    IFE2
        movlw   B'10111000'
        subwf   INDF,w
        btfsc   STATUS,Z
        goto    IFE3
        movlw   B'00000000'
        subwf   INDF,w
        btfsc   STATUS,Z
        goto    LOOPE

```

PRNT2	incf FSR,F		goto FORE1
	goto PRNT1		goto PRNT2
		FORE1	
LOOPE			movf CNT1,w
	movlw 20H		movwf FSR
	movwf FSR		goto PRNT1
	goto MAIN		
;-< 解釈した命令に基づき			
; 実際には実行されるプログラム >-			
GFOR		IFRA0	btfss PORTA,0
			goto IFRAN0
			goto PRNT2
		IFRAN0	
	movlw B'10010101'		incf FSR
	movwf PORTB		movlw B'10001100'
	call WAIT1		subwf INDF,w
	clrf PORTB		btfsc STATUS,Z
	call WAIT1		goto PRNT2
	goto PRNT2		movlw B'10001000'
GRIT			subwf INDF,w
	movlw B'10110100'		btfsc STATUS,Z
	movwf PORTB		goto PRNT2
	call WAIT1	IFELO	goto IFRAN0
	clrf PORTB		
	call WAIT1		incf FSR
	goto PRNT2		movlw B'10001000'
GLEF			subwf INDF,w
	movlw B'00011101'		btfsc STATUS,Z
	movwf PORTB		goto PRNT2
	call WAIT1	IFE0	goto IFEL0
	clrf PORTB		
	call WAIT1		goto PRNT2
	goto PRNT2	IFRA1	
GSTP			btfss PORTA,1
	movlw B'00111100'		goto IFRAN1
	movwf PORTB		goto PRNT2
	call WAIT1	IFRAN1	
	clrf PORTB		incf FSR
	call WAIT1		movlw B'10011100'
	goto PRNT2		subwf INDF,w
GBAK			btfsc STATUS,Z
	movlw B'10101001'		goto PRNT2
	movwf PORTB		movlw B'10011000'
	call WAIT1		subwf INDF,w
	clrf PORTB		btfsc STATUS,Z
	call WAIT1		goto PRNT2
	goto PRNT2		goto IFRAN1
;-< FOR >-			
FORS		IFEL1	incf FSR
	incf FSR,F		movlw B'10011000'
	movf INDF,w		subwf INDF,w
	movwf CNT0		btfsc STATUS,Z
	movlw B'00001111'		goto PRNT2
	andwf CNT0,F		goto IFEL1
	movf FSR,w	IFE1	
	movf CNT1		goto PRNT2
	incf CNT1,F	IFRA2	
	goto PRNT2		btfss PORTA,2
FORE			goto IFRAN2
	decfsz CNT0		goto PRNT2

```

IFRAN2
    incf    FSR
    movlw  B'10101100'
    subwf  INDF,w
    btfsc  STATUS,Z
    goto   PRNT2
    movlw  B'10101000'
    subwf  INDF,w
    btfsc  STATUS,Z
    goto   PRNT2
    goto   IFRAN2

IFEL2
    incf    FSR
    movlw  B'10101000'
    subwf  INDF,w
    btfsc  STATUS,Z
    goto   PRNT2
    goto   IFEL2

IFE2
    goto   PRNT2

IFRA3
    btfss  PORTA,3
    goto   IFRAN3
    goto   PRNT2

IFRAN3
    incf    FSR
    movlw  B'10111100'
    subwf  INDF,w
    btfsc  STATUS,Z
    goto   PRNT2
    movlw  B'10111000'
    subwf  INDF,w
    btfsc  STATUS,Z
    goto   PRNT2
    goto   IFRAN3

IFEL3
    incf    FSR
    movlw  B'10111000'
    subwf  INDF,w
    btfsc  STATUS,Z
    goto   PRNT2
    goto   IFEL3

IFE3
    goto   PRNT2
< 待ち時間を作るためのサブルーチン >
WA1  MOVLW  25H
     MOVWF  TIM2
WA2  MOVLW  0H
     MOVWF  TIM3
WA3  NOP
     DECFSZ TIM3, F
     GOTO   WA3
     DECFSZ TIM2, F
     GOTO   WA2
     RETURN

WAIT1
    MOVLW  13H
    MOVWF  TIM1
    
```

```

WA11  MOVLW  0H
      MOVWF  TIM2
WA12  MOVLW  0H
      MOVWF  TIM3
WA13  NOP
      DECFSZ TIM3, F
      GOTO   WA13
      DECFSZ TIM2, F
      GOTO   WA12
      DECFSZ TIM1, F
      GOTO   WA11
      END
    
```