

## ライントレースロボットの仕様書

### 1. 機能

走行方式は、**3** センサのうち真ん中のセンサがライン上にあるとき **MAX** スピードで走行し、真ん中のセンサが、ライン外になった際は、徐行をする。左センサがライン上に乗った際は、右のタイヤのみをセンサがライン外に出るまで **MAX** スピードで回転させる、右センサがライン上に乗った際は、左モータを右センサがライン外になるまで **MAX** スピードで回転させる。

方向指示器は、左センサがライン上に載ってる再は、右側の **LED** が点灯する。

右センサがライン上のときは左の **LED** が点灯する。

### 2. 基本仕様

表 1. 基本仕様表

ライン検出方法	<b>3</b> センサ
使用モータ	<b>FA130</b>
ギアボックス形式	ダブルギアボックス（左右 <b>4</b> 速独立タイプ）
ギア比	<b>B (38.2:1)</b>
車輪径	<b>56mm</b>
車輪幅	<b>25mm</b>
重量	<b>0.365kg</b>
制御部電源電圧	<b>3V</b>
モータ駆動電圧	<b>3V</b>

### 3. 基本性能

今回製作したライントレースの速度は、約 **0.4m/s** 程度であった。

### 4. 電子回路図

電子回路図は図 **1** として **3** ページに示す。

### 5. 制御プログラムのフローチャート

制御プログラムのフローチャートを図 **2** として **4** ページに示す。

### 6. プログラミングコード

プログラミングコードは **5~7** ページに示す。

### 7. 電子部品表

電子部品表は **8** ページに示す。

### 8. 費用

費用は、総額 **3754.5** 円かった。

## 9. レース結果

レース結果は、本番の計測日には **49** 秒ほどであった。

前日の計測では、**37** 秒ほどであったが、改良のため、ギア比を C から B に変更したら、制御がうまくいかず、タイムが落ちてしまった。

## 10. 工夫した点

私の工夫した点は、ハードウェア面では、**3** センサでラインを感知すること、センサ基板に半固定抵抗まで実装したこと、基板をきれいに配線し、なるべく作業エラーを少なくしたことである。

ソフトウェア面では、擬似 **PWM** 制御を行った点だ。

**High** と **Low** の待機時間差により擬似的に電圧を下げ速度調整を行った点だ。この速度調節により、急カーブでも曲がれるようになった。

この時間差を何度も調整することにより、高速化することが可能になった。

## 11. その他

今回の実習ではトランジスタにより、モータを制御したが、現在トランジスタでモータを制御していることは稀である。多くの **DC** モータは、**FET** または、モータドライバで制御している。そのようなことを考慮して指導していただきたいと感じた。



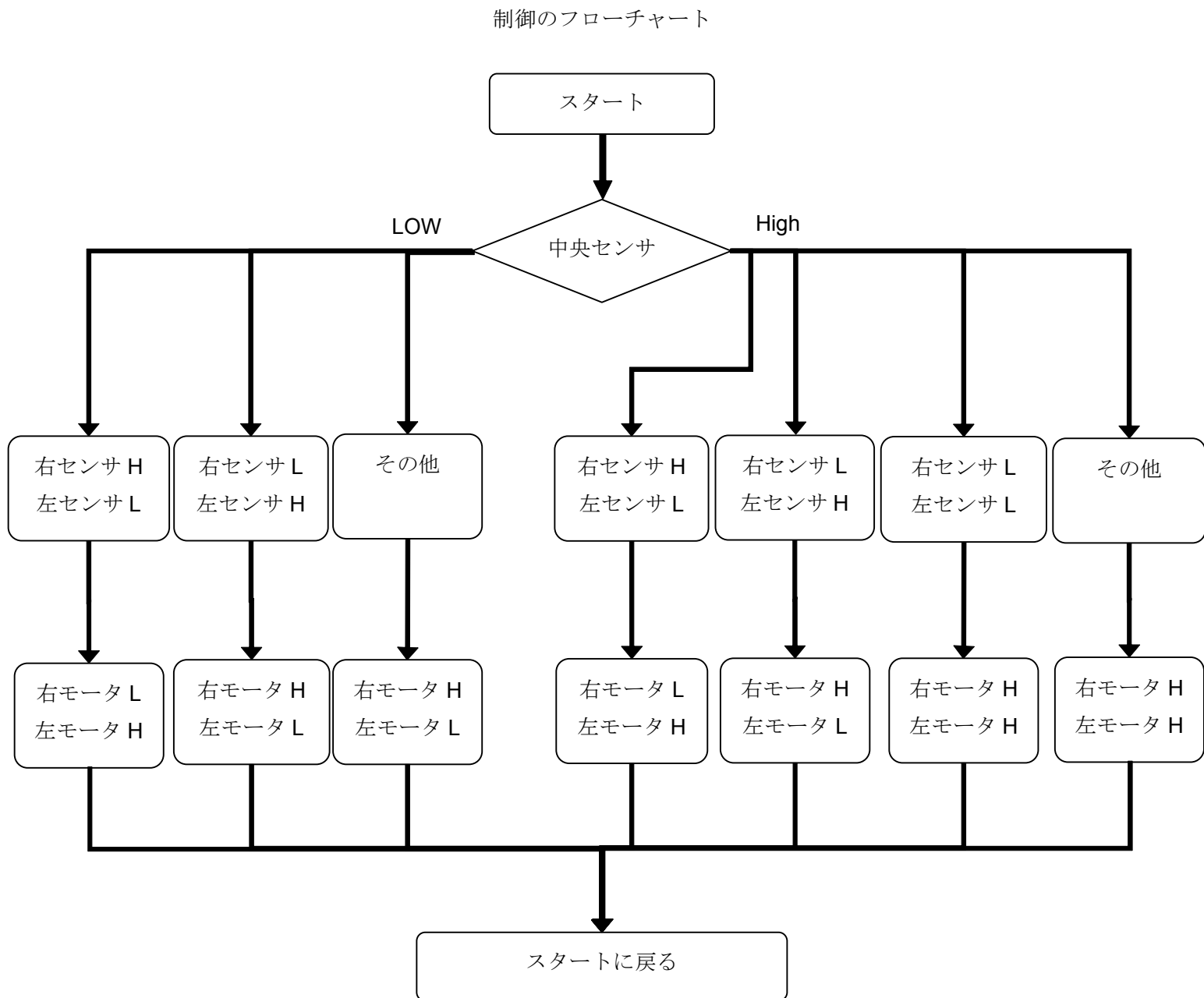


図 2. 制御のフローチャート

```

/*      右センサ RA0 中央センサ RA1 左センサ RA2
      右モータ RB7 左モータ RB0 */

```

```

#include <PIC.h>

```

```

__CONFIG(0xFFFA);

```

```

wait(short k){
    short i;
    short j;
    for(j=0; j<k; j++){
        for(i=0; i<30; i++){
        }
    }
}

```

```

int Left(void){          //左に曲がる関数
    RB0=1;
    RB7=0;
}

```

```

int Right(void){         //右に曲がる関数
    RB0=0;
    RB7=1;
}

```

```

int Straight(void){      //前進の関数
    RB0=1;
    RB7=1;
}

```

```

int Stop(void){          //停止の関数
    RB0=0;
    RB7=0;
}

```

```

Delay_ms(short x){      //wait 時間調整の関数
    wait(5);
}

```

```

Int Sens(void){          //LED とセンサを対応させる」関数
    RA0=RB3;
    RA2=RB4;
}

int main(void){          //メイン関数
    TRISA=0xFF;
    TRISB=0x00;
    PORTB=0xFF;
    While(1){           //無限ループ
        Sens();
        if(RA1==1){      //中央センサがライン上るとき
            if(RA0==0&&RA2==1){          //左センサのみライン上
                Right();
            }

            else if(RA0==1&&RA2==0){      //右センサのみライン上
                Left();
            }

            else if(RA0==1&&RA2==1){      //左右センサがライン上
                Straight();
                Delay_ms(1)
                Stop();
                Delay_ms(3)                //25%のパワーで前進(徐行)
            }

            else{          //その他
                Straight();
            }
        }

        else{              //中央センサがライン外
            if(RA0==0&&RA2==1){          //右センサのみライン上
                Right();
            }
        }
    }
}

```

```
    }  
  
    else if(RA0==1&&RA2==0){           //左センサのみライン上  
        Left();  
    }  
  
    else{                               //その他  
        Right();  
    }  
}  
}
```