

ライントレースロボット

OAKS - LABO

ユーザズマニュアル Ver2.1

TM統合環境 + Nc30wa Ver4.00 Release2対応版



オークス電子株式会社

目 次

| | |
|---------------------------------|----|
| 1.はじめに | 3 |
| 2.OAKS - LABOの特徴 | 3 |
| 3.OAKS - LABOの仕様 | 4 |
| 4.モータドライブボード | 5 |
| (1) 励磁電流調整用ボリューム | 5 |
| (2) 励磁遮断スイッチ | 6 |
| (3) LCD表示器 | 6 |
| (4) その他のスイッチ | 7 |
| 5.センサボード | 7 |
| (1) 感度調整ボリューム | 7 |
| (2) フォトレフレクタの特徴 | 8 |
| (3) 接触センサ | 8 |
| 6.CPUボード | 9 |
| (1) パラレル入出力ポート | 9 |
| (2) シリアルポート | 10 |
| 7.OAKS - LABOの調整方法 | 12 |
| (1) モータ電流の調整 | 12 |
| (2) ラインセンサの調整 | 13 |
| 白ラインコースの場合 | 14 |
| 黒ラインコースの場合 | 14 |
| 8.OAKS - LABOの走行テスト | 15 |
| (1) モード選択スイッチの割り当てと機能 | 15 |
| (2) 走行テスト手順 | 17 |
| 9.OAKS - LABOのしくみ | 20 |
| (1) ステッピングモータの駆動方法 | 20 |
| ソフトウェアタイマ方式 | 22 |
| タイマ割り込み方式 | 23 |
| (2) ラインセンサ入力 | 25 |
| 10.添付ファイルの構成と使い方 | 28 |
| (1) デバッガを使用する為のフラッシュライターソフトの使い方 | 28 |
| (2) デバッガの起動 | 32 |
| ファイルのロード | 32 |
| プログラムの実行方法 | 33 |

| | |
|--------------------------|-----|
| 11. T M統合化開発環境での使い方 | 3 4 |
| (1)インストールと注意点 | 3 4 |
| (2)エントリー版(新バージョン)の仕様について | 3 4 |
| (3) T M統合化開発環境の使い方 | 3 6 |
| 12. プログラムの作成手順 | 3 7 |
| (1)プログラムの実行手順 | 3 7 |
| (2)コンパイル方法 | 3 8 |
| (3)コンパイル用バッチファイル例 | 4 0 |
| (4) I / O ポートの記述例 | 4 0 |
| (5)割り込み処理 | 4 1 |
| (6)割り込みベクタテーブルへの登録 | 4 2 |
| 13. Flash R O Mへの書き込み | 4 4 |

付 録

| | | |
|------|----------------------|-----|
| 付録 1 | ドライバボード回路図 | 4 6 |
| 付録 2 | センサボード回路図 | 4 6 |
| 付録 3 | ドライバボード構成図 | 4 7 |
| 付録 4 | センサボード回路図 | 4 8 |
| 付録 5 | O A K S - L A B O外形図 | 4 8 |
| 付録 6 | コネクタ一覧 | 4 9 |
| 付録 7 | オプション(L A B O無線ユニット) | 5 1 |
| 付録 8 | 停止ラインの作り方 | 5 2 |
| 付録 9 | ライントレースロボットのコース例 | 5 3 |

安全上の注意

- 1) 万一、動作中及び充電中煙が出ている、変なにおいがする、発熱するなどの異常状態がみられる場合はすぐに電源を切ってください。感電や火傷に注意しながら速やかにコネクタをはずしてください。充電器は別売り専用急速充電器をご使用ください。
- 2) 電池から漏れた液が肌に触れると、火傷の原因になります。破損した電池に触れた場合は、すぐに水で洗い流してください。
- 3) 電池は絶対にショートさせないでください。発熱、液漏れ、破裂、発火の原因になります。
- 4) 液晶モニターが破損した場合、中の液晶には十分注意してください。体に付着したときはきれいな水で十分洗い流し、特に目に入った場合は医師の診断を受けてください。
- 5) 本製品はメカトロニクス学習用です。幼児、子供の手の届かない所に置いてください。

1 . はじめに

ロボットブームの今日、アイボ（ホンダ製）に代表される人間型ロボットや工場等での搬送用ロボットや介護用ロボットからホビー用ロボットまで、様々なロボットが次々に紹介されております。このようなロボットブームの21世紀を迎えた今日、ロボットの仕組みを知りたいと言う願望を持つ一般ユーザー人口が急激に増加しております。

ライントレースロボットは、床に貼った白又は黒いテープをなぞって走るロボットです。最も基本的なロボットですが、ロボットの原理を学ぶのに最も適したロボットと言えます。

初心者でも経験者でもライントレースロボットの製作に取り組む人にとって、その楽しさを経験し、より優秀なロボットの製作が容易に実現できます。

2.OAKS-LAB0 の特徴

最近ではP I Cマイコンシリーズ、H 8マイコンシリーズ、Z 8 0マイコンシリーズ等をC P Uポートに採用したライントレースロボットが多数紹介されています。しかし、これらのC P Uボードは、必ずしもユーザ側から見ると使い易いとは言えないのが現状です。理由は次の点にあります。

パラレルポートやタイマが足りずに増設が必要となる。

C P Uの周辺にR O MとR A Mを必ず接続し、ロムライターを用いてR O Mに書き込みが必要となる。

デバッガやコンパイラが高価でソフト開発には容易に使用できない。

処理速度が遅く、ロボット制御には向かない。

R O MやR A Mの容量が少なく、満足なプログラムが開発できない。

これらの問題を全て解決したライントレースロボットがO A K S - L A B Oです。C P UにはM 1 6 C / 6 2シリーズのM 3 0 6 2 0 F C A F Pと呼ばれる16ビット・ワンチップマイコンを搭載したC P Uポート「O A K S 1 6」が用いられています。

このような特徴以外に、次のような多くの特徴をもっています。

- プログラム作成に必要なコンパイラ、デバッガ、フラッシュライターソフトが付属している。
- ステッピングモータ駆動回路にはP W M I Cを使用し、高速回転を可能にしている。
- センサ基板は取り外しが可能となっているので、拡張コネクタ(Port_3)を併用することにより、マイクロマウス用センサ等が搭載可能となる。
- 拡張用R S 2 3 2 Cコネクタ(チャンネル0 : T x 0 , R x 0)にオプションの双方向R S 2 3 2 C無線ユニット又は単方向無線ユニットを取り付けると、無線による遠隔操作が可能となる。

3.OAKS-LABO の仕様

OAKS - LABOの仕様は、表1で示されるように、CPUには16ビットマイコンを搭載したOAKS 16マイコンポートがコネクタを介して取り外し可能となっています。

表1 OAKS - LABOの仕様

| 機 構 | 内 容 |
|---------|---|
| CPU | OAKS 16マイコンボード 三菱電機 30620FCAFP 使用 フラッシュROM 128KB、RAM 10KB 内臓 |
| モータ | 日本サーボ株式会社 KH39GM2-801 6.4V/0.47 |
| モータドライブ | サンケン LSA7020M 2個使用 |
| センサ | ローム RPR-220 4個使用 |
| 接触センサ | オムロン SS-01GL2-F 2個使用 |
| シャーシ | アルミニウム板組み立て一体型 ボールキャスタ2個付き |
| 車輪 | 直径48mm ゴムタイヤ付き |
| ソフトウェア | コンパイラNC30WA (V4.00 Release2 エントリー版)付き リモートデバッガKD30 付き フラッシュライターソフトM16CFIsh 付き |
| 電源 | 12[V] / 1100[mA] ニッカド電池 |
| マニュアル | ユーザーズマニュアル |
| 重量・サイズ | 780g(バッテリー無し) バッテリ:240g 横:110mm 長さ:205mm 高さ:107mm |

また、表2のようなオプションが別価格として用意されています。(詳細は付録 7をご覧ください。)

表2 オプション仕様

| オプション | 構 成 | 備考覧 |
|-------------------------|--|------------------------------|
| 単方向RS232C LABO無線ユニット | ・送信ユニット: No 9152 ・受信ユニット: No 9153 ・取り付けケーブル、アンテナ付き | チャンネル0:遠隔操作用 |
| 双方向RS232C LABO無線ユニット | ・送受信ユニット: No 9154 ・送受信ユニット: No 9155 ・取り付けケーブル、アンテナ付き | チャンネル0:遠隔操作用 チャンネル1:デバッグ用 |

4. モータドライブボード

モータドライブボードの構成図を図1に示します。サンケン社製のPWM(Pulse Width Modulation)制御によるチョッパ(Chopper)方式の駆動回路ICであるSLA7020Mを使用し、高速回転と高トルクを実現します。電源電圧をモータ定格の数倍にし、モータにはパルス状の電圧を加え、流れる電流の調整はパルスの幅で行うのでPWM制御と呼ばれます。

ステッピングモータは小型で高速回転が可能な中から定格電圧6.4[V]、電流0.47[A]用の日本サーボ(株)製KH39GM2-801を使用しています。

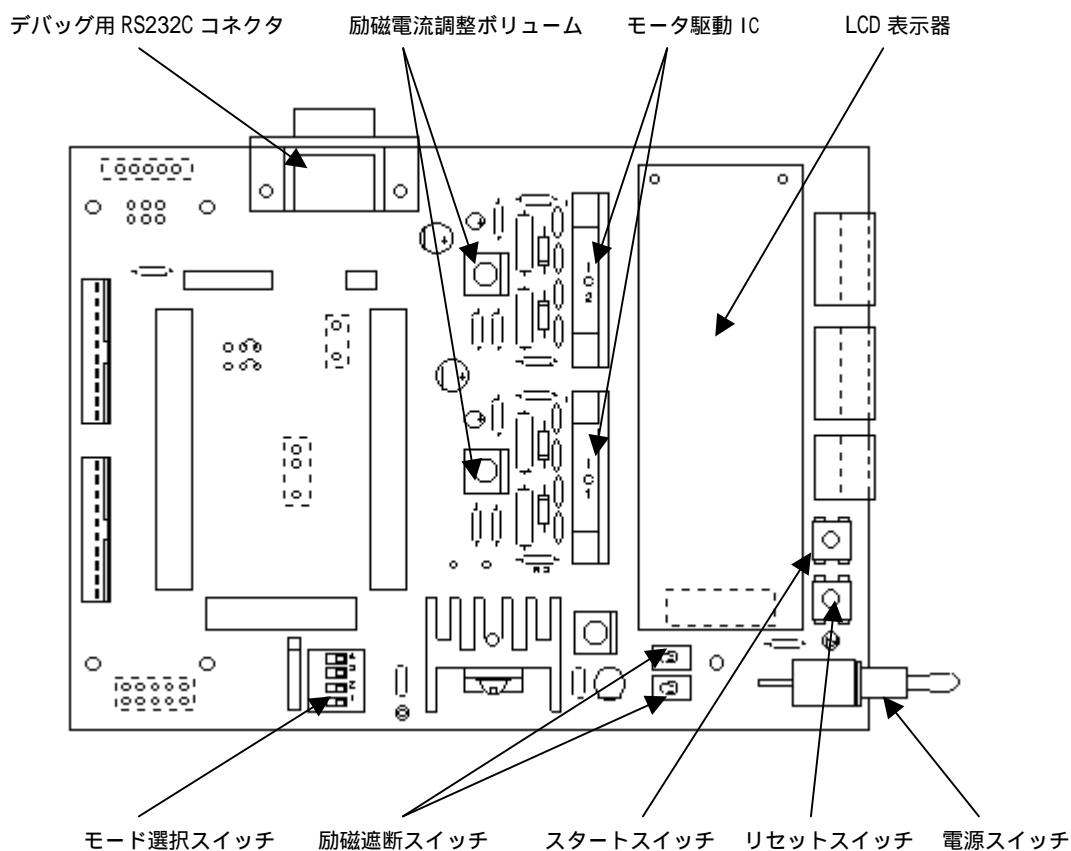


図1 モータドライブボード構成図

(1)、励磁電流調整ボリューム

左右のモータに流れる励磁電流を調整するのが100の半固定ボリュームです。この電流は、ドライバICの7番と9番ピンに接続されている1の抵抗電圧を計測すれば電圧値が電流値と1対1に対応する事から判ります。OAKS-LABOでは、この電圧をA/D変換器を介してLCD表示器でモータ電流として表示します。左右のモータ電流は約500[mA]程度になるように調整します。詳しくは、調整方法の項目をご覧ください。

(2)、励磁遮断スイッチ

励磁遮断スイッチは、停止時やプログラムのデバッグ時でのモータを駆動する必要がない場合にバッテリーの消耗を防ぐ為のスイッチで、左右のモータ駆動回路に取り付けられています。スイッチをオフにするとモータ電流はゼロになります。

(3)、LCD表示器

LCD表示器は、2行×16桁タイプのキャラクターコード付きを使用しています。このLCD表示器の仕様とキャラクターコード表は付録4をご覧ください。図2は、LCD表示器とCPUポートの接続を示したものです。8ビットのデータは、ポート7のP70～P73の4ビットのデータ線を通して下位4ビットと上位4ビットに分けて2回転送する方式です。

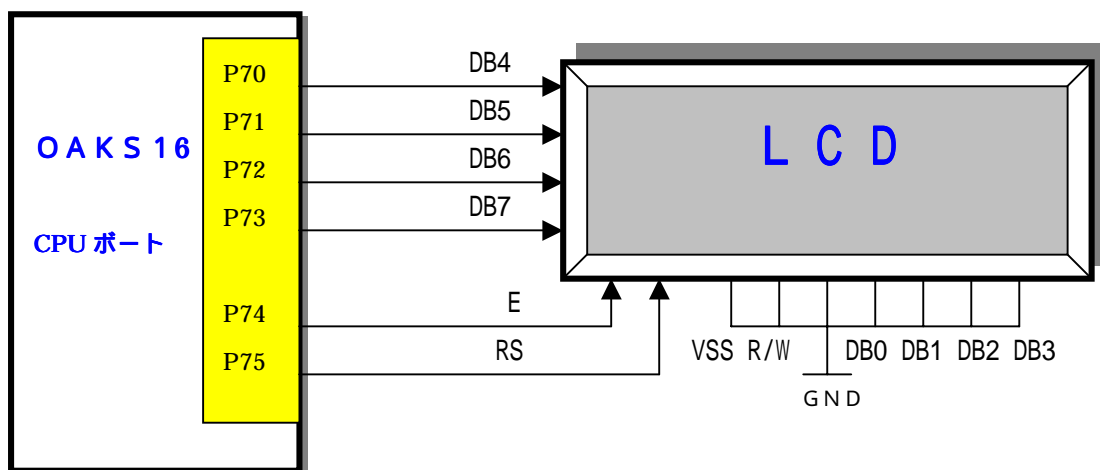


図2 LCD表示器とCPUポートの接続

LCD表示には次のような情報を表示します。

左右モータ電流値の表示

モード選択スイッチ状態の表示

速度選択、停止ライン選択、白ライン走行と黒ライン走行選択

走行状態表示

ラインレース走行情報、右接触センサ情報、左接触センサ情報

走行時間表示

(4)、その他のスイッチ

電源スイッチ

電源スイッチをONすると、バッテリーから電圧12[V]がモータドライブ回路に供給されます。また、3端子レギュレータを介して電圧5[V]がCPUポートに供給します。

スタートスイッチ

OAKS-LABOをスタートするスイッチです。

リセットスイッチ

このスイッチを押すとCPUポート「OAKS16」にリセットがかかります。

5. センサボード

センサボードの構成図を図3に示します。センサボードは独立した基板になっており、取り外しが可能です。回路図は付録3でご覧下さい。ロームのフォトレフレクタRPR-220を4個使用し、反射出力をOPアンプによるコンパレータ回路を介してマイコンに入力します。

(1)、感度調整ボリューム

反射光が入力されるとフォトトランジスタに電流が流れ、エミッタに接続された抵抗に電圧が生じます。この電圧と感度調整用ボリュームの電圧をコンパレータに入力し、感度調整を行います。コンパレータの出力側には赤色発光ダイオード(LED)を接続してありますので、センサボード単独で感度調整と動作確認ができます。

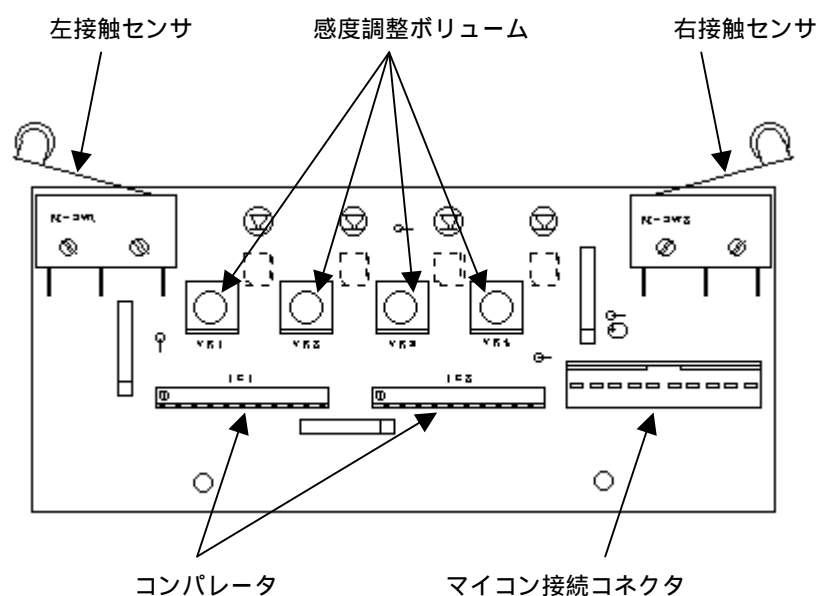


図3 センサボード構成図

(2)、フォトレフレクタの接続

センサボードに取り付けられている4個のフォトレフレクタは、図4に示されるように、コンパレータを介してCPUボードのポート4(P40~P43)に接続しています。

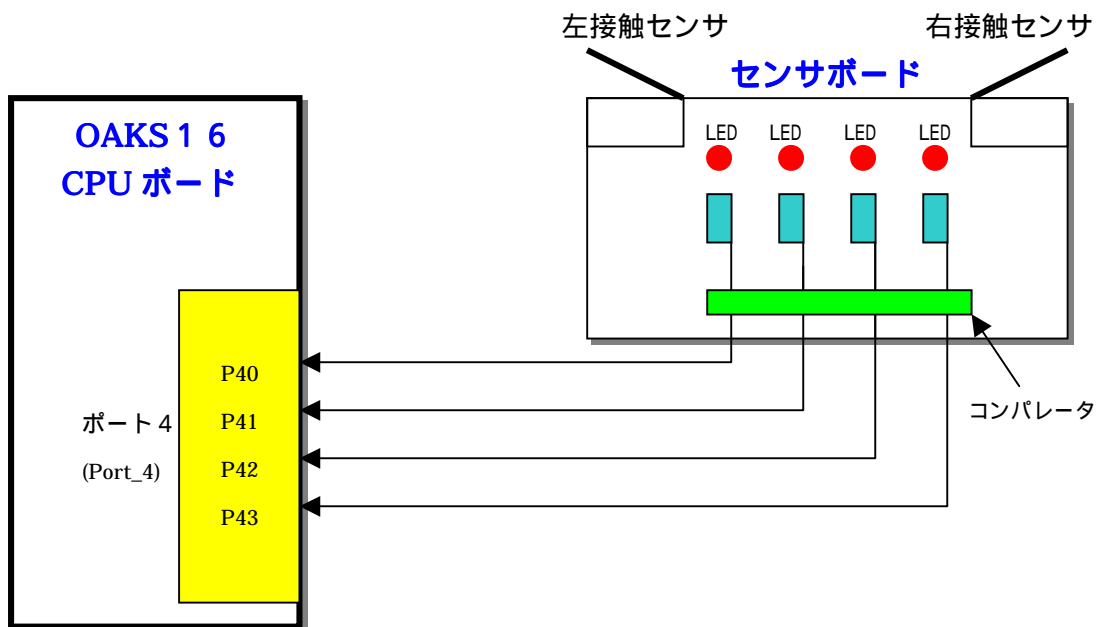


図4 フォトレフレクタの接続

(3)、接触センサ

接触センサには、オムロンのリミットスイッチ SS-01GL2-F が2個それぞれ基板の両端に取り付けられています。

図5で示されるように、左端の接触センサ出力は、CPUボードの外部割込みINT0に接続されます。また、右端の接触センサ出力は、CPUボードの外部割込みINT1に接続されます。

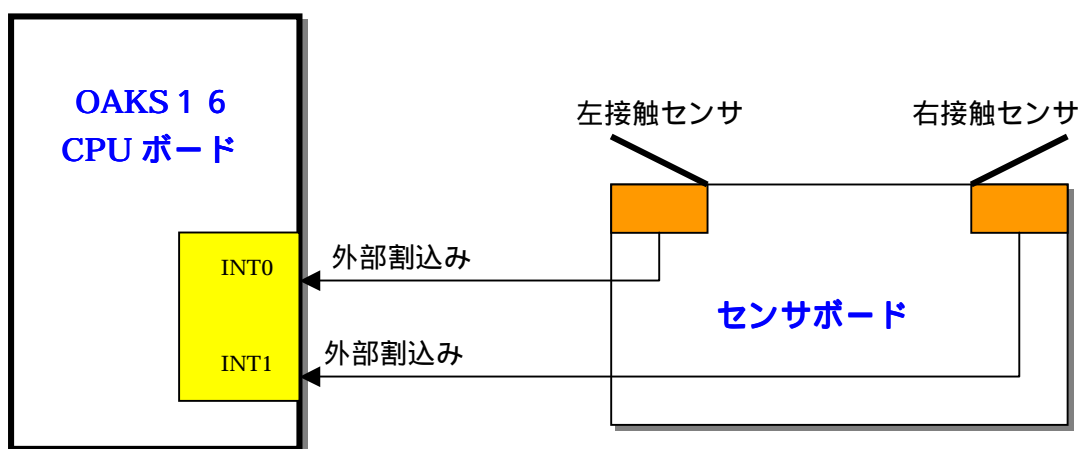


図5 接触センサとCPUボードの接続

6. CPUボード

図6はM16C/62シリーズのM30620FCAFPを搭載したOAKS16マイコンボードです。このボードには128KBのフラッシュROMと10KBのRAMがCPUに内蔵されています。

RAM領域は、400h~02BFFhまでの10KB空間に割り当てられます。ただし、後部の1.8KB空間はデバッガのMonitor Programとして使用されるのでユーザは使用できません。また、フラッシュROM領域は、E0000h~FFFFFFhまでの128KB空間に割り当てられます。ただし、ユーザ領域はE0000h~FBE00h(111.5KB)空間に限定されます。FBE00h以降はデバッガが使用するモニタ領域、割り込みの為にベクタエリアとして使用されています。

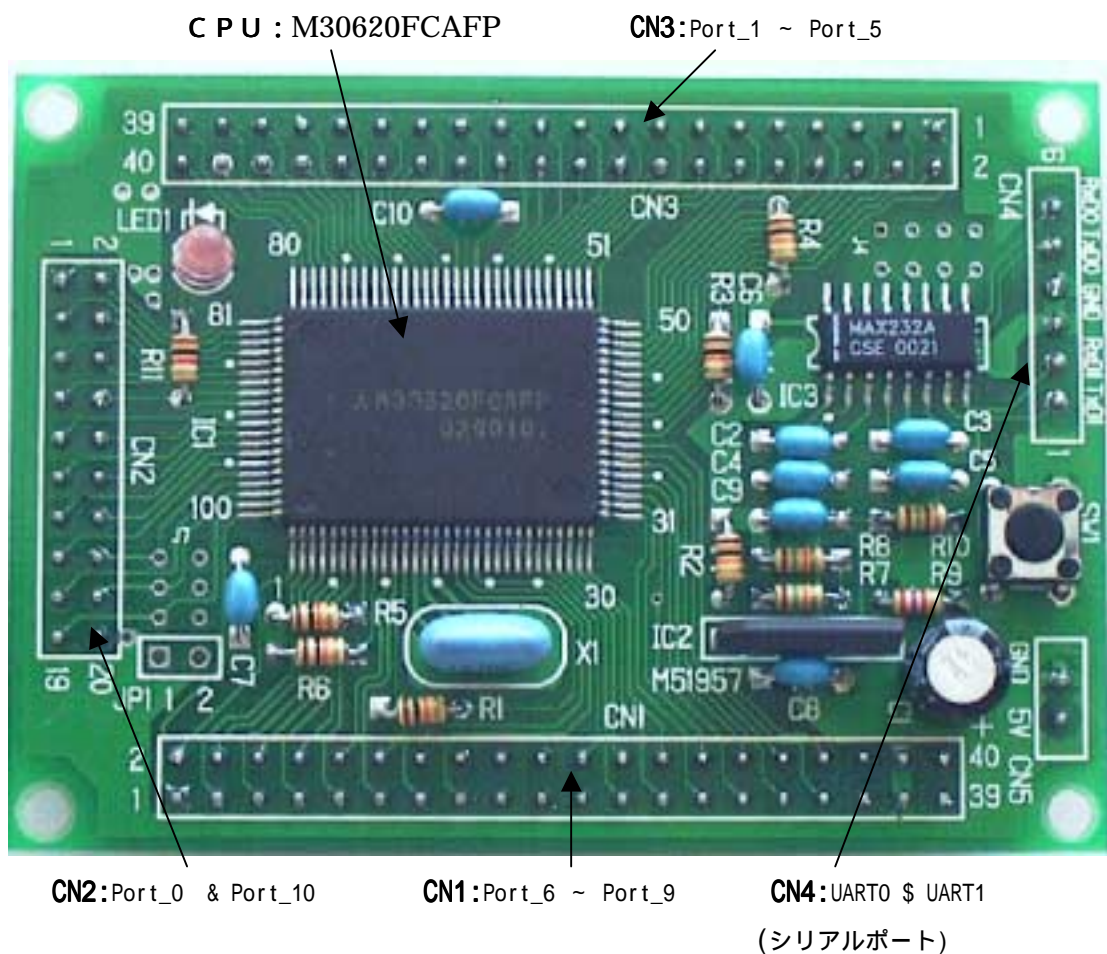


図6 OAKS16 CPUボード

(1)、パラレル入出力ポート

パラレル入出力ポートは、最大で11ポート(88bit)使用できます。実際は、ポート端

子がシリアルポート、A/Dコンバータの入力端子、D/Aコンバータの出力端子と併用されている端子は、その使い方で使用できない場合があります。

表3はOAKS-LABOが使用するパラレル入出力ポートの割り当てを一覧表で表したものです。

ポート1 (Port_1)のビットP10とビットP11は、右側モータの駆動用として使用します。また、ポート2 (Port_2)のビットP20とビットP21は、左側モータの駆動用として使用します。

走行線からのラインセンサ情報は、ポート4 (Port_4)のビットP40～ビットP43から入力します。

表3 入出力ポートの割り当て

| 入出力ポート名 | | 割り当て機能 |
|---------|-----------|-----------|
| Port_1 | P10 ~ P11 | 右側モータドライバ |
| | P12 ~ P17 | 未使用 |
| Port_2 | P20 ~ P21 | 左側モータドライバ |
| | P22 ~ P27 | 未使用 |
| Port_3 | | 拡張用空きポート |
| Port_4 | P40 ~ P43 | ラインセンサ |
| | P44 ~ P45 | 未使用 |
| | P46 ~ P47 | 接触センサ |
| Port_5 | P50 ~ P53 | モードスイッチ |
| | P54 ~ P56 | 未使用 |
| Port_7 | P70 ~ P73 | LCD 表示器 |
| | P73 | 未使用 |

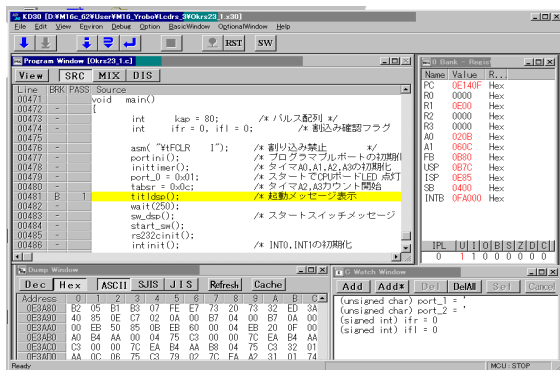
(2)、シリアルポート

シリアルポートは、2チャンネル(RS232Cドライバ実装済み)が標準装備されています。チャンネル1(Tx1, Rx1)は、リモートデバッグKD30で使用できます。また、チャンネル0(Tx0, Rx0)は、オプションの双方向RS232C無線ユニットまたは単方向RS232C無線ユニットを用いた遠隔操作として使用します。図7は、シリアルポートの接続を示したものです。チャンネル1は9ピンのD-SUBコネクタが標準で実装されています。パソコンのRS232Cとストレートケーブルで接続し、リモートデバッグKD30を起動します。また、チャンネル0は、オプション使用で、無線ユニットに専用の接続ケーブルが付属しております。

(オプション)
 双方向 RS232C 無線ユニット



リモートデバッガ KD30



(オプション)
 チャンネル 0
 (Tx0, Rx0)

シリアルポート

チャンネル 1 (Tx1, Rx1)

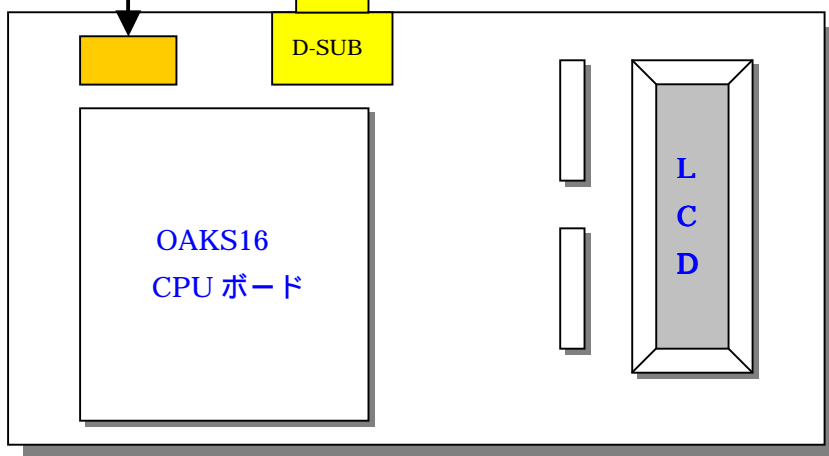


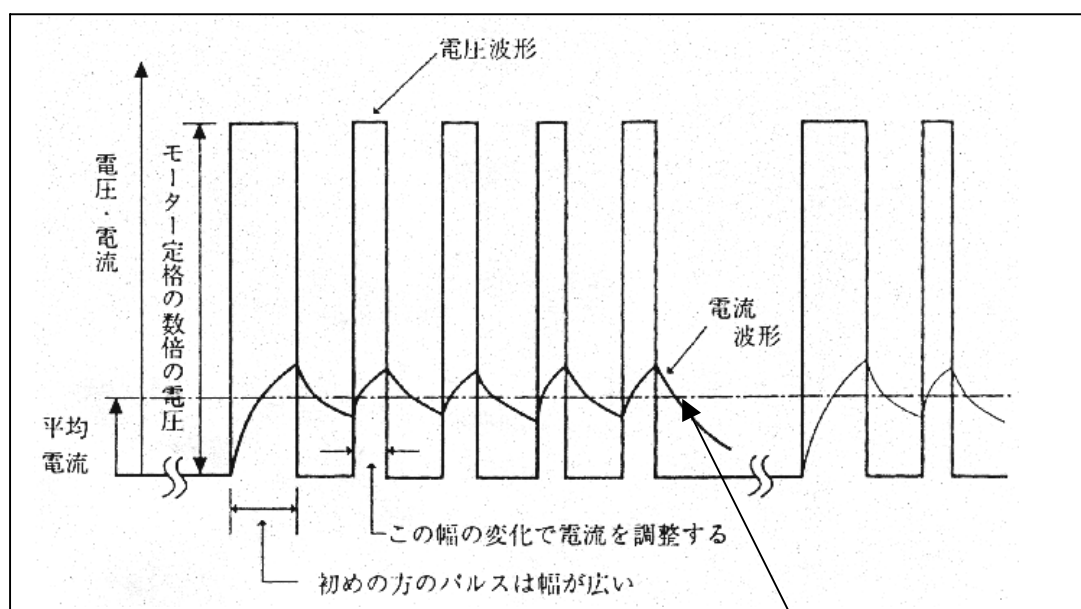
図7 シリアルポートの使用例

7.OAKS-LABO の調整方法

(1)、モータ電流の調整

OAKS-LABOには、バッテリーに12[V] 1100[mA]ニッカド電池を使用します。モータを2相励磁方式で走行させる場合の電源電流を求めます。

このモータを2相励磁方式で使用すると定格電流は $0.47 \times 2 = 0.94$ [A]になります。しかし、チョッパ回路により電流を供給しているため電源からの電流が 0.94 [A]流れるわけではありません。電源電圧とモータに流れる電流の関係を図8に示します。電源電圧が高い程、電源から出る平均電流は低くなります。電源電圧12[V]を与えると、この電源より流れる励磁電流をモータ定格電流の2分の1($0.94 \div 2 = 0.5$ [A])に調整します。電流波形は図のように平均値から ± 40 [mA]程度変動しています。



電流波形は ± 40 mA程度変動しています。

図8 PWM波形のモータ電流

次の手順で調整して下さい。

電源スイッチをオンにします。LCD表示器には「Welcome to OAKS-LABO」メッセージが数秒間表示後、図9のような表示画面に切り替わります。

左右モータ用の2つの励磁遮断スイッチを共にOFFします。この時、LCD表示器の電流値は「L = 000 mA R = 000 mA」のように共に0 [mA]となっている事を確認して下さい。

左モータ用励磁遮断スイッチをONにします。図9で示されるように、左モータ

励磁電流設定ボリュームを調整し、左モータ電流が約「 $L = 500 \text{ mA}$ 」となるように調整して下さい。電流値は、常に $\pm 40 [\text{mA}]$ 程度変動変動しています。ロボット自身を走行させるだけですがトルクは最大を得る必要はありません。したがって、 500 mA 程度が表示されればOKです。

右モータ用励磁遮断スイッチをONにします。図9で示されるように、右モータ励磁電流設定ボリュームを調整し、右モータ電流が約「 $R = 500 \text{ mA}$ 」となるように調整して下さい。電流値は、常に $\pm 40 [\text{mA}]$ 程度変動変動しています。変動の中間値として $500 [\text{mA}]$ 程度が表示されればOKです。

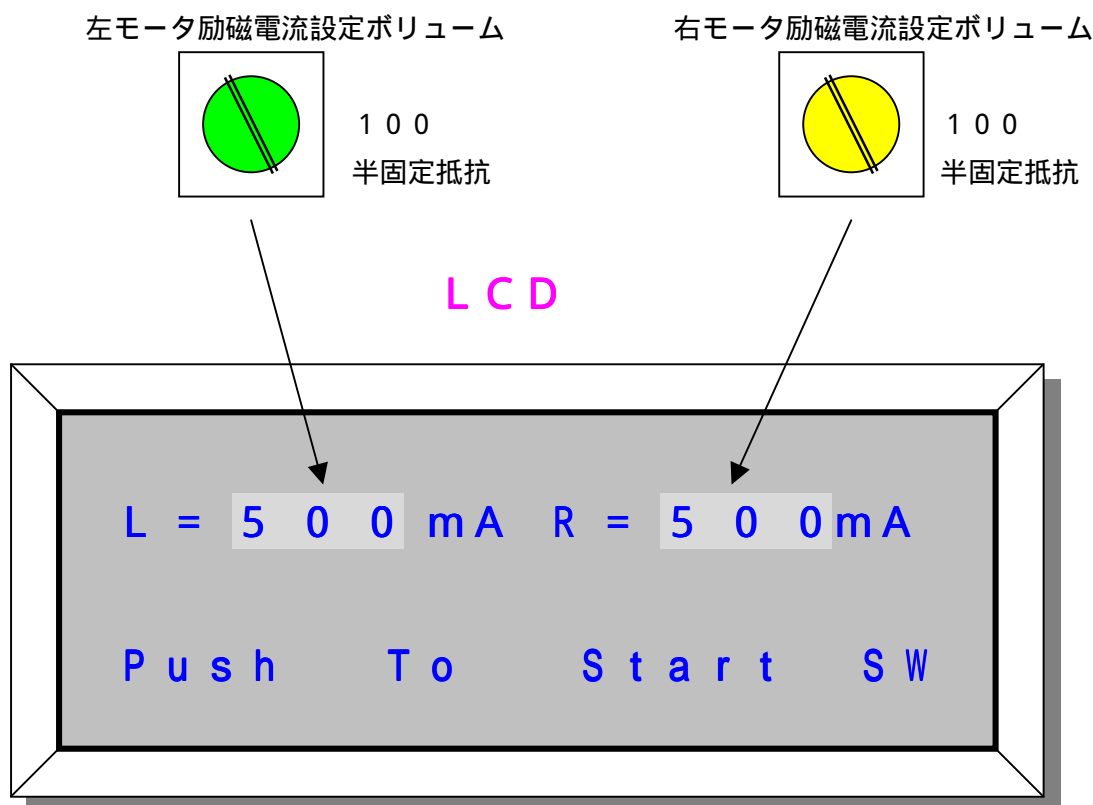


図9 モータ電流の調整

(2)、ラインセンサの調整

ラインセンサポートを車体に取り付けた状態でコースからセンサの先端までの間隔が約 $5 [\text{mm}]$ から $7 [\text{mm}]$ 間になっているか確認します。また、4つのセンサが全てコースと平行になっている事も確認して下さい。確実に走行するために2つのボールキャストが少し浮き気味になっています。次の手順で4つのラインセンサを調整して下さい。調整が「あまい」とカーブでのコースアウトの原因となります。確実に調整して下さい。

白ラインコースの場合。

ロボットがコースから外れた位置すなわち黒エリアの所に置き、図10で示されるNo1～No4の感度調整ボリュームを調整し、センサボード上の各センサに対応するLEDが点灯から消灯する位置に調整して下さい。ロボットをボールキャストの浮いている範囲で前後に揺らして全てのLEDが点灯しない事を確認して下さい。

次に、ロボットのNo1センサを白ライン上に置き、LEDが点灯する事を確認します。もし、点灯しない場合には、No1感度調整ボリュームを再度調整し、LEDが消灯から点灯する位置に調整します。

同様にNo2からNo4センサを順番に白ライン上に置き、LEDが点灯する事を確認します。

白ライン上でもロボットをボールキャストの浮いている範囲で前後に揺らして全てのLEDが点灯する事を確認して下さい。

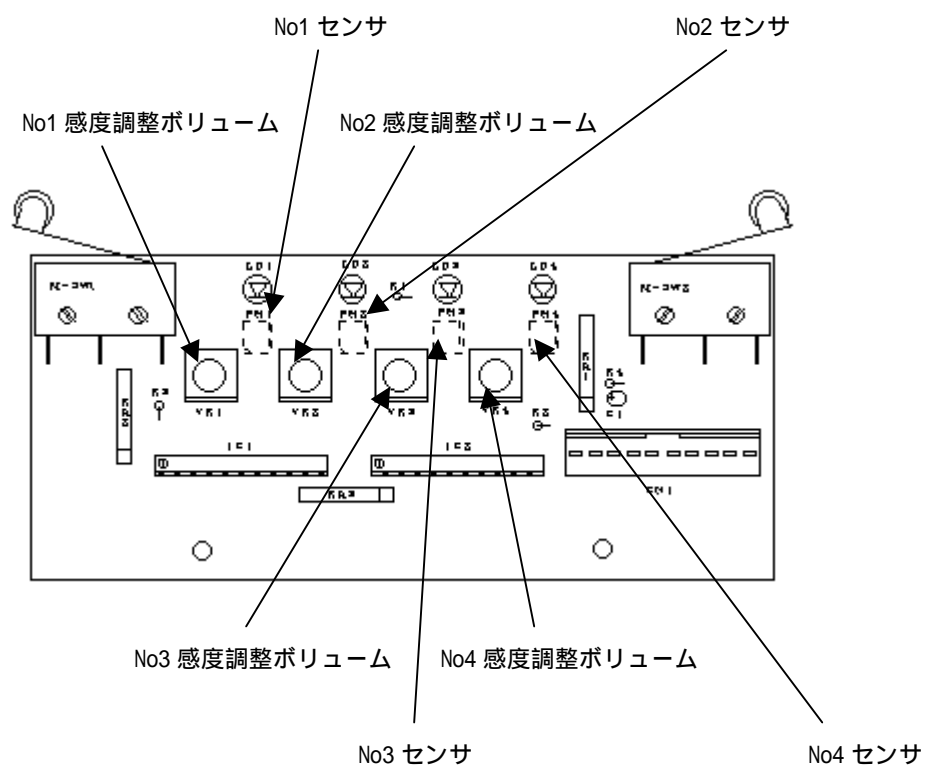


図10 ラインセンサの調整

黒ラインコースの場合。

ロボットがコースから外れた位置すなわち白エリアの所に置き、図10で示されるNo1～No4の感度調整ボリュームを調整し、センサボード上の各

センサに対応するLEDが消灯から点灯する位置に調整して下さい。ロボットをボールキャストの浮いている範囲で前後に揺らして全てのLEDが点灯する事を確認して下さい。

次に、ロボットのNo 1センサを黒ライン上に置き、LEDが消灯する事を確認します。もし、消灯しない場合には、No 1感度調整ボリュームを再度調整し、LEDが点灯から消灯する位置に調整します。

同様にNo 2からNo 4センサを順番に黒ライン上に置き、LEDが消灯する事を確認します。

黒ライン上でもロボットをボールキャストの浮いている範囲で前後に揺らして全てのLEDが消灯する事を確認して下さい。

以上の調整手順を二三回繰り返し、完全にラインセンサを調整します。また、異なったコースで走行させる場合にも、その都度、確実に調整して下さい。

8.OAKS-LAB0 の走行テスト

OAKS - LAB0には、走行テストとデモを兼ねたプログラムが書き込まれています。ライントレースロボットとして走行させる場合には、コースが必要です。床または卓上が白色の場合には、黒のビニールテープを付録5のコース例を参考に貼って下さい。また、床または卓上が黒色の場合には、白のビニールテープを付録8のコース例を参考に貼って下さい。

コースはなるべく段差を少なくして下さい。段差があると車輪の空回りやボールキャストの引っ掛かりなど、ロボットがうまく動かない場合があります。また、紙でコースを作ると車輪のゴムに紙の繊維が付着してグリップ力が無くなり空回りの原因になります。

走行テストの前にモータ電流とラインセンサの調整をしておきます。また、付属のバッテリーパックの充電は、できる限り専用の充電器を使用して下さい。もし、手元に無い場合には、DC 14V ~ 16V、200 ~ 300mA程度(定格110mA 15時間)でゆっくり充電して下さい。

(1)、モード選択スイッチの割り当てと機能

モード選択スイッチは、表4のように4つのスイッチの状態によって、速度の設定、オールセンスの判定による停止または通過の設定、白ライン走行または黒ライン走行の設定ができます。

スイッチ1番と2番の機能

スイッチ1番と2番の4通りの組み合わせによって、表4のように低速、中低速、中高速、高速の4段階に設定できます。設定変更は、電源を投入する前またはLCD表示器に表示されるスタートまでの10秒間以内に行います。

スイッチ3番の機能

3番のスイッチは、オールセンス時での動作を選択します。“ON”で使用する場

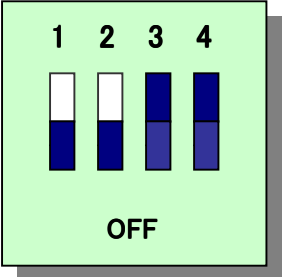
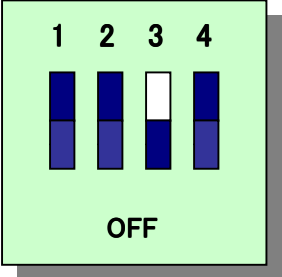
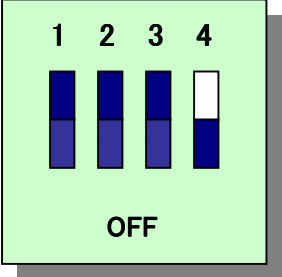
合には、オールセンスの無いコース(交差点の無いコース)で走行して下さい。ストップライン(4つのセンサが同じに消灯(黒ライン走行)または点灯(白ライン走行))で停止します。

スイッチを“OFF”で使用する場合には、交差点などオールセンスのあるコースを走行させることができます。設定変更は、電源を投入する前またはLCD表示器に表示されるスタートまでの10秒以内に行います。

スイッチ4番の機能

4番スイッチは、“ON”にすると白地に黒ラインのコースで走行できます。また、“OFF”で使用すると黒地に白ラインで走行できます。設定変更は、電源を投入する前またはLCD表示器に表示されるスタートまでの10秒以内に行います。

表4 モード選択スイッチの割り当てと機能

| モード選択スイッチ | 機 能 | | | | | | | | | | | | | | | | | | | | |
|---|--|--------|------|--------|------|-----|-----|----|----|-----|----|----|-----|----|-----|----|-----|----|----|----|----|
|  | <p>モード選択スイッチの1番と2番は走行速度を選択します。</p> <table border="1" style="margin-left: auto; margin-right: auto;"> <thead> <tr> <th>1</th> <th>2</th> <th>LCD 表示</th> <th>走行速度</th> </tr> </thead> <tbody> <tr> <td>OFF</td> <td>OFF</td> <td>Lo</td> <td>低速</td> </tr> <tr> <td>OFF</td> <td>ON</td> <td>MI</td> <td>中低速</td> </tr> <tr> <td>ON</td> <td>OFF</td> <td>Mh</td> <td>中高速</td> </tr> <tr> <td>ON</td> <td>ON</td> <td>Hi</td> <td>高速</td> </tr> </tbody> </table> | 1 | 2 | LCD 表示 | 走行速度 | OFF | OFF | Lo | 低速 | OFF | ON | MI | 中低速 | ON | OFF | Mh | 中高速 | ON | ON | Hi | 高速 |
| 1 | 2 | LCD 表示 | 走行速度 | | | | | | | | | | | | | | | | | | |
| OFF | OFF | Lo | 低速 | | | | | | | | | | | | | | | | | | |
| OFF | ON | MI | 中低速 | | | | | | | | | | | | | | | | | | |
| ON | OFF | Mh | 中高速 | | | | | | | | | | | | | | | | | | |
| ON | ON | Hi | 高速 | | | | | | | | | | | | | | | | | | |
|  | <p>モード選択スイッチの3番は通過と停止を選択します。</p> <p>ON・・・減速・停止します。(LCD表示: “St”) オールセンスの無いコースで、ゴール線で停止。</p> <p>OFF・・・通過します。(LCD表示: “Pa”) 交差点等、オールセンスが在るコース。</p> | | | | | | | | | | | | | | | | | | | | |
|  | <p>モード選択スイッチの4番は白ラインと黒ラインを選択します。</p> <p>ON・・・白地に黒ラインの場合。 (LCD表示: “Bk”)</p> <p>OFF・・・黒地に白ラインの場合。 (LCD表示: “Wh”)</p> | | | | | | | | | | | | | | | | | | | | |

(2)、走行テスト手順

電源スイッチを“ ON ”またはリセットスイッチを押すと図 1 1 に示される表示が数秒間表示されます。



図 1 1 メッセージの表示

数秒後に図 1 2 で示されるモータ電流調整の為の表示に切り替わります。励磁遮断スイッチが共に“ OFF ”にすると「 L = 0 0 0 m A R = 0 0 0 m A 」が表示されます。左側の励磁遮断スイッチを“ ON ”にしてから左モータ用の励磁電流調整ボリュームを廻して、図 1 2 のように約 L = 5 0 0 m A となるように調整します。電流値は ± 4 0 m A 程度常に変動しています。故障ではありません。次に右側の励磁遮断スイッチを“ ON ”にしてから右モータ用の励磁電流調整ボリュームを廻して、図 1 2 のように約 R = 5 0 0 m A となるように調整します。電流値は同様に ± 4 0 m A 程度常に変動しています。故障ではありません。調整が終了後、スタートスイッチを押して下さい。この場合、画面が切り替わる迄しばらく押し続けて下さい。

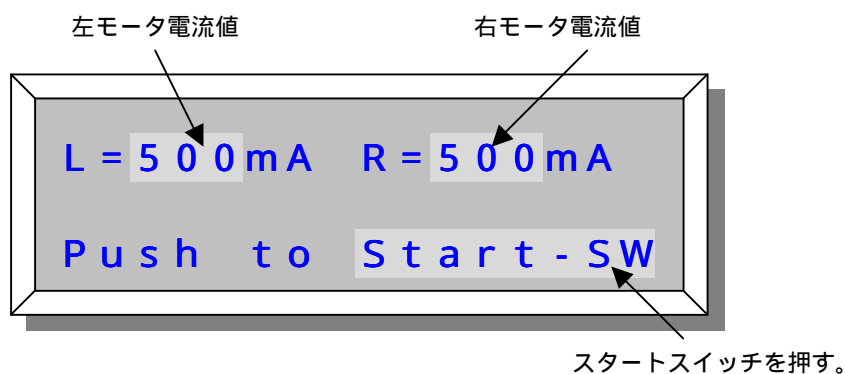


図 1 2 モータ電流の調整表示

スタートスイッチを押すと、図 1 3 で示されるモード表示画面に切り替わります。上段には、走行開始までの残り時間が表示されます。時間は、1 0 秒から開始し、0 秒になると走行を開始します。下段は、表 4 で表記されるように、モード選択ス

スイッチの状態を表示します。変更を行う場合には、走行カウンタが0秒になる迄にモード選択スイッチで行って下さい。

10秒から0秒までカウントダウンします。

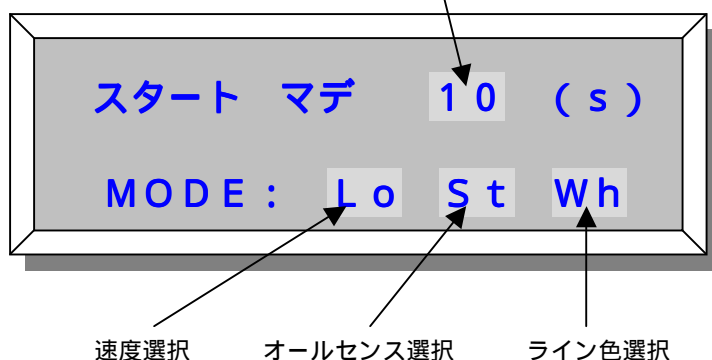


図 1 3 モード表示画面

走行カウンタが0秒になると、図14で示される走行表示画面に切り替わります。

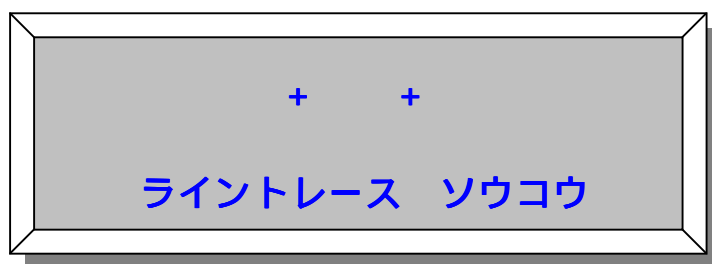


図 1 4 走行表示画面

モード選択スイッチで走行開始前に、予め停止ラインの減速・停止の設定がされていると、図15で示されるように、走行開始から停止迄の所要時間が表示されます。

再び走行させる場合には、スタートスイッチを押して下さい。 項の画面表示に戻ります。



図 1 5 走行表示画面

左接触センサが障害物に接触すると、図16で示されるように、左側が障害物に接触した表示を行い、急停止します。その後、数十センチ程度バック走行を行い完全停止します。

再び走行させる場合には、コースに戻し、スタートスイッチを押して下さい。項の画面表示に戻り再走行を開始します。

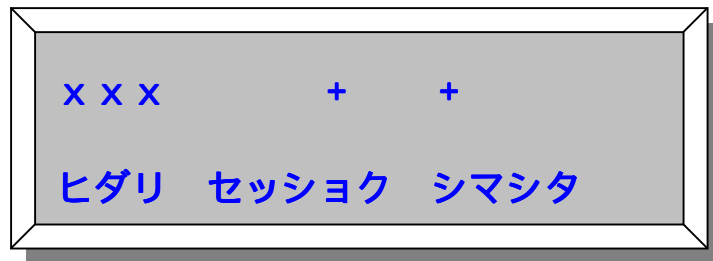


図16 左側接触の表示画面

右接触センサが障害物に接触すると、図17で示されるように、右側が障害物に接触した事を知らせる表示を行い、急停止します。その後、数十センチ程度バック走行を行い完全停止します。

再び走行させる場合には、コースに戻し、スタートスイッチを押して下さい。項の画面表示に戻り再走行を開始します。

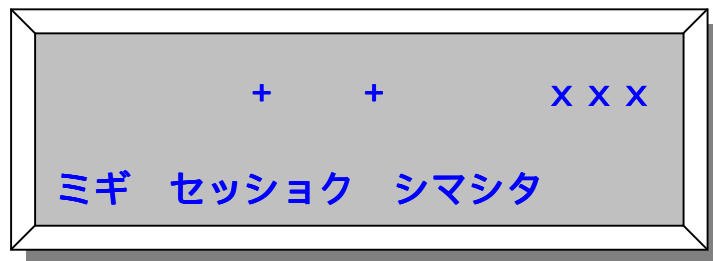


図17 右側接触の表示画面

9. OAKS - LABOのしくみ

OAKS - LABOには2個のステッピングモータを使用しています。ステッピングモータは、パラレルポートから1と0のデータを与えるだけで簡単に回転制御が可能です。また、ロボットの位置は4個のラインセンサの位置を検出し、その値から右方向に制御するか左方向に制御するかを判断し、走行ラインに沿って走行させる事ができます。

(1)、ステッピングモータの駆動方法

図18は、CPUボードとステッピングモータの接続を表したものです。右モータは、パラレルポート1 (Port_1)のビットP10とビットP11の2本で制御します。また、左モータは、パラレルポート2 (Port_2)のビットP20とビットP21の2本で制御します。

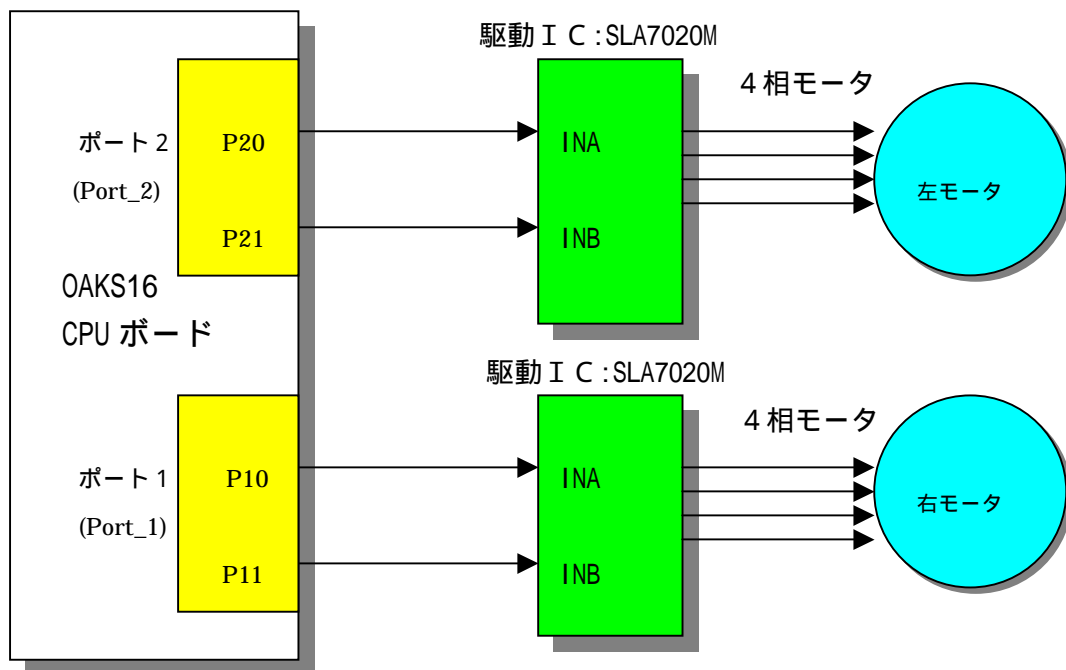


図18 CPUボードとステッピングモータの接続

各ポートの出力は、モータ駆動IC (SLA7020M)のINA端子とINB端子から入力し、IC内部で2相励磁パターンを作ります。

図19は、2相励磁パターンを作る為に各ポートから出力するタイムチャートです。

| ポート | | CW 回転 | | | |
|-------|-----|-------|---|---|---|
| Clock | | 0 | 1 | 2 | 3 |
| P0 | INA | 1 | 1 | 0 | 0 |
| P1 | INB | 0 | 1 | 1 | 0 |

(a) 時計回転

| ポート | | CCW 回転 | | | |
|-------|-----|--------|---|---|---|
| Clock | | 0 | 1 | 2 | 3 |
| P0 | INA | 0 | 0 | 1 | 1 |
| P1 | INB | 0 | 1 | 1 | 0 |

(b) 反時計回転

図 19 励磁信号タイムチャート

CPUポートのポート1(Port_1)とポート2(Port_2)の各P0(ビット0)とP1(ビット1)を用いているので、モータをCW回転させるには(a)図のように、01h 03h 02h 00hの順番に繰り返し出力します。また、CCW回転させるには(b)図のように、00h 02h 03h 01hの順番に繰り返し出力します。図20は、励磁信号タイムチャートを配列を用いてC言語で記述した例です。

```

時計方向回転データ関数名           時計方向回転データ配列
    ↓                               ↓
char cwdata(char dr)
{
    char cw_tbl[4] = {0x01,0x03,0x02,0x00};
    return cw_tbl[dr];
}

char ccwdata(char dl)
{
    char ccw_tbl[4] = {0x00,0x02,0x03,0x01};
    return ccw_tbl[dl];
}
    ↑                               ↑
反時計方向回転データ関数名       反時計方向回転データ配列

```

図 20 モータ回転データのプログラム記述例

モータの回転速度を制御するには、CPUボードから励磁信号を出力してから次の励磁信号を出力するまでの時間間隔を与えます。この時間は、一般的に次のような方法で作ることができます。

ソフトウェアタイマ方式

ステッピングモータのパルス間隔をプログラムのループによる時間稼ぎによって回転を制御する方式です。図 2 1 はソフトタイマ方式のフローチャートです。この方式では、それぞれのモータのタイマデータを交互にデクリメントしてゼロになった方のモータに回転データを出力します。モータカウント値が 0 になるまでループし、0 になったらモータを駆動し、カウント値を初期値に戻して再びカウントダウンを行います。この方法では簡単にプログラムは記述できますが、両モータを交互に制御する過程でプログラムの時間ループが終わるまで次の処理ができないので高速走行には不向きです。

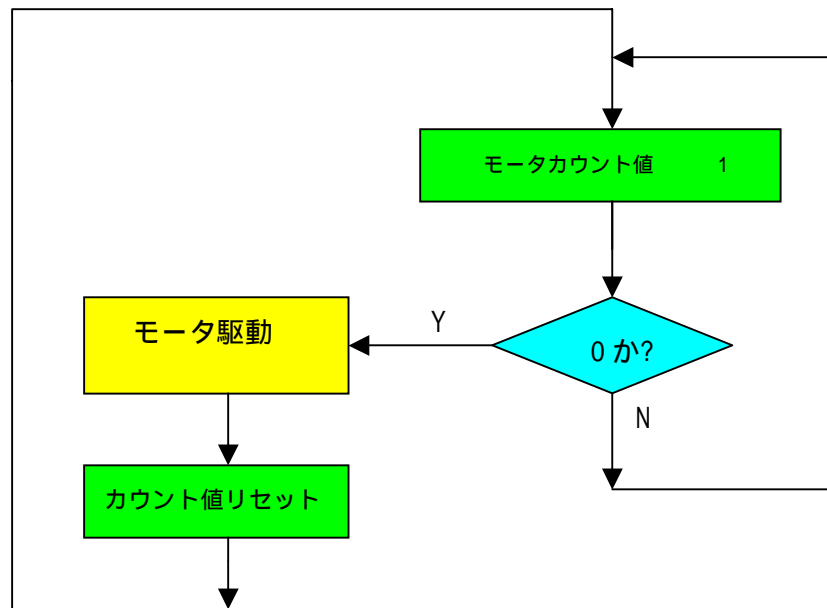


図 2 1 ソフトタイマ方式でのモータ駆動

図 2 2 は、図 2 1 で示されたソフトタイマ方式フローチャートを C 言語で記述したプログラム例です。右モータのカウント値を入れる変数が `R_CONST P` です。また、左モータのカウント値を入れる変数は `L_CONST P` です。それぞれのカウント値が“0”になるとモータを駆動し、カウントデータを初期値 `C O N S T P` に戻して無限ループから抜け出します。

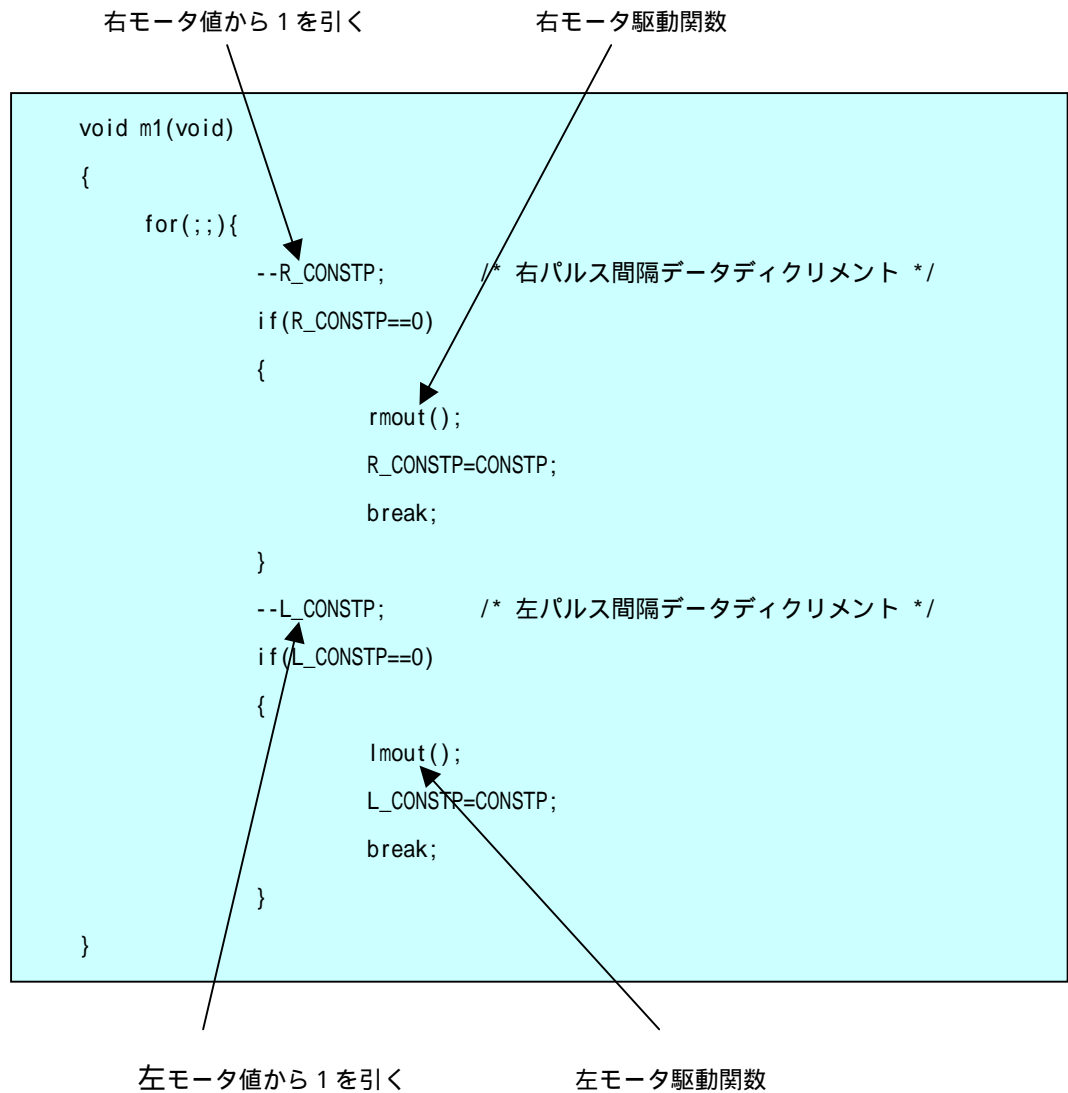


図 2 2 ソフトタイマ方式のプログラム記述例

タイマ割り込み方式

左右のモータをそれぞれ別々のタイマ割り込みによって回転速度を制御する方式です。図 2 3 はタイマ割り込み方式のフローチャートです。

この方式は、プログラムにタイマ割り込みの知識が必要となりますが、左右のモータに割り当てたタイマの時間定数を変更することで独立して回転速度を制御できます。割り込みのあり、なしの判断に割り込みフラグを使用します。タイマ A をタイマモードに設定し、その割り込み先をモータ駆動にします。時間定数値をセットし、割り込みがある迄待ちます。割り込みが発生し、モータを駆動してフラグを 1 にセットします。時間定数に加速データを与える事により、より高速走行が可能となります。

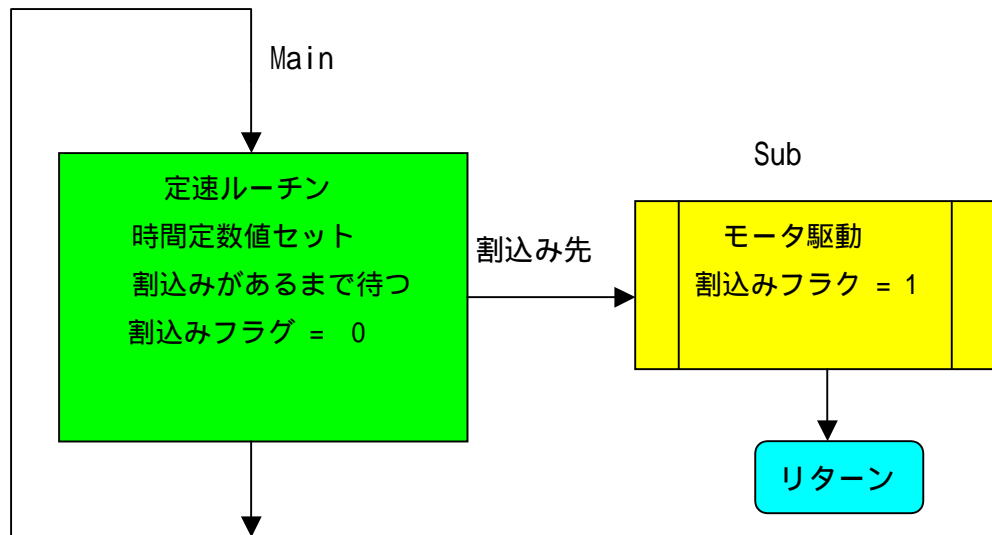


図 2 3 タイマ割り込み方式でのモータ駆動

図 2 4 (a) は、図 2 3 図で示されるフローチャートを C 言語で記述したプログラム例です。右モータの時間設定を行うタイマ変数が t a 0 です。また、左モータの時間設定を行うタイマ変数は t a 1 です。直進する場合には、2 つのタイマ変数に規定値 MAXSPEED を設定します。右モータ駆動割り込みが終了すると割り込みフラグ「i l r」を「1」にします。また、左モータ駆動割り込みが終了すると割り込みフラグ「i f l」を「1」にします。したがって、無限ループで何れかのフラグが「1」になるまで待ちます。最後に 2 つのフラグを「0」に戻して割り込みを可能にします。

```

Void noncr(void) /* 直進 ルーチン */
{
    ta0 = MAXSPEED; /* 右定速度回転 */
    ta1 = MAXSPEED; /* 左定速度回転 */
    for(;;){
        f(ifr == 1 | ifl == 1) break;
    }
    ifr=0;
    ifl=0;
}

```

規定値

(a) 直進ルーチン

図 2 4 (b) は、ロボットを少し右に修正する場合のプログラム例です。直進する場合には、右モータのタイマ変数には規定値 MAXSPEED と可変値 COR 1 D _L

を加えます。また、左モータのタイマ変数は規定値 MAXSPEED と可変値 COR1D_H を加えます。

可変値は、COR1D_L > COR1D_H となる値を与えるので、右モータのタイマ値が左モータのタイマ値よりも大きくなります。したがって、右モータの回転速度が左モータの回転速度よりも遅くなり、ロボットは右方向に進みます。

左右2つの可変値を組み合わせると、方向を自由に変える事が可能になります。

ta0 > ta1 となるように設定。

```
void rc1(void) /* 少し右修正 ルーチン */
{
    ta0 =MAXSPEED+COR1D_L; /* 右モータを少し低速に設定 */
    ta1 =MAXSPEED+COR1D_H;
    for(;;){
        if(ifl == 1) break;
    }
    ifl=0;
}
```

(b) 少し右修正ルーチン

図 2 4 タイマ割り込み方式のプログラム記述例

(2)、ラインセンサ入力

コース上のロボットの位置と 4 個のセンサ反応の位置関係を図 2 5 に示します。センサの反応は、コース上(白)で " 0 " (Lレベル)、コースから外れた場所では " 1 " (Hレベル)の出力になります。このデータをビット反転してコース上で " 1 "、コースから外れた場所では " 0 " にして判定した真理値表が表 5 です。

ロボットがコースに対して " かなり右寄り " の位置では、センサ出力は一番左側のセンサがコースに位置しているので、 " 0 " となり、残りの 3 のセンサ出力は " 1 " となります。したがって全出力は " 0 1 1 1 " となりますが、表 5 では、このデータをビット反転したデータ " 1 0 0 0 " で表現します。

ロボットを停止させるには、4 個全てのセンサが " 1 1 1 1 " となるように、コースに対して直角に 4 個のセンサの距離長分のラインテープを貼れば、ロボットはその位置で停止します。

制御の考えは車の運転と同じで右に寄ったら左側のモータの回転を下げればロボットは左に曲がります。また逆に、左に寄ったら右側のモータの回転を下げればロボットは右に曲がります。

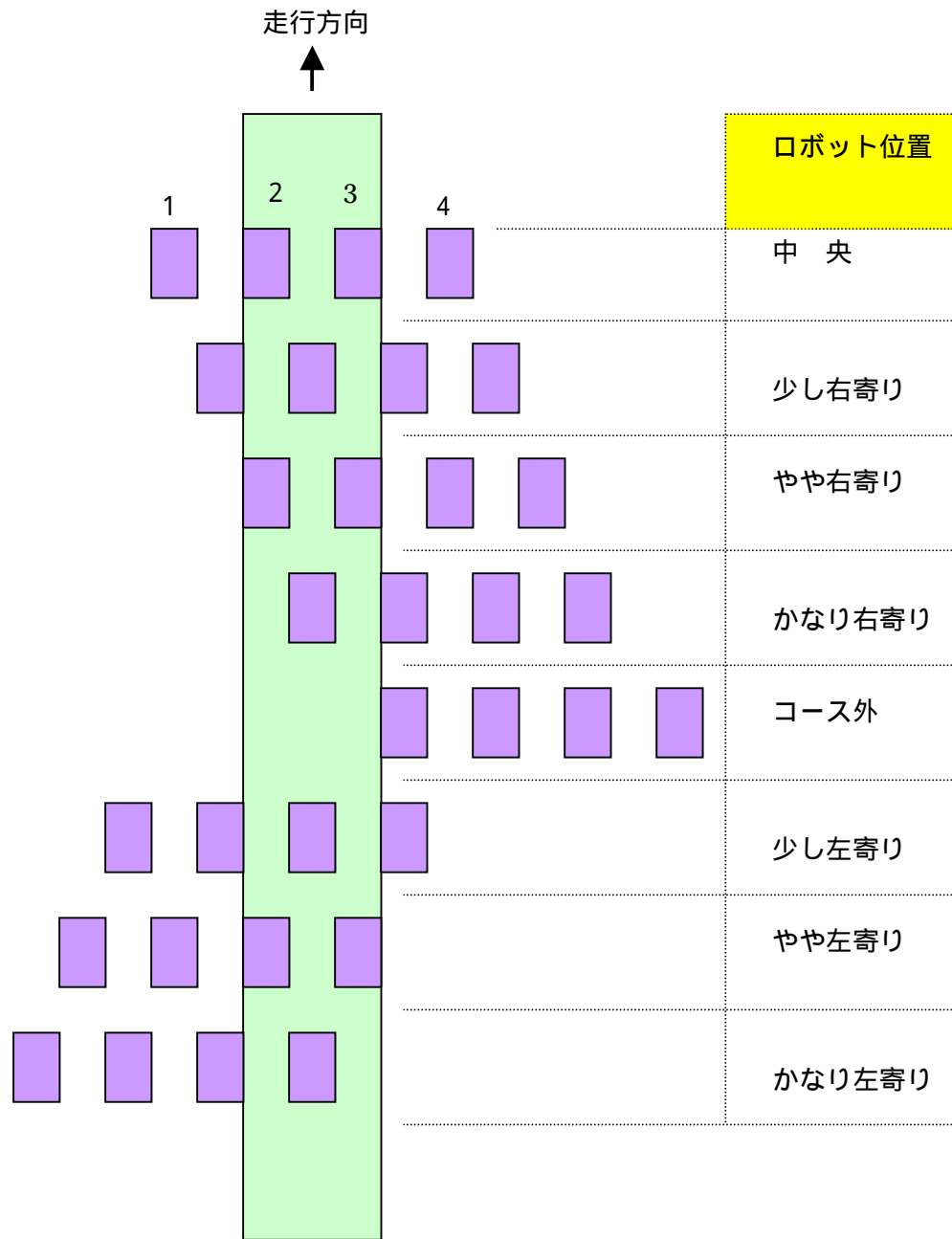


図 2 5 ロボットの位置とセンサの関係

図 2 6 は、表 5 を元に C 言語で記述した位置修正プログラム例です。センサからの入力は `sence()` 関数です。このデータを `switch` 文でロボットの位置を判別します。例えば、センサデータが " 1 " ならば、`case 1` 文を実行します。ロボットは、かなり右に寄っているので `lc3()` 関数でかなり左に修正します。

表5 コース位置判定真理値表

| ロボットの位置 | センサ出力データ (ビット反転出力) | | | | |
|---------|-----------------------|---|---|---|------|
| | 2進数 | | | | 16進数 |
| | 0 | 1 | 2 | 3 | |
| 中央 | 0 | 1 | 1 | 0 | 06h |
| 少し右寄り | 0 | 1 | 0 | 0 | 02h |
| やや右寄り | 1 | 1 | 0 | 0 | 03h |
| かなり右寄り | 1 | 0 | 0 | 0 | 01h |
| はずれ | 0 | 0 | 0 | 0 | 00h |
| 少し左寄り | 0 | 0 | 1 | 0 | 04h |
| やや左寄り | 0 | 0 | 1 | 1 | 0Ch |
| かなり左寄り | 0 | 0 | 0 | 1 | 08h |

```

void teisoku() /* 定速ルーチン */
{
    int sed;
    sed = sence();
    switch(sed)
    {
        case 1: lc3(); break; /* かなり左修正 */
        case 2: lc1(); break; /* 少し左修正 */
        case 3: lc2(); break; /* やや左修正 */
        case 4: rc1(); break; /* 少し右に修正 */
        case 6: noncr(); break; /* 中央 */
        case 8: rc3(); break; /* かなり右修正 */
        case 12: rc2(); break; /* やや右に修正 */
        case 15: stop_line(); break; /* ストップライン */
        default:
            noncr(); break; /* 何れでもない */
    }
}

```

センサからのデータ → sed

スイッチ文で位置を判別 → switch(sed)

図26 位置修正プログラム例

10. 添付ファイルの構成と使い方

OAKS-LABOには表6で示されるファイルがCD-Rに入っています。目的に応じて自由に取り出してご使用下さい。ただし、添付されている全てのC言語プログラムには著作権が存在しますので商用として使用する事はできません。

表6 OAKS-LABO添付ファイル一覧表

| ディレクトリ名 | ファイル名 | 内 容 |
|-----------|----------------------------|--|
| Demo_1 | Oklabo_1.c Oklabo_1.x30 | ソフトタイマ方式ソースプログラム。 ソフトタイマ方式デバッグファイル。 |
| Demo1_rom | Oklabo_2.c Oklabo_2.mot | ソフトタイマ方式ROM化ソースプログラム。 ソフトタイマ方式ROM化ファイル。 |
| Demo_2 | Oklabo_3.c Oklabo_3.x30 | タイマ割込み方式ソースプログラム。 タイマ割込み方式デバッグファイル。 |
| Demo2_rom | Oklabo_4.c Oklabo_4.mot | タイマ割込み方式ROM化ソースプログラム。 タイマ割込み方式ROM化ファイル。 |
| Document | OAKSLABO 取扱説明書 | OAKS-LABO ユーザーズマニュアル。 (PDF ファイル形式) |
| Monitor | Mon_main.mot | デバッガ起動用モニタプログラム。 |

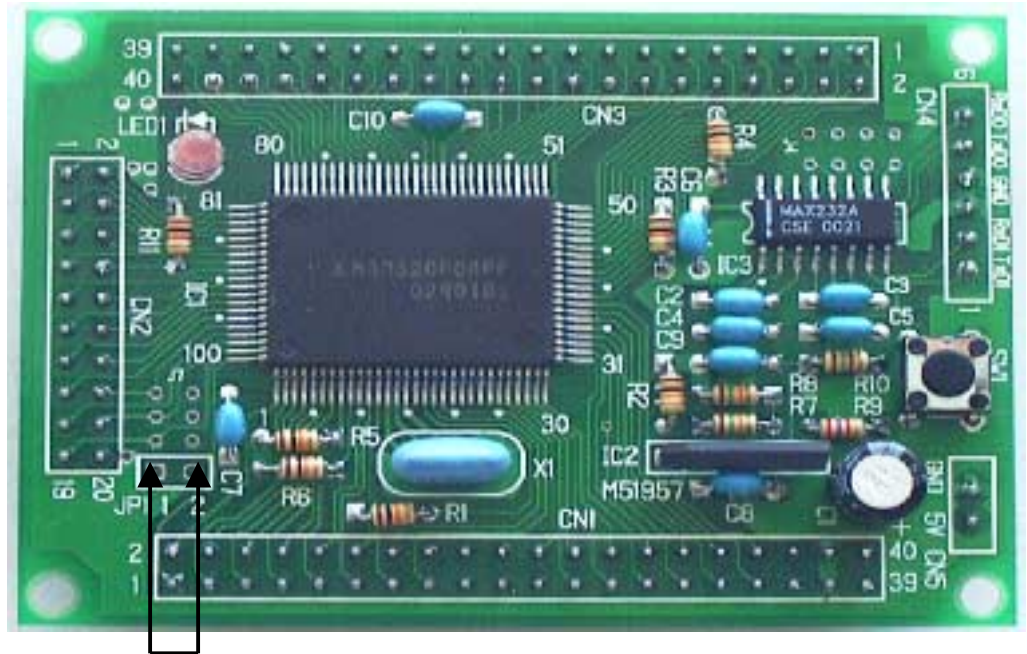
ファイルの拡張子が「.c」はC言語ソースプログラムです。「.x30」はデバッガが使用するオブジェクトファイルです。「.mot」は「.x30」ファイルをもとローラ2形式に変換したファイルで、フラッシュROMに書き込むファイルです。

出荷時には、タイマ割込み方式のROM化ファイル「Oklabo_4.mot」がフラッシュROMに書き込まれております。

(1)、デバッガを使用する為のフラッシュライターソフトの使い方

デバッガ「KD30」を用いてデモプログラムを実行するには、予めフラッシュROMにモニタファイル「Mon_main.mot」ファイルを書き込まなければなりません。次の手順で書き込んで下さい。

OAKS-LABOとパソコン間をRS232Cケーブル(ストレート線)で接続します。また、図27に示されるCPUボード「OAKS16」のJP1端子1-2間を何らかのショート端子を用いてショートして下さい。



JP1の1-2間をショート端子でショートします。

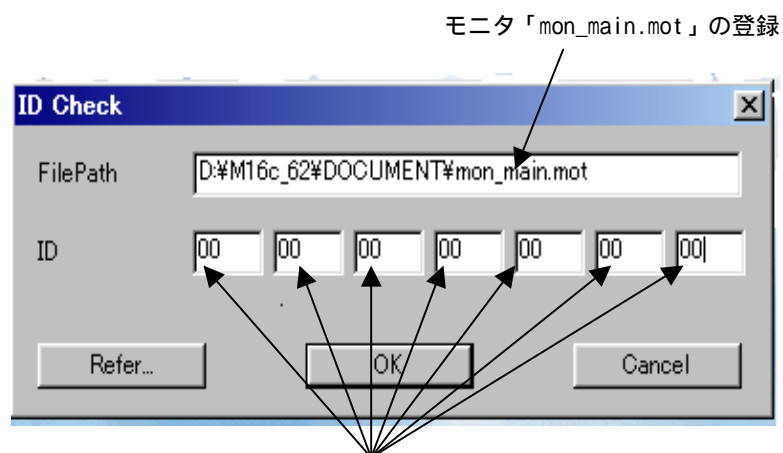
図27 CPUボード「OAKS16」のJP1端子

OAKS-LABOの電源をオンにし、フラッシュライタソフト「M16Flash」アイコンをクリックすると図28で示されるウインドウが表示します。

「Refer」ボタンをクリックし、モニタプログラム「mon_main.mot」を登録します。

空欄になっている七つのIDウインドウに「00」を書き込んで下さい。

「OK」ボタンをクリックします。

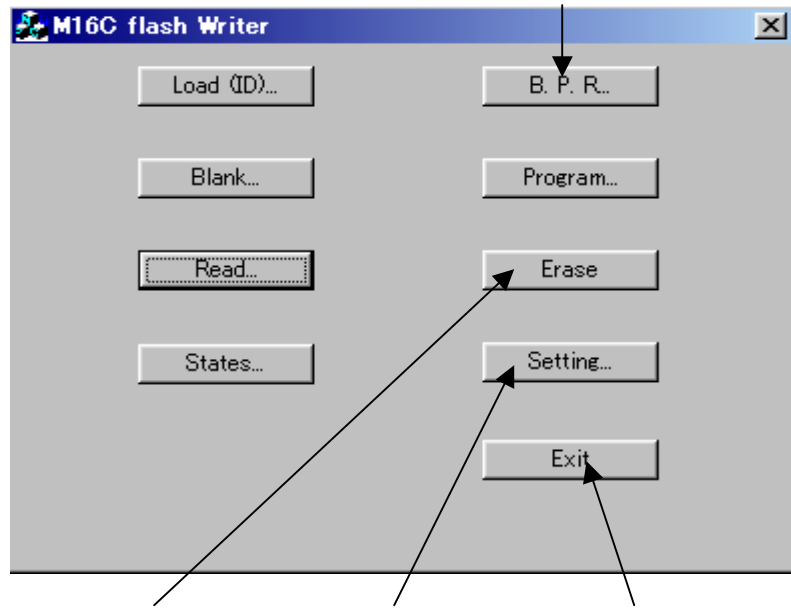


全てのIDウインドに「00」を入れる

図28 「ID Check」ウインドウ

図29で示される「M16C flash Write」ウィンドウが表示します。

: ブランクチェック、書込み、ベリファイ



: 消去チェック

: ボーレート設定

: 終了

図29 「M16C flash Write」ウィンドウ

「Setting」ボタンを押します。図30で示されるウィンドウ内の「Baud rate」設定プルダウンメニューからボーレートを図のように57600を選んで下さい。設定後、「OK」ボタンを押して下さい。

57600bps を選びます。

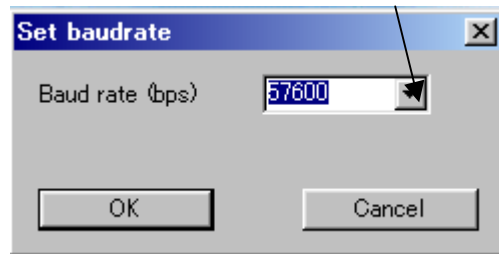
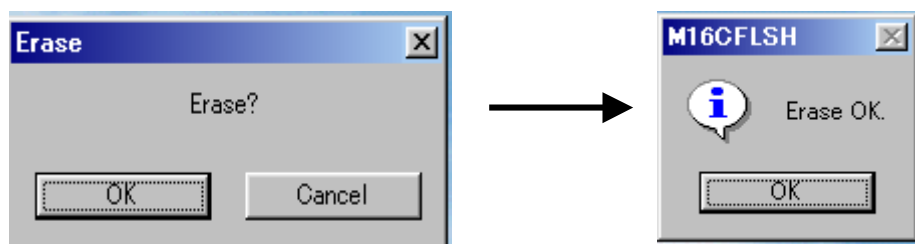


図30 「Set baudrate」ウィンドウ

「Erase」ボタンを押します。図31(a)で示されるウィンドウの「OK」ボタンを押すと、(b)図で示される「Erase OK」ウィンドウが表示します。



(a)

(b)

図31 「Erase」チェックウィンドウ

「B . P . R」ボタンを押します。このボタンは、ブランクチェック、プログラムの書込み、書込み確認を一連で実行します。図3 2 (a)で示されるプログラムのメモリ範囲が表示されます。確認後、「OK」ボタンを押します。図(b)のウインドウが開きブランクチェックを開始します。

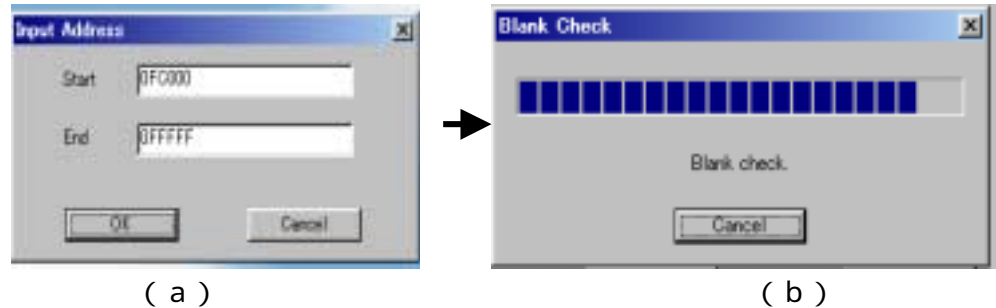


図3 2 「B l a n k C h e c k」ウインドウ

ブランクチェックが正常に終了すると、図3 3 (a)に示されるウインドウが表示されます。「OK」ボタンを押すと図(b)のウインドウが開き、書込みを開始します。

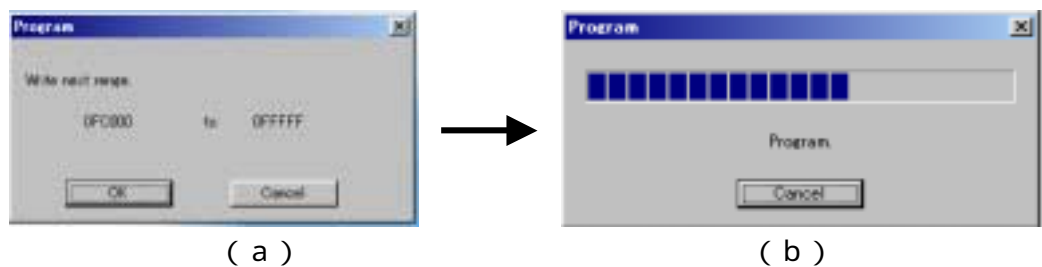


図3 3 「P r o g r a m」ウインドウ

プログラムが正常に書き込まれると、図3 4 (a)に示されるウインドウが表示され、書込みの確認を行います。正常に書き込まれていると図(b)のウインドウが開き、書込み確認が終了します。OK」ボタンを押して下さい。



図3 4 「R e a d c h e c k」ウインドウ

最後に、図29で示される「M16C flash Write」ウインドウから「Exit」ボタンを押して下さい。

(2)、デバッガの起動方法

デバッガ「KD30」を起動するには、OAKS-LABOのシリアルポートからチャンネル1とパソコン間をRS-232Cケーブル（スレートケーブル）で接続します。

デバッガ「KD30」アイコンをクリックすると図35に示される起動ウインドウが表示されるので、パソコン側のシリアル端子はCOM1に固定されています。

ロボットの電源をONにして下さい。また、RS-232Cケーブルの接続を確認後、「OK」ボタンを押します。もし、エラーウインドウが表示される場合には、ケーブルの接続や電源を再確認します。また、予めフラッシュROMにモニタファイル「Mon_main.mot」ファイルが書き込まれていなければなりません。

図35で示される起動ウインドウの詳しい使用方法は「KD30」に添付されているマニュアルをご覧ください。

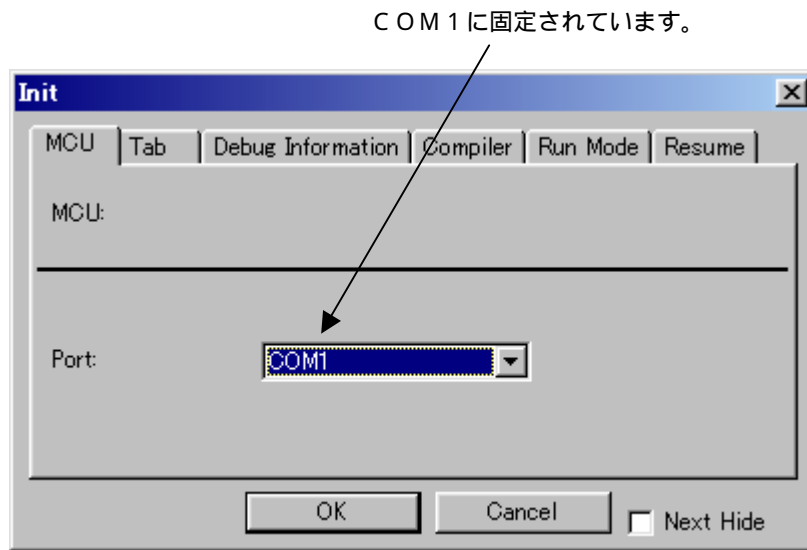


図35 デバッガ起動ウインドウ

「OK」ボタンを押すと、図36で示されるデバッグウインドウが表示されます。

ファイルのロード方法

ファイルをロードするには、メニューから左隣の「File」をクリックします。プルダウンメニューから「Download」を選びます。更にこの中の「Load Module」を選ぶとファイルロードウインドウが開きます。ロード可能なファイルは、拡張子に「.x30」です。デモ用のファイルから「Oklabo_1.x30」と「Oklabo_3.x30」がロード可能です。

プログラムの実行方法

プログラムを実行するには、通常プログラムカウンタにロードしたユーザプログラムの先頭番地を予め設定する必要があります。しかし、デモプログラムの場合には、ロードの段階で自動的にプログラムカウンタに先頭番地が設定されます。


プログラムの実行は、左隅の実行アイコン「」をクリックするとプログラムが“RUN”します。

図36で示されるデバッグウィンドウの詳しい使用方法はリモートデバッガ「KD30」に添付されているマニュアルをご覧ください。

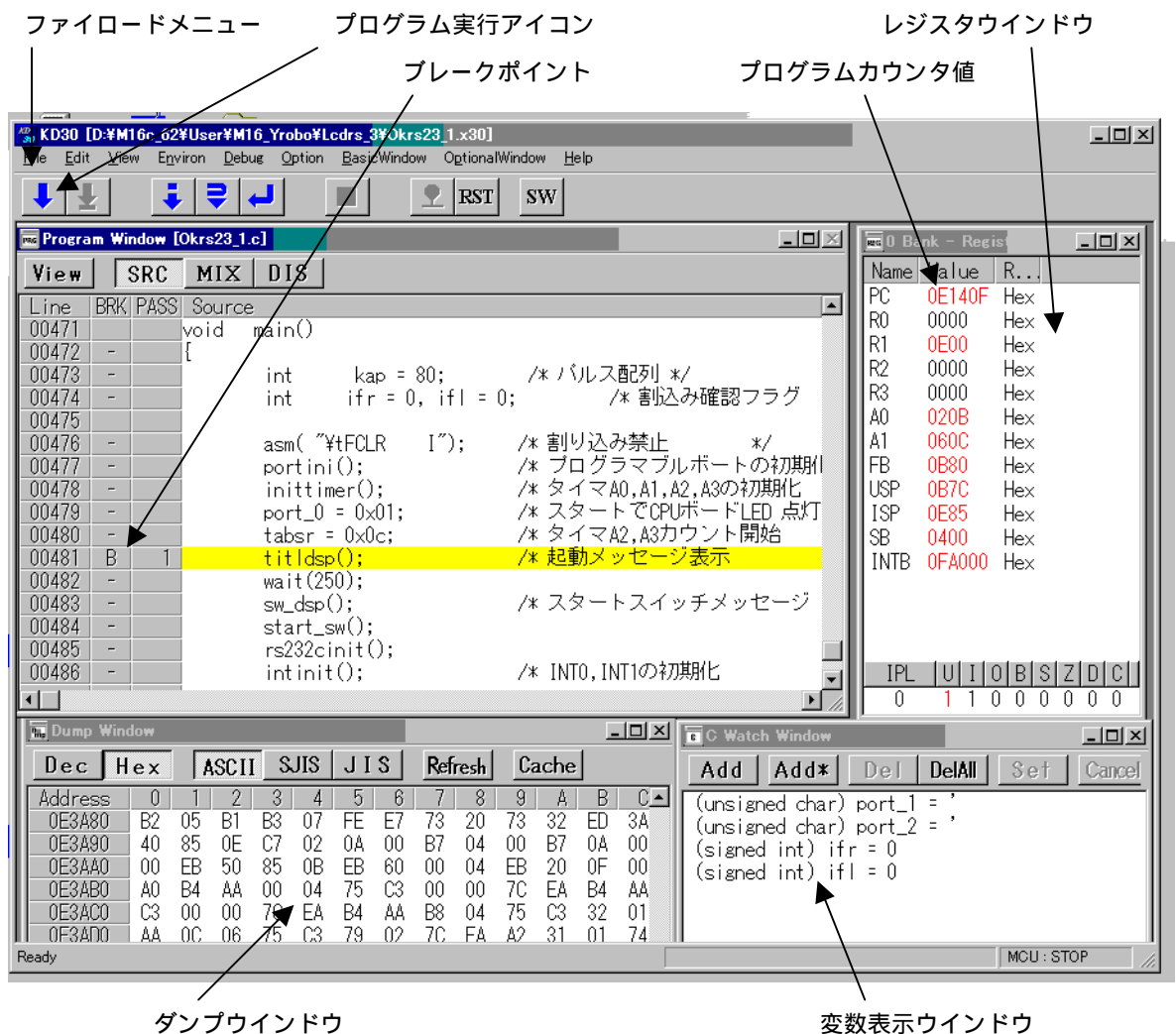


図34 デバッガKD30デバッグ画面

11. TM統合化開発環境での使い方

OAKS-LABO用RS-232C無線ユニットには、最新のTM統合開発環境「TMV3.11」が付属しています。このTM統合化開発環境を用いることにより、バッチファイルまたはメイクファイルによりDOS窓上でコンパイルを行っていた方式からWindows上でエディタの起動、コンパイラ、デバッガの起動等、統合操作を可能にしました。

(1) インストールと注意点

付属のCD-Rには、TM統合化開発環境「TMV3.11」とコンパイラ「NC30WA V4.00 Release 2 (エントリー版)」が入っています。既に、ライントレースロボット「OAKS-LABO」に添付されているコンパイラ「NC30WA V3.00」がインストールしている場合、インストール中の「コンポーネントの選択」画面で「インストール先ディレクトリ」項目を「参照(R)」を使用して別のインストール先を指定してインストールして下さい。もし、同じディレクトリにインストールする場合、必ず、既にインストールされているNC30WAをアンインストールした後で、本製品のNC30WAをインストールして下さい。

OAKS-LABOに添付されている全てのデモサンプルプログラムを旧バージョン「KNC30 V3.00」でコンパイルするとエラーが発生します。したがって、本製品に入っている全てのデモサンプルプログラムは、旧バージョンではコンパイルできません。

(2) エントリー版 (新バージョン) の仕様について

プログラムの記述について

旧バージョンのプログラムをエントリー版でコンパイルする際、表7で示される語の前にアンダースコア()を付加して下さい。

表7 新旧バージョンの記述相違一覧

| KNC30 旧バージョン版 | NC30 エントリー版 |
|---------------|---------------|
| inline | <u>inline</u> |
| near | <u>near</u> |
| far | <u>far</u> |
| asm() | <u>asm()</u> |

TM統合開発環境「TMV3.11」について

- ・ インスペクタ機能はご使用になれません。
- ・ ブラリプロジェクトは作成できません。

ソフトウェアおよびユーティリティについて

下記に示すソフトウェアおよびユーティリティはご使用になれません。

表 8 使用できないソフトウェアおよびユーティリティー一覧

| | |
|---------|---|
| ソフトウェア | STK ビューワ, MPA ビューワ, アセンブルオプティマイザ (aopt30), ライブラリアン (lb30), 構造化記述アセンブラ (pre30),] 標準関数ライブラリソースファイル |
| ユーティリティ | utl30(SBDATA 宣言&SPECIAL ページ宣言ユーティリティ) |

NC30とAS30のオプションについて

マニュアルに記載されている下記のオプションはご使用になれません。

表 9 使用できないCコンパイラオプション一覧

| Cコンパイラ (NC30) | |
|----------------|--|
| デバッグ用オプション | genter -gno_reg |
| 最適化オプション | -O[1-5], -OR, -OS, -Oconst(-OC), -Ono_bit(-ONB), -Ono_break_source_debug(-ONBSD), -Ono_float_const_fold(-ONFCF), -Osp_adjust(-OSA), -Ostack_frame_align(-OSFA), Oloop_unroll(-OLU), Ono_asmopt(-ONA), -Ono_logical_or_combine(-ONLOG), -Ocompare_byte_to_word(-OCBTW), -Ono_stdlib |
| 生成コード変更オプション | -finfo, -fuse_DIV(-fUD), -fans, -fnear_ROM(-fNROM), -fsmall_array(-fSA), -fno_align(-fNA) |
| アセンブル、リンクオプション | -as30, -ln30 |
| その他のオプション | -dsource(-dS), -dsource_in_list(-dSL) |

表 10 使用できないアセンブラオプション一覧

| アセンブラ (AS30) | |
|--------------|----------------|
| オプション | -finfo, -P, -M |

(3) T M 統合化開発環境の使い方

T M 統合化開発環境「T M」を使用する場合、ライントレースロボット「O A K S - L A B O」に添付されているコンパイラ「K N C 3 0」(旧バージョン)ではご使用になれません。「T M」を使用する場合には、本製品に入っているエントリー版コンパイラ「N C 3 0」(新バージョン)を使用します。「T M」を起動すると細長いプロジェクトバーが表示されます。マウスによって右端を縮めると図 3 5 で示されるフローティング状態のプロジェクトバーが表示されます。

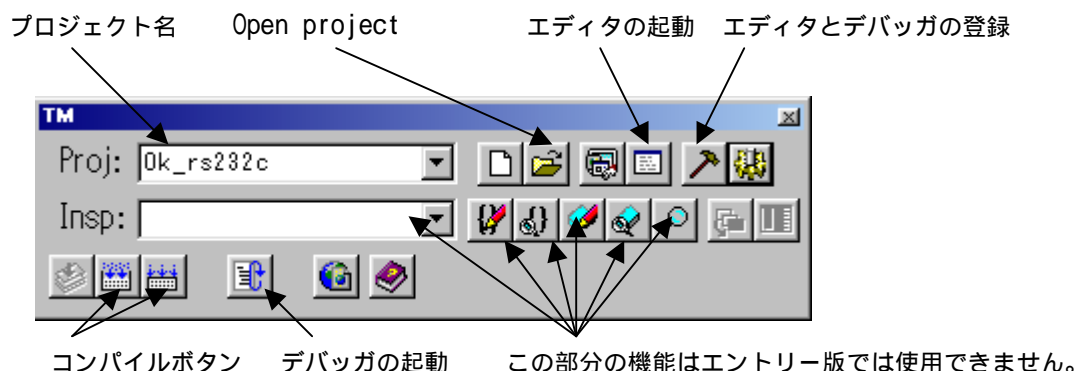





図 3 5 プロジェクトバー


エディタとデバッガの登録

最初に  (ツールの登録) ボタンをクリックし、使い慣れたエディタとリモートデバッガ「K D 3 0」を登録します。

ここで登録しないと、図 1 9 図の  (エディタの起動) ボタンと  (デバッガの起動) ボタンが使用できません。


プロジェクトファイルのロード

T M 統合化開発環境「T M」を用いて W i n d o w s 上でコンパイルを行うには、プロジェクトファイルを作成しなければなりません。新規のユーザプログラムからプロジェクトファイルを作成しコンパイルする方法は、付属のユーザズマニュアル (P D F ファイル) をご覧下さい。本製品のデモサンプルプログラムにはプロジェクトファイル「O k _ r s 2 3 2 c . t m k」が入っています。



プロジェクトファイルをロードするには図 1 9 で示される  (プロジェクトを開く) ボタンをクリックします。開くウインドウから付属の C D - R からコピーしたドライブとディレクトリを選び「O k _ r s 2 3 2 c . t m k」をロードします。図 1 9 で示されるプロジェクトウインドウにファイル名が表示されます。

コンパイル方法

プロジェクトで設定したソースプログラム「O k _ r s 2 3 2 c . c」をコンパイル

するには、 (ビルドまたはリビルド)ボタンをクリックします。ウインドウにコンパイルメッセージが表示されます。スクロールバーによって前後のメッセージを読むことが可能です。

エディタの起動

 (ツールの登録)ボタンによって前もって使い慣れたエディタを登録しておく、 (エディタの起動)ボタンでエディタが起動します。ソースプログラムをロードしてプログラムを修正できます。

デバッガの起動



 (ツールの登録)ボタンによってリモートデバッガを登録しておく、 (デバッガの起動)ボタンをクリックするとリモートデバッガ「KD30」が起動できます。

図35において、2段目のインスペクタとその関連機能は、エントリー版では使用できません。また、TM統合化開発環境「TM」の詳しい使用方法は、本製品のCD-Rに入っているユーザーズマニュアルをご覧ください。

12. プログラムの作成手順

OAKS-LABOには、プログラム開発が容易に行えるようにCコンパイラ、アセンブラ、リモートデバッガ、フラッシュライターソフトが付属しています。C言語プログラムは、他社のCPUポートで開発したソースプログラムを、OAKS-LABOで使用するM16C/62シリーズ用プログラムに容易に移植することが可能です。また、アセンブラも熟知する必要がありませんので、C言語でプログラムする事をお勧めします。また、商品に添付しているデモプログラムは全てC言語プログラムとなっています。

(1)、プログラム開発の手順

プログラムを開発するには、図36に示されるように次の手順でプログラムを開発します。

プログラム図法を用いてフローチャート等のアルゴリズムを作成します。

使い慣れたエディタを用いてソースプログラムを記述します。

コーディングが終了したら、コンパイラ「KNC30WA」を用いてコンパイルを行います。

エラーメッセージが表示された場合には、 項に戻り、エラーを修正します。

コンパイルが終了したら、リモートデバッガ「KD30」を用いて、ロボットがソフトウェア設計通りに動作するか確かめます。

正常に動作しない場合には、原因を究明し、 項に戻りプログラムを修正します。

正常にロボットが動作したら、フラッシュライターソフト「M16CF1sh」を用いて

フラッシュROMにプログラムを書き込みます。
プログラム開発を終了します。

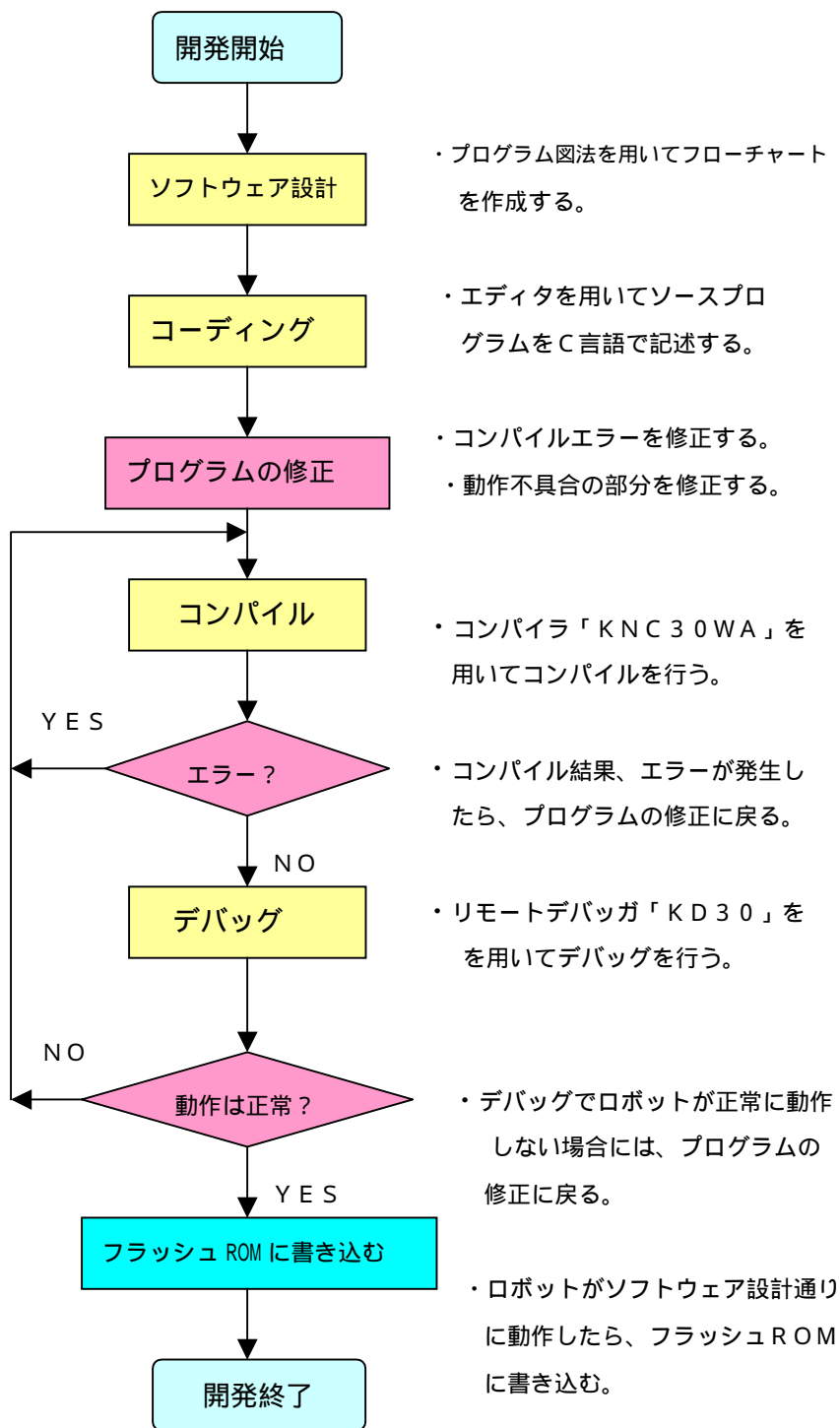


図36 プログラム開発の手順

(2)、コンパイル方法

M16C / 60シリーズ専用のコンパイラ「NC30WA」にはCコンパイラとアセンブラが標準で付属しております。図37はコンパイル手順を示したフローチャートです。作成したC言語プログラム「Oklabo_4.c」はコンパイルドライバ「nc30」によってアセンブラプログラム「Oklabo_4.r30」に変換します。

Cソースプログラムにエラーがなければ、スタックポインタ等を記述したスタートアップルーチン「ncrt0.a30」と割り込みを使用する場合には、インクルードファイル「sect30.inc」の割り込みベクタテーブルを書き換えてアセンブルします。全てのファイルにエラーがなければリンカでリンクします。リンク段階でエラーがなければデバッグ可能な実行ファイルが完成します。次に、リモートデバッガ「KD30」を用いてプログラムが正常に動作する事を確認後、ファイル形式変換ソフト「Lmc30」によってモトローラ2形式のファイル「Oklabo_4.mot」に変換し、フラッシュライタソフト「M16CF1sh」で書込み、プログラム作成は終了です。

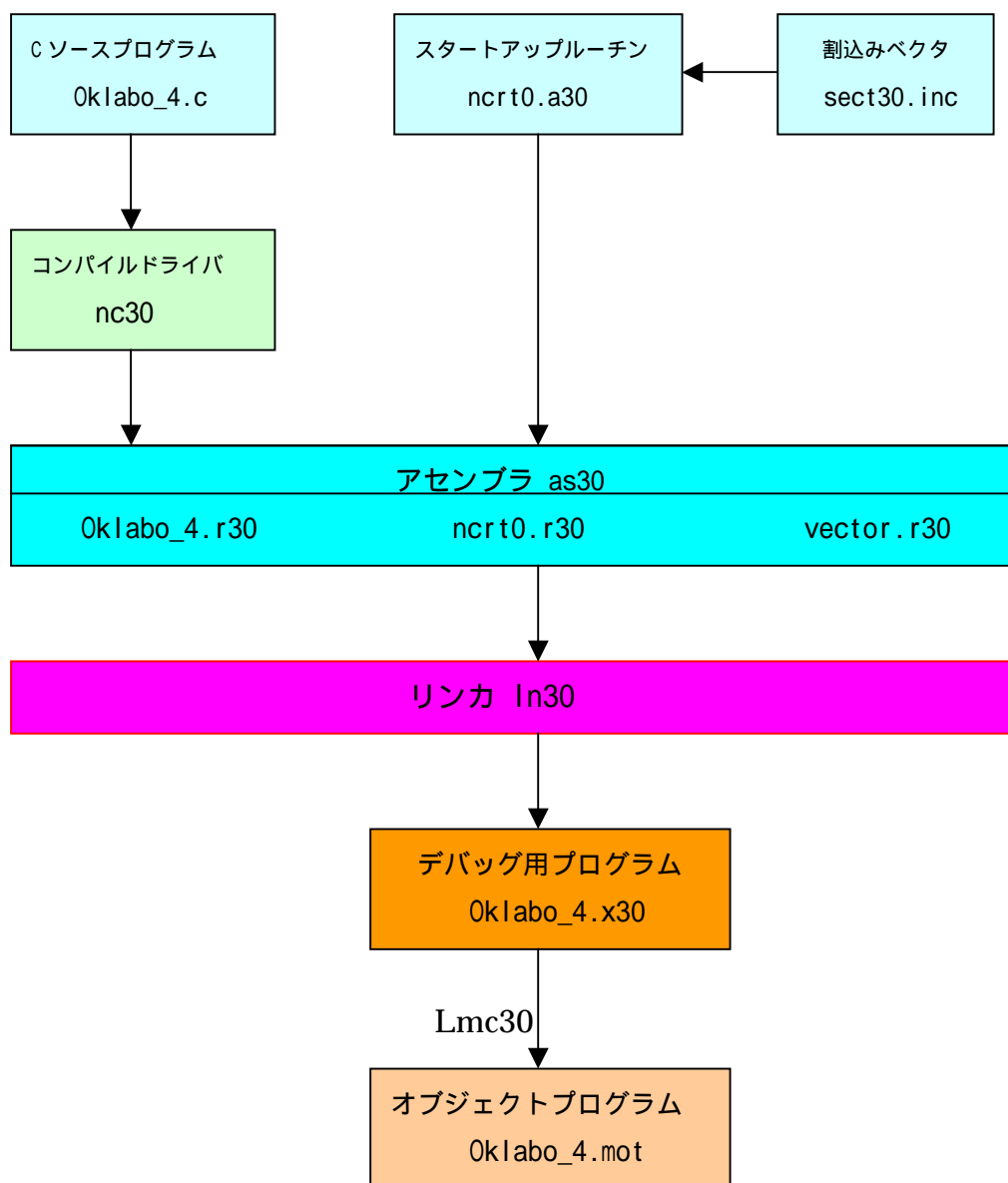


図37 コンパイラ「NC30WA」のコンパイル手順

(3)、コンパイル用バッチファイル例

コンパイラ「NC30WA」は通常、TM統合化開発環境の元でコンパイルすると便利ですが、バッチファイルによってDOS窓上でコンパイルすることができます。

図38は、バッチファイルに予めコンパイルの手順を記述した記述例です。

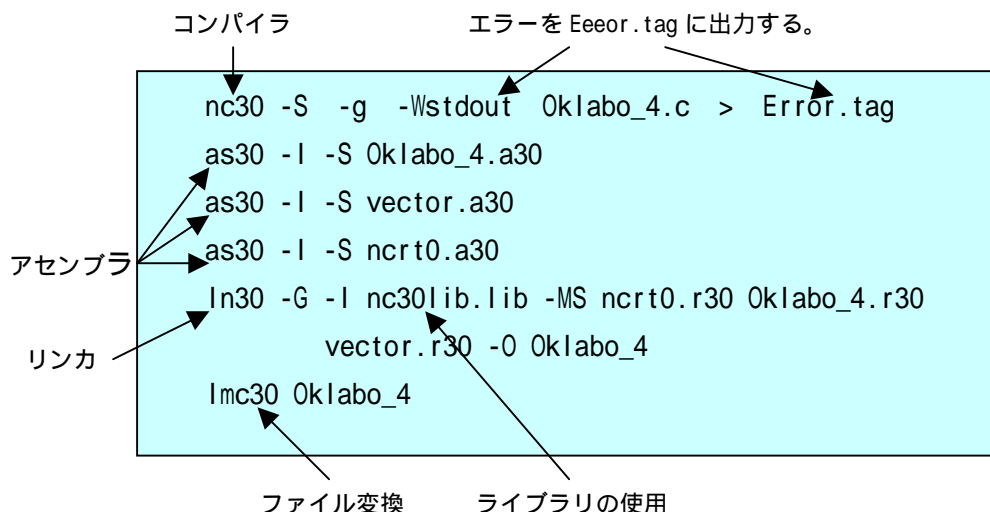


図38 コンパイル用バッチファイル記述例

図38で示されるバッチファイルは、出荷時にフラッシュROMに書き込まれているデモソフト「Oklabo_4.c」をコンパイルする為のバッチファイルです。コンパイラのオプションに記述されている「Wstdout」は、エラーリストを出力する指定です。行の最後に「> Error.tag」のように記述すると、エラーリストをError.tagファイルに書き込まれます。このエラーファイルを見ながらソースプログラムを修正すると便利です。

図37に使用されているオプションの詳しい使い方は、コンパイラ「NC30WA」マニュアルをご覧ください。

(4) I/Oポートの記述方法

M16C/62シリーズのCPUは、メモリマップI/O方式で全てのI/Oアドレスを00000番地から00400番地に割り付けます。C言語でI/Oポートを定義するには、図39のように拡張機能による絶対アドレス指定を用いると便利です。

“#program ADDRESS”は、関数外で定義された変数および関数内でスタティック宣言された変数のみ有効となります。ポート名(変数名)には、適当な名前を付けて、次のように記述します。

```
# program ADDRESS 変数名 絶対アドレス
```

上記の宣言により、変数名を絶対アドレスに配置します。このような拡張機能による絶対アドレス指定を用いると、メモリ使用量を節約できます。

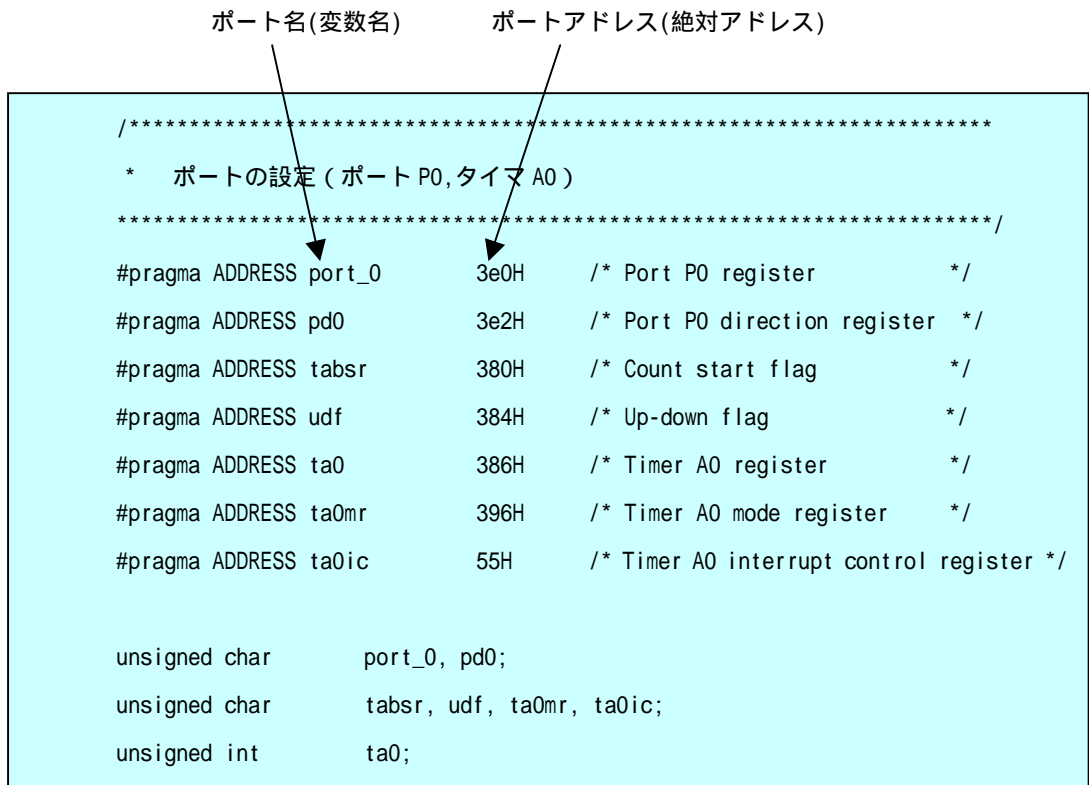


図 3 9 I / Oポートの記述例

(5)、割り込み処理

M16C / 62 シリーズの CPU で割り込みを C 言語関数として記述する手順は次の 2 つです。

割り込み処理関数の定義。

割り込みベクタテーブルの登録。

割り込み処理関数の定義は、割り込み関数名に適当な名前を付けて、次のように記述します。

```
# program INTERRUPT 割り込み関数名
```

上記のように記述すると、指定した関数の入り口と出口において、通常の間数の手続き以外に、全レジスタの退避、復帰と“`reit`”命令を生成します。割り込み処理関数の型は、引数 / 戻り値共に `void` 型のみ有効です。図 4 0 は、タイマ A 0 を割り込みに使用する為の割り込み処理関数の記述例です。

割り込み関数名

```

/*****
*   割り込み関数定義(タイマ A0)
*****/
void far   ta0int();   /* 割り込み関数 */
#pragma INTERRUPT ta0int

```

図 4 0 割り込みの記述例

(6)、割り込みベクタテーブルへの登録

割り込みを正常に使用する為には、割り込み処理関数を定義すると共に割り込みベクタテーブルに登録する必要があります。

割り込みベクタテーブルの変更は、次の手順で行います。

割り込み処理関数名を疑似命令“`.glob`”で外部定義します。

使用する割り込みのダミー関数“`dummy__int`”を、割り込み処理関数名に変更します。

図 4 1 は、スタートアップルーチン「`ncrt0.A30`」に記述されている見出し文「`Interrupt section start`」内に割り込み処理関数を記述したものです。外部定義では次のように必ずアンダーバーを付けて記述して下さい。

```
.glob    _ta0int
```

また、タイマ A 0 を割り込みで使用するには、図 4 2 に示されるインクルードファイル「`sect30.inc`」のベクタテーブル見出し名「`variable vector section`」21番のダミー関数“`dummy__int`”を次のように必ずアンダーバーを付けて記述して下さい。

```
.lword   _ta0int
```

割り込み処理関数名の記述

```

;=====
; Interrupt section start
;-----
.insf      start,S,0
.glob     _ta0int
.glob     start
.section  interrupt

start:

```

図 4 1 スタートアップルーチンに割り込み処理関数名を記述する例

割り込み処理関数名の記述

```

.else
.lword  dummy_int      ; vector 0 (BRK)
.org    (VECTOR_ADR +44)
.lword  dummy_int      ; DMA0 (for user)
.lword  dummy_int      ; DMA1 2 (for user)
.lword  dummy_int      ; input key (for user)
.lword  dummy_int      ; AD Convert (for user)
.org    (VECTOR_ADR +68)
.lword  dummy_int      ; uart0 trance (for user)
.lword  dummy_int      ; uart0 receive (for user)
.lword  dummy_int      ; uart1 trance (for user)
.lword  dummy_int      ; uart1 receive (for user)
.lword  _ta0int      ; TIMER A0 (for user) (vector 21)
.lword  dummy_int      ; TIMER A1 (for user) (vector 22)
.lword  dummy_int      ; TIMER A2 タイマ A2(vector 23)
.lword  dummy_int      ; TIMER A3 タイマ A3(vector 24)
.lword  dummy_int      ; TIMER A4 (for user) (vector 25)
.lword  dummy_int      ; TIMER B0 (for user) (vector 26)
.lword  dummy_int      ; TIMER B1 (for user) (vector 27)
.lword  dummy_int      ; TIMER B2 (for user) (vector 28)
.lword  dummy_int      ; INT0 (for user) (vector 29)
.lword  dummy_int      ; INT1 (for user) (vector 30)
.lword  dummy_int      ; INT2 (for user) (vector 31)
.endif

```

図 4 2 ベクターテーブルの記述例

13. Flash ROM への書込み

デバッガによってトレースロボットが正常に動作することが確かめられ後、最後に Flash ROM に書き込んで完成です。デバッグで使用したデバッグ用プログラムをコンパイラに添付されているファイル形式変換プログラム「lmc30」を用いてモトローラ S2 形式の ROM 化ファイル(Oklabo_4.mot)に変換します。デバック時の状態で RS232C ケーブルはそのまま接続しておきます。また、CPU ポート上の JP1 をショートすると Flash ROM への書込みが可能となります。書込みにはフラッシュライターソフト「M16CF1sh」を起動します。使用方法は、第10章の(1)項「デバッガを使用する為のフラッシュライターソフトの使い方」をご欄下さい。

表7は、図29で示される「M16C flash Write」ウインドウのメニューボタンの機能を表したものです。

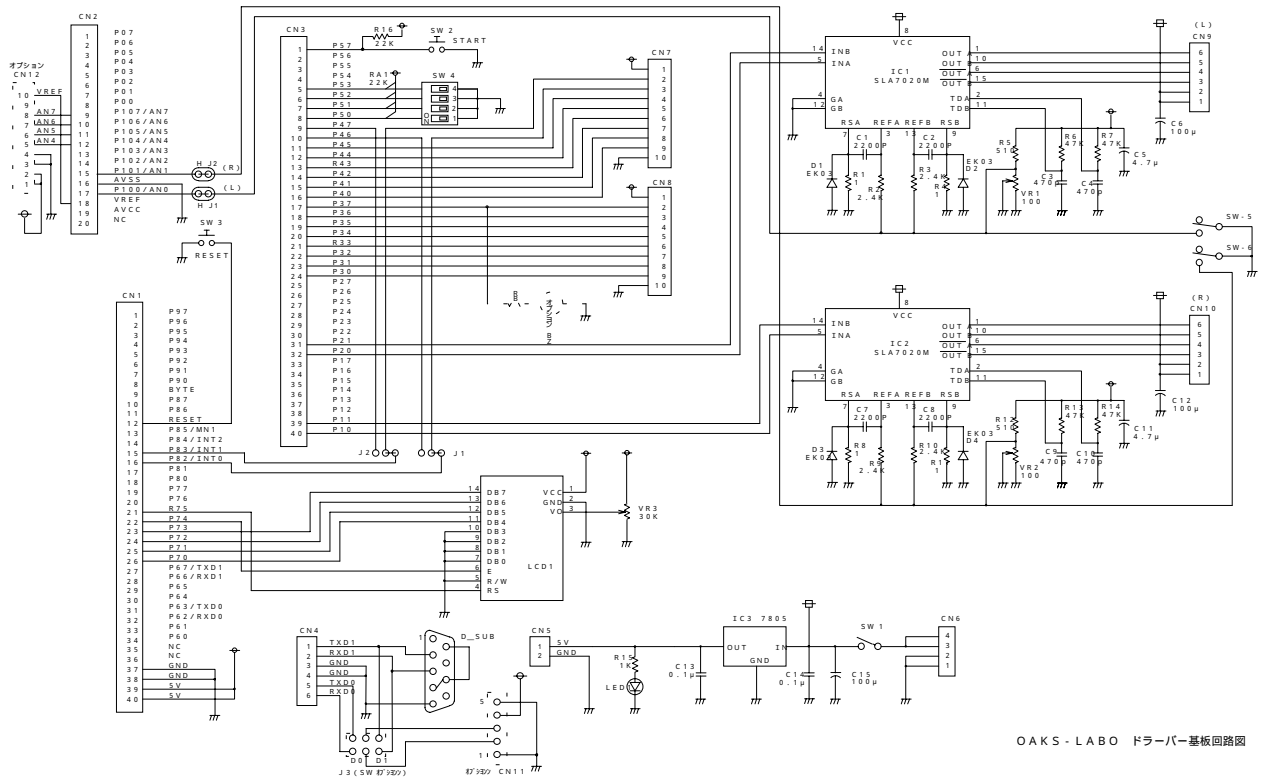
表 1 1 各メニューボタンの機能

| メニュー | 機 能 |
|----------|------------------------------|
| Load(ID) | ファイルの選択と ID チェックを行います。 |
| Blank | 指定した領域に対してブランクチェックを行います。 |
| Read | 指定したファイルの内容とを比較します。 |
| Status | マイコンのステータスを表示します。 |
| B.P.R | ブランクチェック、プログラム、ベリファイを順次行います。 |
| Program | 指定したファイルをフラッシュ ROM に書き込みます。 |
| Erase | フラッシュ ROM の全領域をイレースします。 |
| Setting | 通信設定を行います。 |
| Exit | 書込みを終了します。 |

付 録

付録 1 ドライバーボード回路図

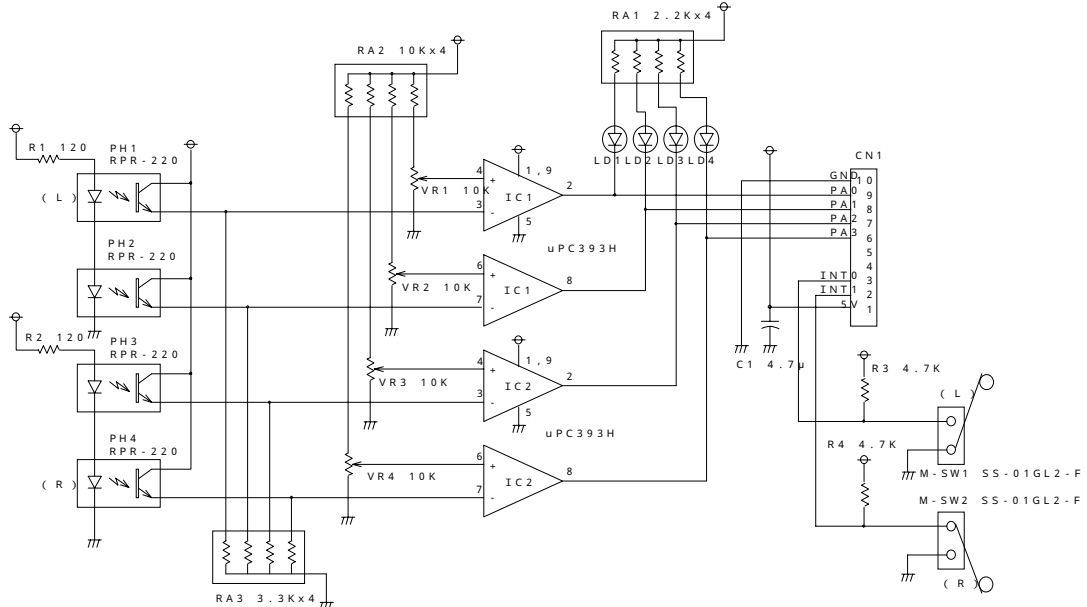
拡大してご覧下さい。



OAKS - LABO ドライバー基板回路図

付録 2 センサボード回路図

拡大してご覧下さい。

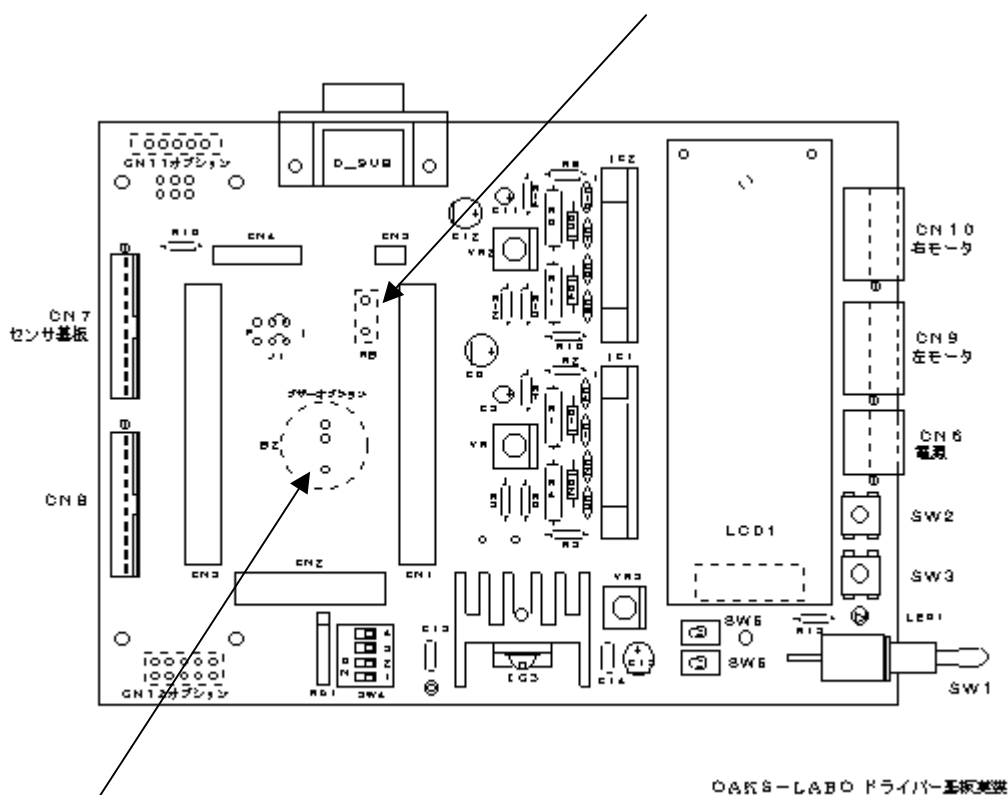


OAKS - LABO センサー基板回路図

付録 3 ドライバーボード構成図

拡大してご覧下さい。

抵抗 500[]程度を接続します。



オプション：圧電ブザーの取り付け

上図の位置に圧電ブザーと抵抗を取り付けます。圧電ブザー端子の一方に2つの穴が空いています。これは、7.5mmタイプと10mmタイプの何れでも取り付けを可能にしています。デモプログラムでは、左右の接触センサに障害物が接触すると2回鳴ります。また、停止ラインで停止すると3回鳴るようにプログラムされています。

推奨圧電ブザー

10mmタイプ

- ・ 村田製作所 PKM17EPP-4001
- ・ 村田製作所 PKM22EPP-4001

7.5mmタイプ

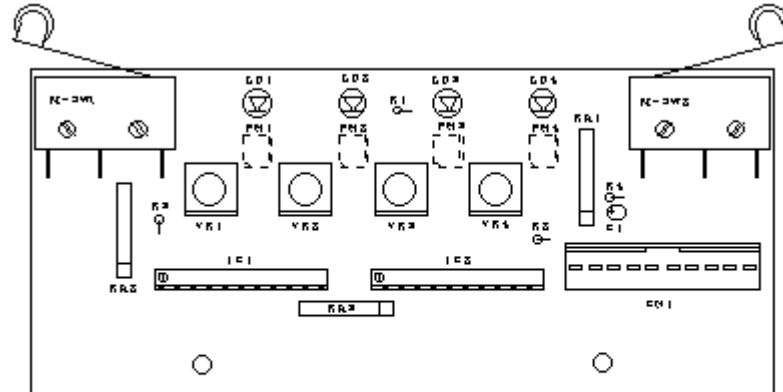
村田製作所 PKM13EPP-4002

抵抗

500[]程度の抵抗を上図の図面位置に取り付けてください。

付録 4 センサボード構成図

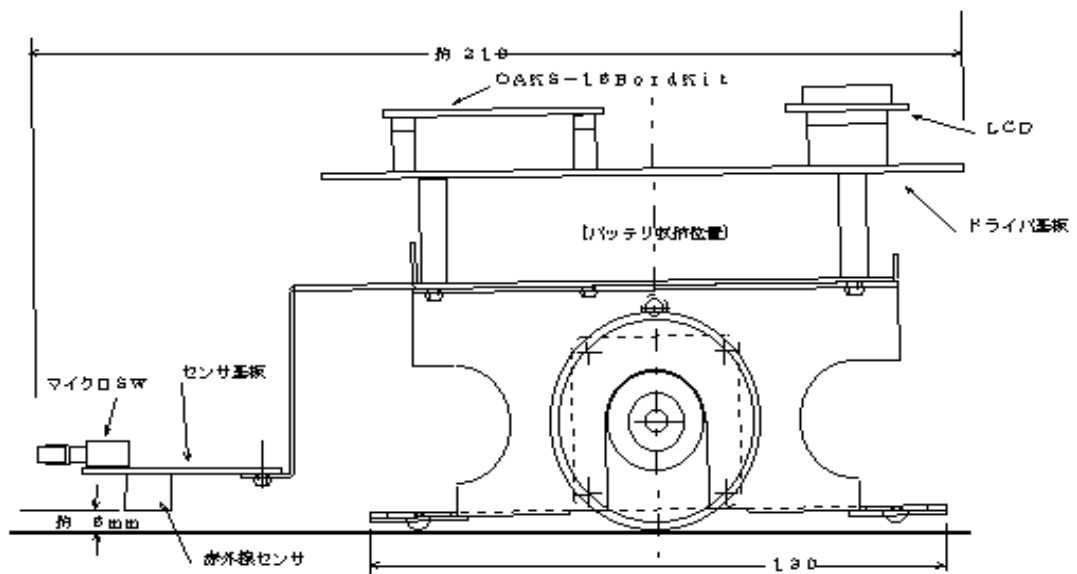
拡大してご覧下さい。



○OAKS-LABO センサ基板組

付録 5 OAKS - LABO外形図

拡大してご覧下さい。



○OAKS-LABO外形図

付録 6 コネクタ一覧

CN 1 (OAKS16 CPU ボード)

| | | | |
|----|--------------|----|--------------|
| 1 | | 2 | |
| 3 | | 4 | |
| 5 | | 6 | |
| 7 | | 8 | |
| 9 | | 10 | |
| 11 | | 12 | RESET |
| 13 | | 14 | |
| 15 | 右接触センサ | 16 | 左接触センサ |
| 17 | | 18 | |
| 19 | | 20 | |
| 21 | P75 RS(LCD) | 22 | P74 E (LCD) |
| 23 | P73 DB7(LCD) | 24 | P72 DB6(LCD) |
| 25 | P71 DB5(LCD) | 26 | P70 DB4(LCD) |
| 27 | TXD1 | 28 | RXD1 |
| 29 | | 30 | |
| 31 | TXD0 | 32 | RXD0 |
| 33 | | 34 | |
| 35 | | 36 | |
| 37 | GND | 38 | GND |
| 39 | +5V | 40 | +5V |

CN 3 (OAKS16 CPU ボード)

| | | | |
|----|------------|----|------------|
| 1 | P57 START | 2 | |
| 3 | | 4 | |
| 5 | P53 4(SW4) | 6 | P52 3(SW4) |
| 7 | P51 2(SW4) | 8 | P50 1(SW4) |
| 9 | P47 JP2 | 10 | P46 JP1 |
| 11 | P45 4(CN7) | 12 | P44 5(CN7) |
| 13 | P43 6(CN7) | 14 | P42 7(CN7) |
| 15 | P41 8(CN7) | 16 | P40 9(CN7) |
| 17 | P37 ブザー | 18 | P36 3(CN8) |
| 19 | P35 4(CN8) | 20 | P34 5(CN8) |
| 21 | P33 6(CN8) | 22 | P32 7(CN8) |
| 23 | P31 8(CN8) | 24 | P30 9(CN8) |
| 25 | | 26 | |
| 27 | | 28 | |
| 29 | | 30 | |
| 31 | P21 INB(左) | 32 | P20 INA(左) |
| 33 | | 34 | |
| 35 | | 36 | |
| 37 | | 38 | |
| 39 | P11 INB(右) | 40 | P10 INA(右) |

CN 2 (OAKS16 CPU ボード)

| | | | |
|----|------------|----|------|
| 1 | | 2 | |
| 3 | | 4 | |
| 5 | | 6 | |
| 7 | | 8 | |
| 9 | | 10 | |
| 11 | | 12 | |
| 13 | | 14 | |
| 15 | AN1(右励磁電流) | 16 | AVSS |
| 17 | AN0(左励磁電流) | 18 | |
| 19 | | 20 | |

CN 6 (電源コネクタ)

| | |
|---|--------|
| 1 | GND |
| 2 | GND |
| 3 | +12[V] |
| 4 | +12[V] |

CN 7 (センサコネクタ)

| | |
|----|---------------|
| 1 | VCC +5[V] |
| 2 | INT1 右マイクロ SW |
| 3 | INT0 左マイクロ SW |
| 4 | P45 未使用 |
| 5 | P44 未使用 |
| 6 | P43 右外光センサ |
| 7 | P42 右内光センサ |
| 8 | P41 左内光センサ |
| 9 | P40 左外光センサ |
| 10 | GND |

CN8(未使用コネクタ)

| | |
|----|-------------|
| 1 | VCC +5[V] |
| 2 | P37 ブザー(OP) |
| 3 | P36 未使用 |
| 4 | P35 未使用 |
| 5 | P34 未使用 |
| 6 | P33 未使用 |
| 7 | P32 未使用 |
| 8 | P31 未使用 |
| 9 | P30 未使用 |
| 10 | GND |

CN9(右モータコネクタ)

| | |
|---|---------------------------|
| 1 | 12[V]電源 |
| 2 | 12[V]電源 |
| 3 | $\overline{\text{OUT B}}$ |
| 4 | OUT B |
| 5 | $\overline{\text{OUT A}}$ |
| 6 | OUT A |

CN10(左モータコネクタ)

| | |
|---|---------------------------|
| 1 | 12[V]電源 |
| 2 | 12[V]電源 |
| 3 | $\overline{\text{OUT B}}$ |
| 4 | OUT B |
| 5 | $\overline{\text{OUT A}}$ |
| 6 | OUT A |

CN11(オプションコネクタ)

| | |
|---|-----------|
| 1 | GND |
| 2 | RXD |
| 3 | TXD |
| 4 | VCC +5[V] |
| 5 | GND |

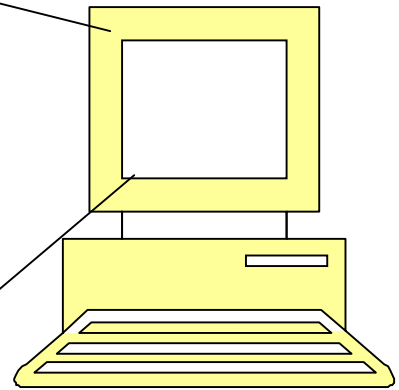
CN12(オプションコネクタ)

| | |
|----|-----------|
| 1 | VCC +5[V] |
| 2 | VCC +5[V] |
| 3 | GND |
| 4 | GND |
| 5 | AN4 A/D-4 |
| 6 | AN5 A/D-5 |
| 7 | AN6 A/D-6 |
| 8 | AN7 A/D-7 |
| 9 | NC |
| 10 | V. REF |

C N 7を他のセンサで使用するとき、割り込み端子(INT0, INT1)を P37, P47 に変更することが可能です。この場合、J1, J2 ジャンパの I 側をカットし、P 側に接続して下さい。
C N 8 は未使用コネクタですが、P37 はオプションで圧電ブザーが使用できます。

付録 7 オプション(単方向 & 双方向無線ユニットボードの活用)

OAKS LABO制御ウインドウ例



パソコン側無線ユニット

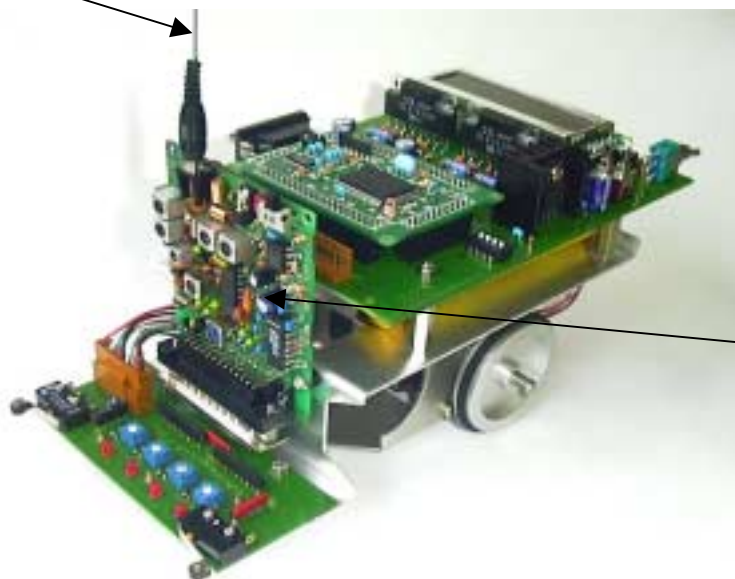
RS 232C
ケーブル



25P D-SUB コネクタ

無線通信

無線ユニット装着オプション



OAKS-LABO 側
無線ユニット

1. 単方向無線ユニットの活用方法

単方向無線ユニットの送信ボードをパソコンのRS 232Cと接続します。また、受信ボードをOAKS-LABOの専用取り付け位置に取り付けます。

パソコンから次のような無線制御が可能です。

速度指令

低速、中速度(低)、中速度(高)、高速

スタート指令

停止指令

180度ターン指令

90度右ターン指令

90度左ターン指令

2. 双方向無線ユニットの活用方法

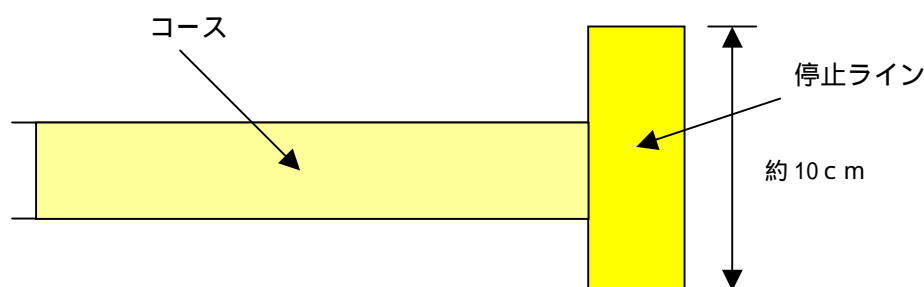
双方向無線ユニットは、お互いに周波数の異なる送信/受信ポートが2セットあります。一方のボードをパソコンのRS 232Cと接続します。また、他方のボードをOAKS-LABOの専用取り付け位置に取り付けます。

双方向無線ユニットを使用すると、1項の使い方の他に、リモートデバッガ「KD30」を使用したデバッグをコードレスで行えます。ライントレースロボット等の移動体は、RS 232Cケーブルを取り付けたまま走行させる事は困難です。このケーブルを無線に置き換える事により、デバッグがスムーズに行えます。

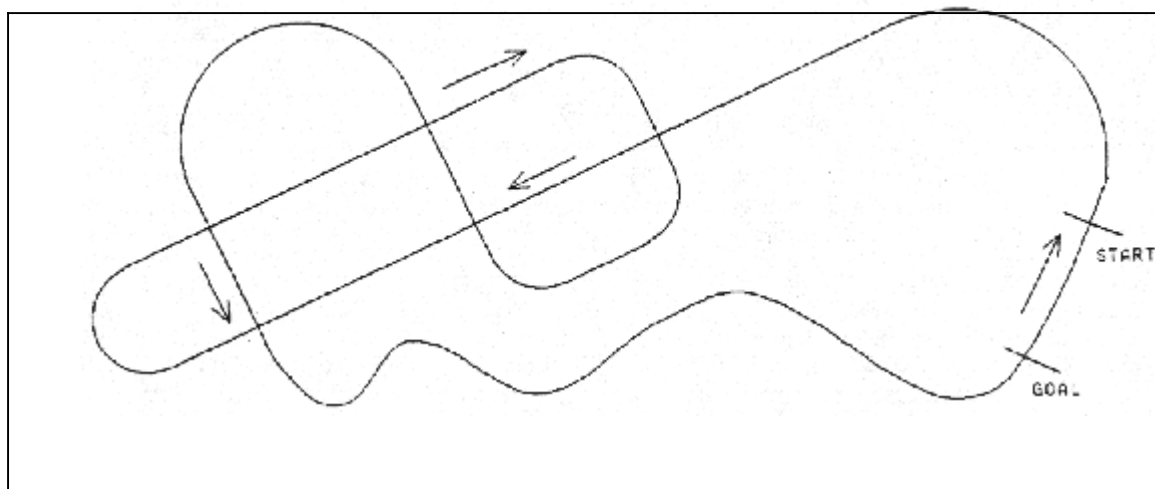
付録 8 停止ラインの作り方

OAKS-LABOは、スタート・ストップ及びゴールマーカの検出機能はありません。OAKS-LABOでの停止ラインは、4個のラインセンサが全て検出した場合とします。

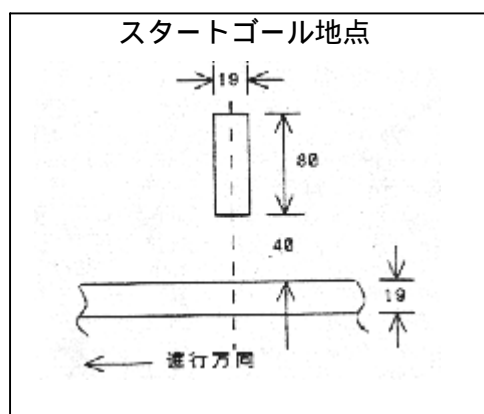
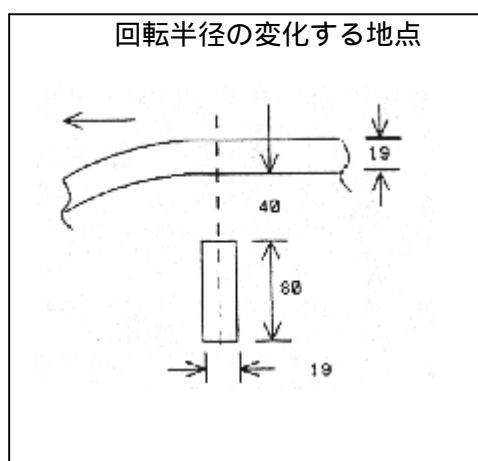
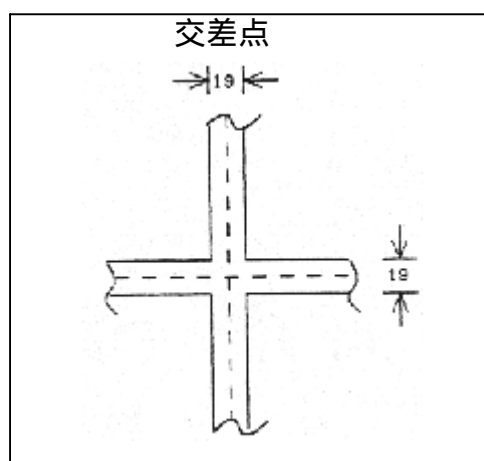
下図で示されるように、コースの終わりにコースと直角に10cm程度のコースと同じ色のテープを貼っておきます。



付録 9 ライントレースロボットのコース例



コースは黒地に白ラインまたは白地に黒ライン何れでも走行可能です。
Rの円弧曲率半径はR 30 cmからR 60 cmです。



OAKS-LABOにはスタート、ストップ及びコーナー・マーカの検出はできません。
OAKS-LABOのストップ検出は、4個のセンサが全て"1"または"0"で停止します。
デモプログラムでは、モードスイッチ3番を"ON"にします。

ご注意

- (1) 本書の内容の一部または全部を無断転載することは禁止されています。
- (2) 本書の内容は将来予告なしに変更することがあります。
- (3) 本書は、内容について万全を期して作成致しましたが、万が一ご不振な点や誤り、記載漏れなどお気づきなことがありましたら、お買い上げの販売代理店にご連絡ください。
- (4) 運用した結果の影響につきましては、(3)項にかかわらず責任を負いかねますのでご了承ください。

参考文献

- (1) 「マイコン制御ロボットの製作」横山直隆著 シータスク発行
- (2) 「Y-ROBO取扱説明書」 サン・マイテック社
- (3) 「コンパイラ ■NC30WA Ver1.0」 三菱電機

商標

Windows 95 / 98 / me / NTは、米国マイクロソフト社の登録商標です。

「OAKS - LABOユーザーズマニュアル」

2001年10月23日 第二冊発行

編集・発行

オークス電子株式会社

〒101-0025 東京都千代田区神田佐久間町3 21 3

第一千代田ビル3F

TEL (03)3863-1121 FAX (03)3863-1130

<http://WWW.oaks-ele.com>

