

ライントレースロボット

OAKS - LABO

デモプログラム(Oklabo_1 & Oklabo_2)の使い方

Ver2.0

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. デモプログラムの構成	2
2. デモプログラムの走行	3
(1) モード選択スイッチの割り当てと機能	3
(2) 走行テスト手順	4
3. デモプログラムの概要	6
(1) ソフトウェアタイマ方式	6
(2) ステッピングモータの駆動	8
(3) センサ入力プログラム	9
4. T M統合化開発環境での使い方	12
(1) インストールと注意点	12
(2) エントリー版（新バージョン）の仕様について	12
(3) T M統合化開発環境の使い方	14
5. コンパイル方法	15
6. コンパイル用バッチファイル例	16
7. I / Oポートの記述方法	17
8. 便利な関数の活用	18

1. デモプログラムの構成

ディレクトリ「Demo1」に入っている「Oklabo_1」ファイルは、デバグ「KD30」で使用するためのデバグ用ファイルです。また、ディレクトリ「Demo1_ROM」に入っている「Oklabo_2」ファイルは、フラッシュROMに書き込むためのROM化用ファイル「Oklabo_2.mot」を作成します。商品には表1で示されるファイルが入っています。

表1 「Demo1」ディレクトリのファイル一覧

ディレクトリ名	ファイル名	内 容
Demo1	Oklabo_1.c Oklabo_1.a30 Oklabo_1.r30 Oklabo_1.map Oklabo_1.lst Oklabo_1.x30 Oklabo_1.tmk Oklabo_1.mak M16io_1.h Lcdtime_1.h ncrt0.a30 sect30.inc lcc.bat	デバグ用ソフトタイマ方式ソースプログラム。 コンパイル出力アセンブラファイル。 アセンブラ出力ファイル。 マップファイル。 リストファイル。 デバグ用出力ファイル。 プロジェクトファイル コンパイル用メイクファイル I/O 記述ヘッダファイル。 LCD 表示とタイマ用ヘッダファイル。 スタートアップルーチンファイル。 割り込みベクターテーブル用インクルードファイル。 コンパイル用バッチファイル。
Demo1_rom	Oklabo_2.c Oklabo_2.a30 Oklabo_2.r30 Oklabo_2.map Oklabo_2.lst Oklabo_2.x30 Oklabo_2.mot Oklabo_2.tmk Oklabo_2.mak M16io_1.h Lcdtime_1.h ncrt0.a30 sect30.inc lcc.bat	ROM 化用ソフトタイマ方式ソースプログラム。 コンパイル出力アセンブラファイル。 アセンブラ出力ファイル。 マップファイル。 リストファイル。 ***** ROM 化用オブジェクトファイル。 プロジェクトファイル コンパイル用メイクファイル I/O 記述ヘッダファイル。 LCD 表示とタイマ用ヘッダファイル。 スタートアップルーチンファイル。 割り込みベクターテーブル用インクルードファイル コンパイル用バッチファイル。

デモプログラムの「Ok labo__1」と「Ok labo__2」は、ライン上を走行する機能だけを備えたソフトタイマ方式による最も簡単なプログラムです。速度制御と接触センサ及び圧電ブザーによる動作はプログラムされておられません。これらの学習は、デモプログラムの「Ok labo__3」と「Ok labo__4」をご覧ください。

2. デモプログラムの走行

デモプログラム「Ok labo__1」または「Ok labo__2」を使用してOAKS-LABOを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ユーザーズマニュアル付録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、DC14V～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2のように4つのスイッチの状態によって、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
未使用

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。

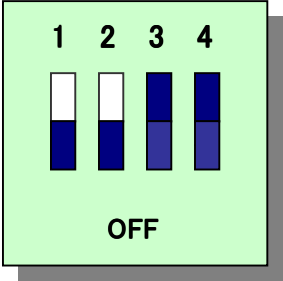
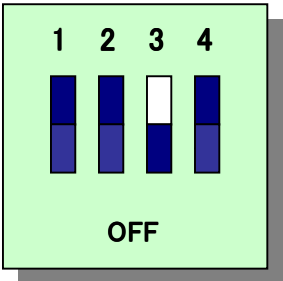
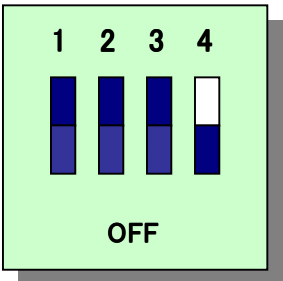
スイッチを“OFF”で使用する場合には、交差点などオールセンスのあるコースを走行させることができます。設定変更は、電源を投入する前またはLCD表示器に表示されるスタートまでの10秒以内に行います。

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2 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番は未使用です。</p>
	<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。 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。 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。</p> <p>OFF・・・黒地に白ラインの場合。</p>

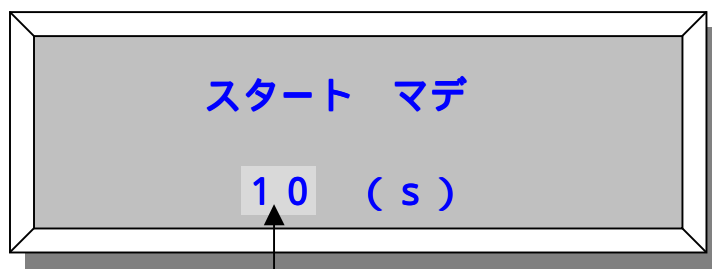
(2)、走行テスト手順

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に示されるメッセージが表示されます。



図1 メッセージの表示

スタートスイッチを押すと、図2で示されるスタート開始までの残り時間が表示されます。時間は、10秒から開始し、0秒になると走行を開始します。



10秒から0秒までカウントダウンします。

図2 モード表示画面

走行カウンタが0秒になると、図3で示される走行表示画面に切り替わります。

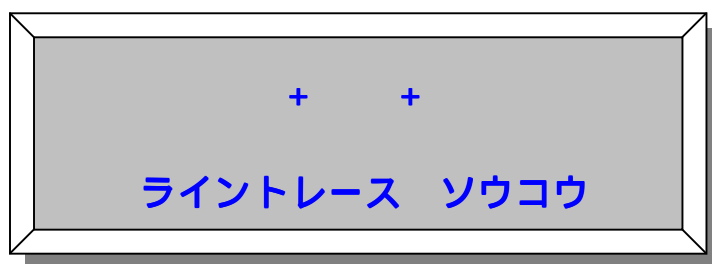


図3 走行表示画面

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(モード選択スイッチ3番:ON)、図4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4 走行表示画面

3. デモプログラムの概要

(1) ソフトウェアタイマ方式

デモプログラム「Ok1abo_1」と「Ok1abo_2」は、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5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になつ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6は、図5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_CONSTPです。また、左モータのカウント値を入れる変数はL_CONSTP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CONSTPに戻して無限ループから抜け出します。

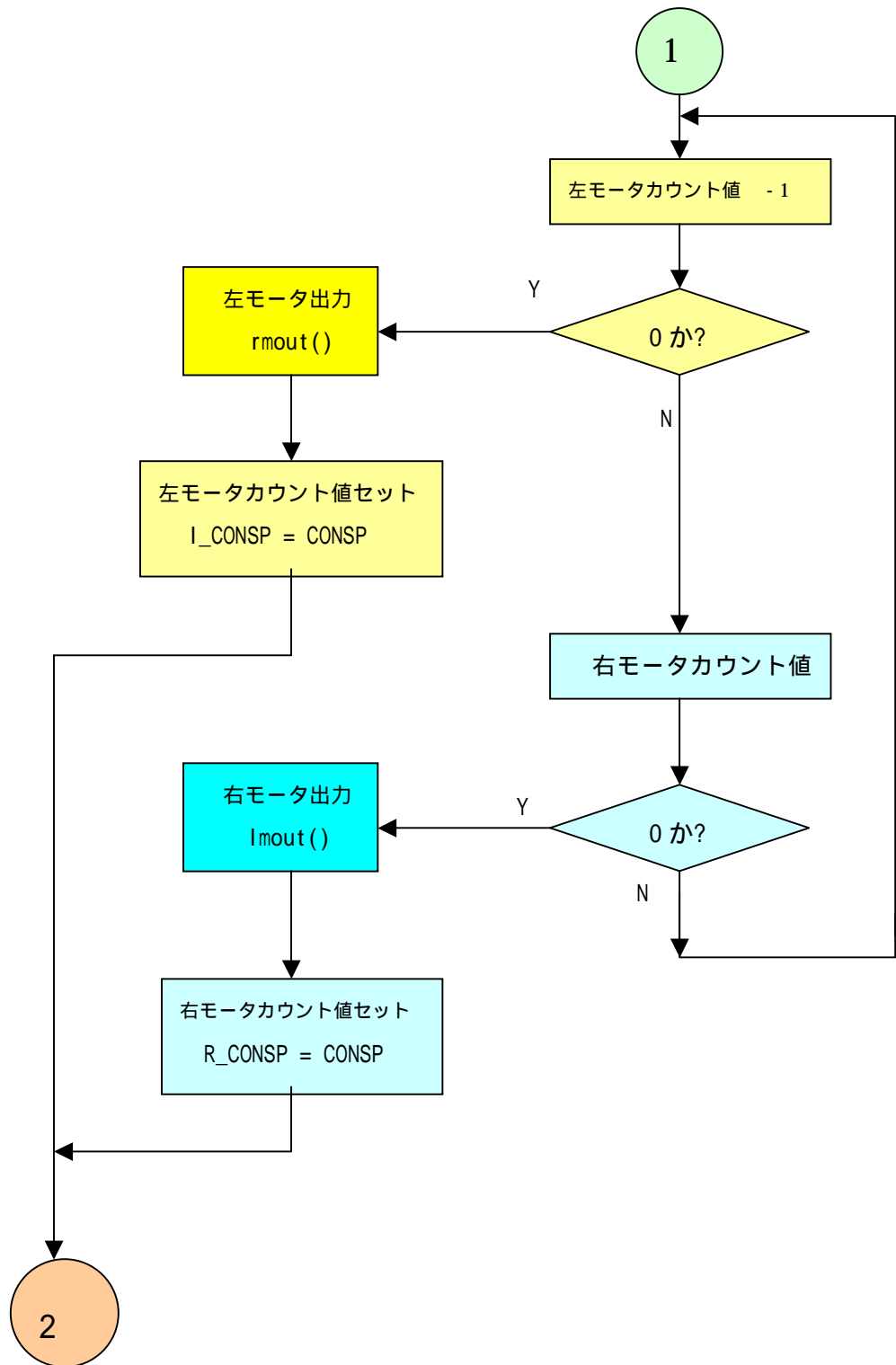


図5 ソフトタイマ方式でのモータ駆動

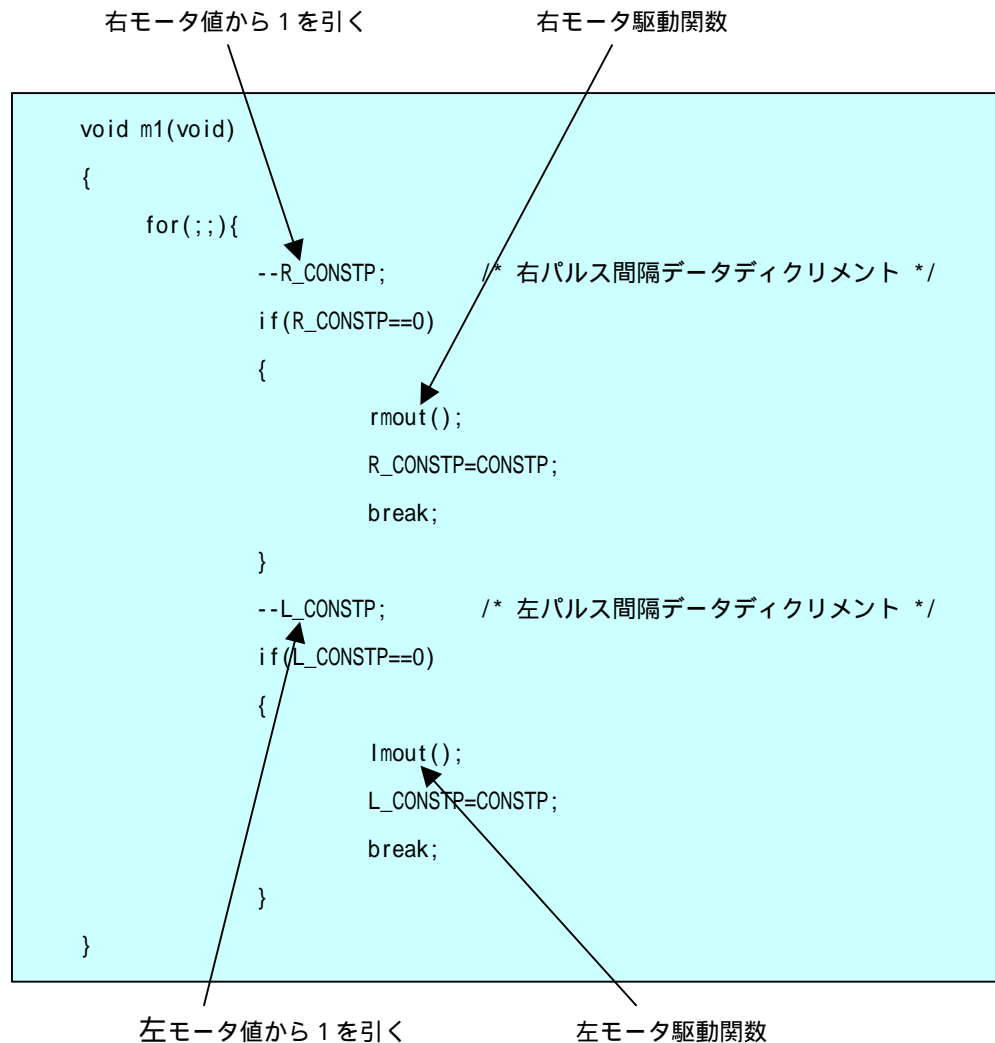


図 6 ソフトタイマ方式のプログラム記述例

(2) ステッピングモータの駆動

図 7 は、右モータの駆動プログラムです。右モータの回転方向を次のように配列データ

```

cw_tbl[4] = {0x03,0x02,0x00,0x01};

```

で表し、このデータを `rmout()` 関数により、図 6 で示されたソフトタイマ方式での時間間隔で順番に取り出します。同様に、図 8 は、左モータの駆動プログラムです。

左モータの回転方向を右モータと逆の配列順にし、次のように配列データで表し、この

```

ccw_tbl[4] = {0x01,0x00,0x02,0x03};

```

データを `lmout()` 関数により、図 6 で示されたソフトタイマ方式での時間間隔で順番に取り出します。

右モータの回転配列

```
char cwdata(char dr)
{
    char    cw_tbl[4] = {0x03,0x02,0x00,0x01};
    return cw_tbl[dr];
}
void rmout(void)                /* 右モータ出力 */
{
    port_1 = cwdata(dtr);
    dtr = (++dtr) & 0x03;
}
```

ポート 1 から出力

図 7 右モータ駆動プログラム

左モータの回転配列

```
char ccwdata(char dl)
{
    char    ccw_tbl[4] = {0x01,0x00,0x02,0x03};
    return ccw_tbl[dl];
}
void lmout(void)                /* 左モータ出力 */
{
    port_2 = ccwdata(dtl);
    dtl = (++dtl) & 0x03;
}
```

ポート 2 から出力

図 8 左モータ駆動プログラム

(3) センサ入力プログラム

図 9 は、ライン上を走行するロボットの位置を光センサが入力するフローチャートです。走行ラインからかなり右寄りに位置している場合、左モータのカウント値を大きくして左モータの速度を遅くします。この結果、ロボットは急激に左方向に向きを変えることが可能となります。また、逆に走行ラインからかなり左寄りに位置している場合、右モータのカウント値を大きくして右モータの速度を遅くします。この結果、ロボットは急激に右方向に向きを変えることが可能となります。

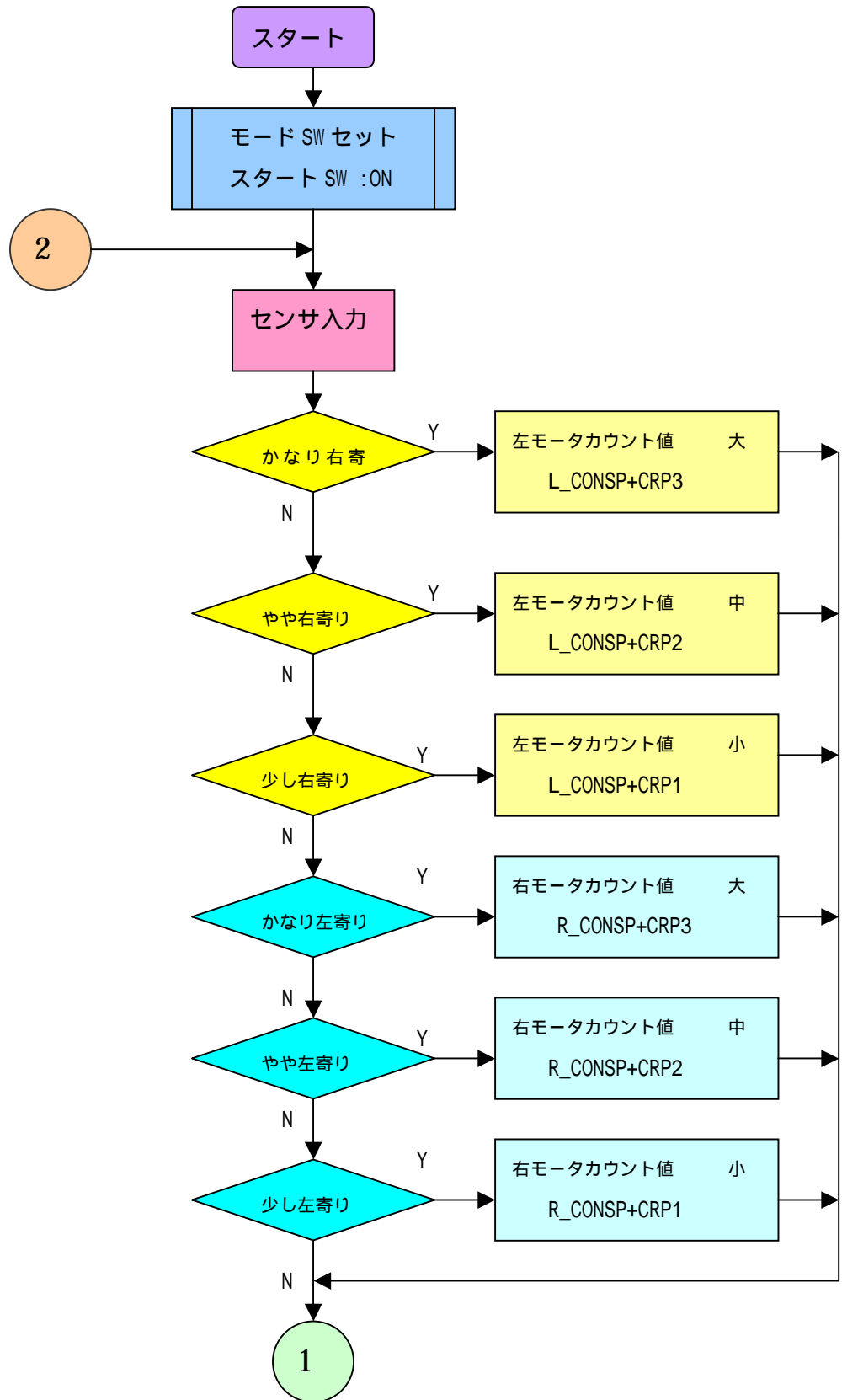


図9 センサ入力とラインコントロールプログラムのフローチャート

図10はソフトタイマ方式でのセンサ入力とラインコントロールのプログラムです。センサデータは `sence()` 関数で検出し、`switch` 文でコースからのずれを判断します。ロボットがかなり右に位置するとプログラムの「かなり左修正」の部分を見て下さい。左モータには最大のタイマデータとして `L_CONSP=L_CONSP+CRP3` を設定した後、`m1()` 関数(ソフトタイマルーチン)で実際にモータを制御します。図6からモータ制御関数 `m1()` プログラムに戻ります。設定したタイマデータをデクリメントし、ゼロになったら左モータを回転 `lmout()` 関数の実行させます。ソフトタイマを定速度データに戻して `switch` 文にリターンします。

Switch 文でセンサの状態を判断します。

```

void select(void)
{
    int    dat;
    while( 1 )                /* 無限ループ */
    {
        dat = sence();
        switch(dat)
        {
            case 1: L_CONSP=L_CONSP+CRP3;    /* かなり左修正 */
                    m1(); break;
            case 2: L_CONSP=L_CONSP+CRP1;    /* 少し左修正 */
                    m1(); break;
            case 3: L_CONSP=L_CONSP+CRP2;    /* やや左修正 */
                    m1(); break;
            case 4: R_CONSP=R_CONSP+CRP1;    /*少し右に修正 */
                    m1(); break;
            case 6: m1(); break;             /* 中央 */
            case 8: R_CONSP=R_CONSP+CRP3;    /* かなり右修正 */
                    m1(); break;
            case 12: R_CONSP=R_CONSP+CRP2;   /* やや右に修正 */
                    m1(); break;
            case 15: moff(); break;         /* 走行停止 */
            default: m1(); break;          /* 何れでもない */
        }
    }
}

```

図10 センサ入力とラインコントロールプログラム

4. 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2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2 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
<code>inline</code>	<code>__inline</code>
<code>near</code>	<code>__near</code>
<code>far</code>	<code>__far</code>
<code>asm()</code>	<code>__asm()</code>

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3 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4 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5 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 1 1 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。

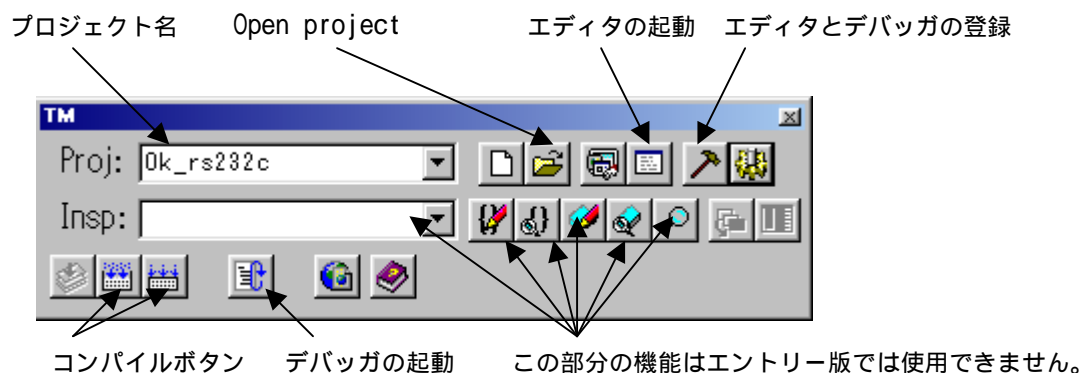





図 1 1 プロジェクトバー


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Windows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(PDFファイル)をご覧ください。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 232 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CD-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 232 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 232 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。

デバグの起動



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

図 1 1 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マニュアルをご覧ください。

5 . 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 1 2 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_2.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_2.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グ可能な実行ファイルが完成します。次に、リモートデバグ「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_2.mot」に変換し、フラッシュライターソフト「M16CF1sh」で書込み、プログラム作成は終了です。

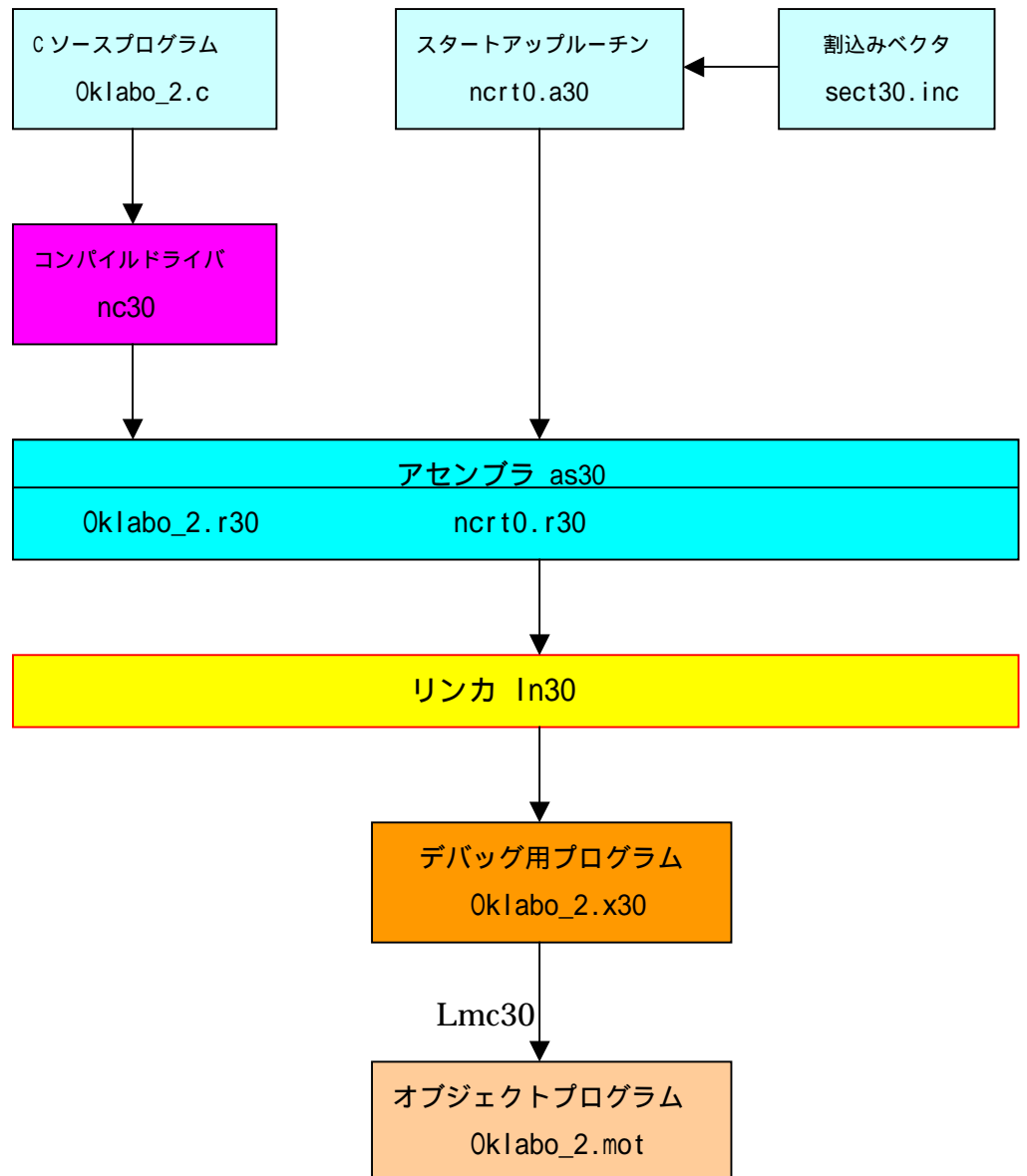


図 1 2 コンパイラ「NC30WA」のコンパイル手順

6. 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事ができます。

図 1 3 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。

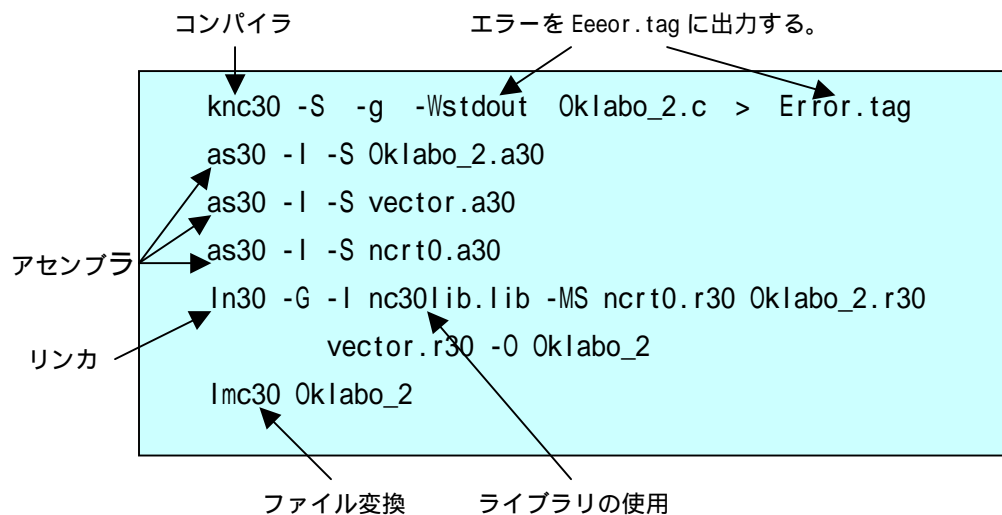


図 1 3 コンパイル用バッチファイル記述例

図 1 3 で示されるバッチファイルは、ディレクトリ「Deno2_rom」に入っているデモソフト「Oklabo_2.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 1 3 に使用されているオプションの詳細な使い方は、コンパイラ「NC30WA」マニュアルをご覧ください。

7. 1 / 0 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 1 4 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。

ポート名(変数名) ポートアドレス(絶対アドレス)

```

/*****
 *   ポートの設定 (ポート P0, タイマ A0)
 *****/

#pragma ADDRESS port_0      3e0H   /* Port P0 register          */
#pragma ADDRESS pd0        3e2H   /* Port P0 direction register */
#pragma ADDRESS tabsr      380H   /* Count start flag          */
#pragma ADDRESS udf        384H   /* Up-down flag              */
#pragma ADDRESS ta0        386H   /* Timer A0 register         */
#pragma ADDRESS ta0mr      396H   /* Timer A0 mode register    */
#pragma ADDRESS ta0ic      55H    /* Timer A0 interrupt control register */

unsigned char    port_0, pd0;
unsigned char    tabsr, udf, ta0mr, ta0ic;
unsigned int     ta0;

```

図 1 4 I / Oポートの記述例

8 . 便利な関数の活用

L C D 表示関数

デモプログラムが使用しているヘッダファイル「`lcdtime_1.h`」にはLCDを表示させるのに便利な関数が記述されています。

表示位置関数「`gotoxy(x, y)`」

表示位置関数の括弧内に記述するXは、表示する桁位置で0から15まで表示位置を記述します。また、Yは、表示する行位置で上段は0、下段は1を記述します。

`gotoxy (桁位置 , 行位置)`

表示関数「`lcdprint (配列名)`」

表示関数の括弧内には直接に表示データを記述することはできません。一旦配列で記述してから括弧内に配列名で記述します。

lcdprint (配列名)

図15は、OAKS-LABOの電源を投入後、最初に表示するメッセージの表示プログラムです。表示データは「titl_1」と「titl_2」に配列で記述します。配列「titl_1」のデータは1行目0桁位置から表示しのか。また、配列「titl_2」のデータは2行目の0桁位置から表示します。

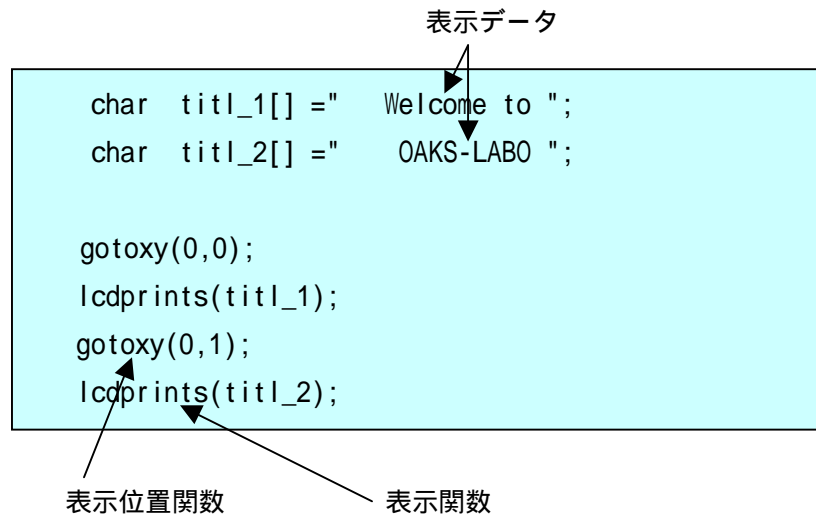


図15 LCDのメッセージ表示例

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 Ver4.0」 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。

デモプログラム(Oklabo_1 & Oklabo_2)の使い方

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3-2-13

第一千代田ビル3F

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

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