

# ROBOLABによる ロボット実習

ROBOLABによるロボット実習



情報技術科

1年 ( )番 氏名

# はじめに

今回の実習では、レゴブロックでロボットを組み立て、ROBOLAB を利用して実際に動かします。ROBOLAB (ロボラボ) は、レゴマインドストームを使って、ロボットのプログラミングをマスターできるすぐれたソフトウェアです。

<< 作業の流れ >>

## 1. 組み立て

ROBOLABの頭脳「RCX」を中心に、レゴ・ブロックで自由にロボットを組み立てます。



## 2. プログラミング

パソコンの画面上で、アイコンをドラッグ&ドロップしながらプログラミング。



## 3. ダウンロード

設計したプログラムは、パソコンに接続した赤外線タワー「IRT」から、ワイヤレスで「RCX」へダウンロードします。



## 4. アクション

ダウンロードが終われば準備完了。「RCX」のRUNボタンを押しプログラムを実行してみよう。



# 第1章 シンプルカーの組立

## 1・1 部品の名前

レゴブロックには沢山の部品があります。

ここでは、よく使う部品を絞り込んで紹介してみましょう。

	<p>普通のブロックと同じ厚みを持っていますが横に穴が空いています。</p> <p>長さはポッチの数が1, 2, 4, 6, 8, 10, 12, 14, 16と9種類あります。</p> <p>幅は全部ポッチ1つの分です。長さ2のビームには穴の空き方によって3種類あります。</p> <p>穴が1つのビーム、穴が2つのビーム、穴が1つなのですが形が十字をしているもの。</p> <p>それ以外のビームは穴は丸です。</p>
ビーム	<p>プレートはブロックの厚さの1/3の物で3枚重ねると普通のブロックの厚みと同じになります。</p> <p>形は穴のあるもの、無いものと沢山の種類があります。</p> <p>また、幅の広いもの細いもの、多種多様です。</p> <p>左のイメージは代表として、2×4の穴ありプレートに登場してもらいました。</p>
	<p>ペグやピンと呼んだりします。</p> <p>10種類くらいあるのですが、ここではよく使う3種類を紹介します。</p> <p>灰色のペグは、ビームなどに入れるとゆるゆるしています。ブラックペグと呼ばれます黒いペグは、ビームの穴などに入れても硬くゆるさがありません。半部分が灰色のペグは、残りの半分は断面が十字になっています。</p>
プレート	ペグ(ピン)

	<p>8 歯ギヤは、俗にピニオンギヤとも呼ばれ、直接モーターに接続して利用する場合も多くあります。</p> <p>24 歯ギヤは、ギヤの中に穴が 4 つ空いているのでその利用法を考えると いろいろなアイデアがわいてきます。</p>
<p>ギヤ</p>	<p>ギヤなどの軸になしたり、いろいろな利用法があるパーツです。</p> <p>長さは、短い物より 2、 3、 3 ポッチが付いている 4、 5、 6、 8、 10、 12 があります。</p>
	<p>ビームを押えたりスペースを調整する為に利用します。</p>
<p>シャフト</p>	
	
<p>ブッシュ</p>	

## 1・2 フレームを作る

1



後でモーターを乗せる部分を作ります。

4x6のプレートの上にビームを載せます。また、ビームにはペグをいれて、次の工程で長さ4のビームと接続できるようにしておきます。

2



1×6のプレートでRCXを搭載するための高さをあわせるためと、補強を行います。

3



左右のフレームを作ります。  
ここでは、長さ 16 の一番長いビームと、長さ 6 のビームをペグを使ってフレーム用に長さを調整しています。

4



前の方は、 $1 \times 8$  のプレートで、長さ 4 のビームをはさみ。  
後ろの方は、 $1 \times 6$  のプレートではさんで補強します。

このようにして、フレームとして強化しておきます。

## 1・3 タイヤをつける

5



長さ 12 (一番長い) のシャフトを使って車軸をつけます。

また、ブッシュをつけてタイヤとフレームの間に隙間を作ります。

後輪の車軸にはギヤ (24 歯) も取り付けます。

ギヤの反対側にはブッシュは 2 つつけておきます。

6



前と後ろに、タイヤをつけます。

ここではギヤのある方を大きなタイヤにしていますが、逆にしてみても面白いかもしれません。

## 1・4 モーターと RCX を搭載する

7



モーターに小さなギヤ( 8 歯 )を付け、後輪のギヤとかみ合うようにモーターを搭載します。

8

RCX を乗せて、ケーブルでモーターと RCX のアウトポート A と接続します。



完成したシンプルカー



## 1・5 RCX とは

レゴブロックで作成したロボットにはモータやランプなどの出力用のパーツや、タッチセンサやライトセンサなどの入力用のパーツを組み込みます。それらの入出力用のパーツをコントロールするのがRCXです。



(ポート)

ラベル	色	名称	機能
1、2、3	濃い灰色	入力	ライト、タッチなどの各センサの接続ポイント
A、B、C	黒	出力	モータやランプなどの出力パーツの接続ポイント

(ボタン)

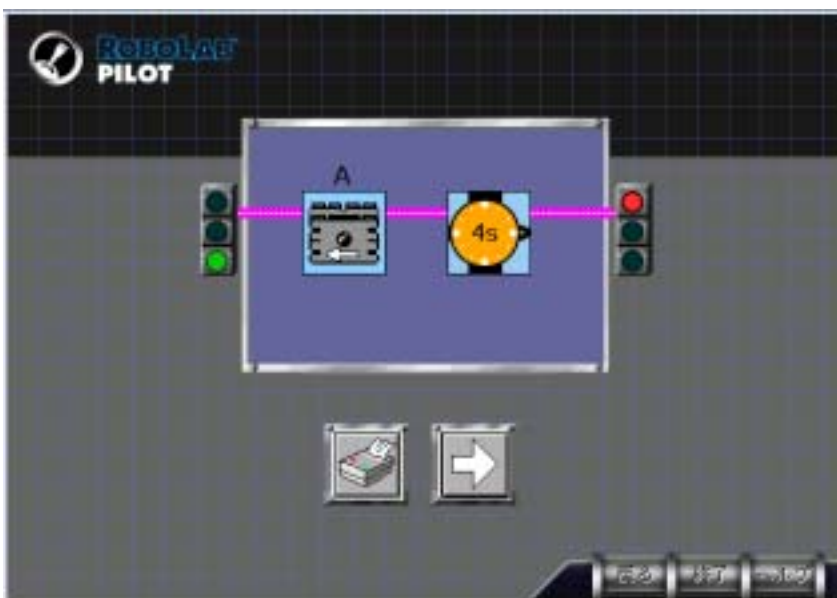
ラベル	色	名称	機能
On・Off	赤	オン/ オフ	RCXをオン/オフします
View	黒	表示	表示ウィンドウでモニタするポートを選択できます
Prgm	灰色	プログラム	RCXに実行させる特定のプログラム(1~5)を選択します
Run	緑	実行/ 停止	RCXのプログラムの実行を開始したり停止します

## 第2章 プログラムミング I

### Pilot (パイロット) 段階のプログラミング

#### 2・1 プログラムを見てみよう:Pilot1

最初に入門レベルである Pilot 1 にさわって、プログラムをダウンロードしてみよう。



(アイコン説明)

				
プログラム開始	プログラム終了	モータ回転	時計 (時間待ち)	ダウンロード

(動作説明)

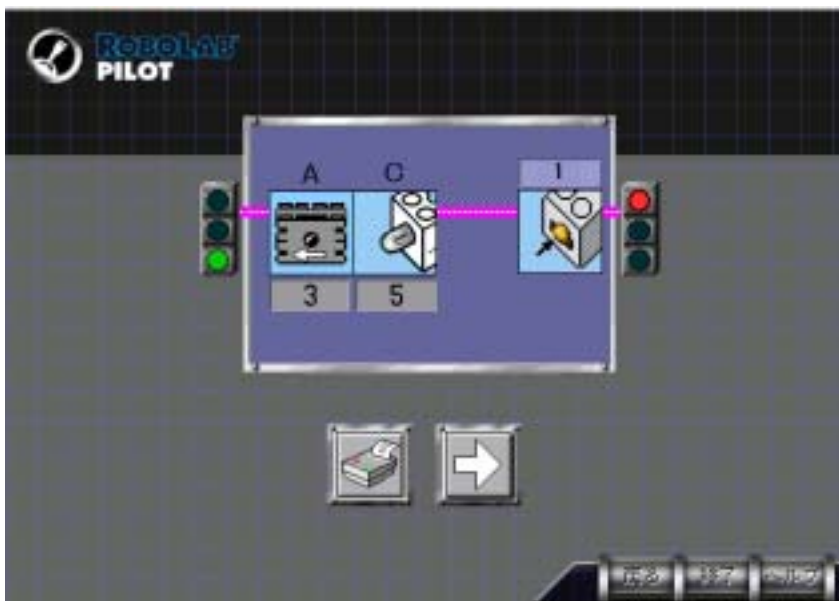
時計で指定した時間だけモータが回転します。

やってみよう

1. 時計コマンドをクリックして、モータが回転する時間を変えてみよう。
2. ダウンロード中に RCX を TR タワーからはなしてみよう。

## 2・2 2つのポートを使ってみよう:Pilot2

Pilot2 を用いて A と C の2つのポートとタッチセンサを使い、モータの  
パワーレベルも変更してみよう。



(アイコン説明)

	
ランプ	タッチセンサ

(動作説明)

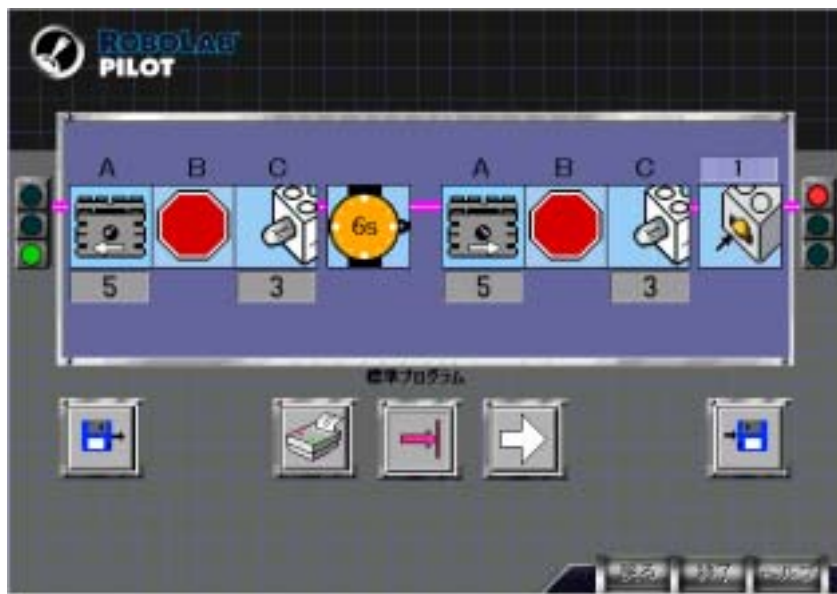
A ポートのモータを3レベルで回転させ、ポートCのランプを5レベルで点灯させる。ポート1のタッチセンサが押されて終了する。

やってみよう



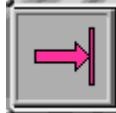


1. 2種類のタッチセンサコマンドを試してみよう。
2. タッチセンサコマンドを時計に変えて、各パワーレベルの移動距離を調べてみよう。

## 2・3 2ステップのプログラムを作ろう:Pilot3

Pilot3 は全ポートが利用できます。2ステップのプログラムをつくって保存してみよう。



(アイコン説明)

				
ファイルロード	ファイル保存	1回実行	連続実行	印刷

(動作説明)

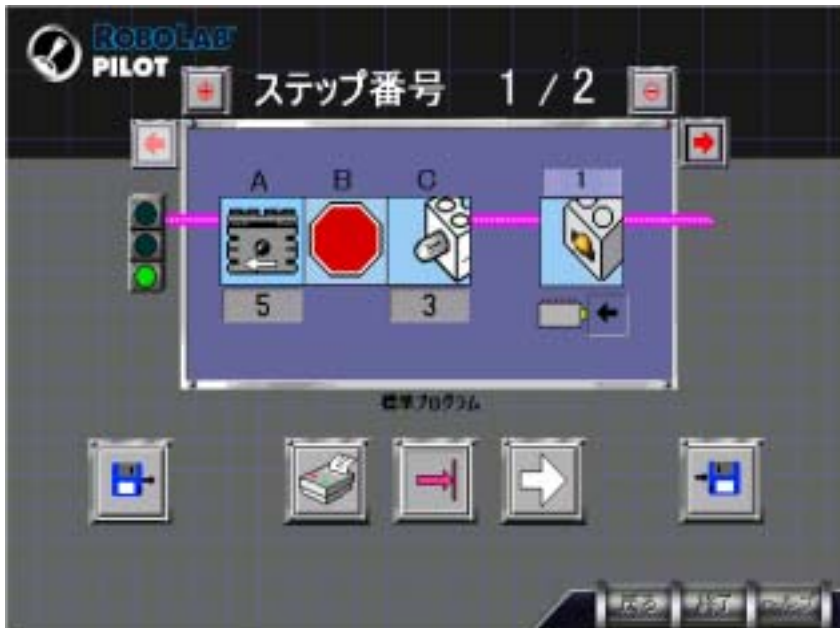
Aポートのモータ、Cポートのランプを6秒間オンにします。そのステップが終わった後、タッチセンサが押されるまでモータの向きを反転させて回転し続けます。

やってみよう

1. 連続実行コマンドに変更し、実行してみよう。
2. フロッピーディスクに保存して、プログラムを印刷してみよう。

## 2・4 複数ステップのプログラムを作ろう:Pilot4

Pilot 3を発展させて、複数（3ステップ以上）のステップのプログラムを作成してみよう。また、ライトセンサの働きについて調べてみよう。



(コマンド説明)



(動作説明)

ステップ番号 1 / 2 . . . タッチセンサが押されるまでポート A のモータ回転させ、ポート C のライトをオンする。

ステップ番号 2 / 2 . . . ポート A のモータの方向を逆に回転させ、ライトセンサが 5 5 より大きい値を読み取り終了する。

やってみよう

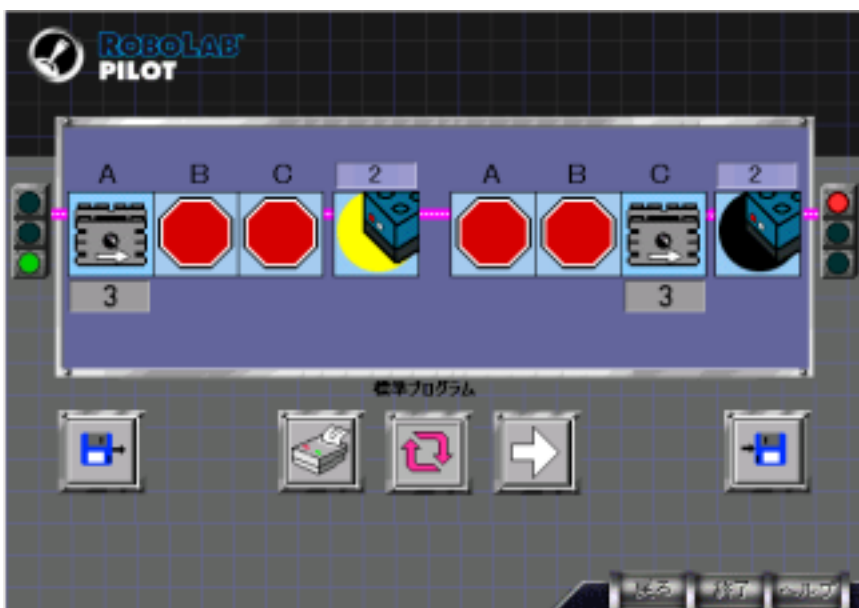
- 1 . 黒い線で止まるようにプログラムを改良してみよう。
- 2 . 黒い線に沿って走るプログラムを作成してみよう。

2 . のような動作のことを、ライントレーサーといいます。

<ヒント>

ここでは、Pilot 3 を利用します。

ポート A とポート C にモータを取り付け、ポート 2 にライトセンサを取り付けます。



# 第3章 プログラミング

## Inventor (発明家) 段階のプログラミング

### 3・1 コマンドアイコンをつないでみよう:Inventor1

簡単なコマンドアイコンをつなぎ合わせて、ポート1のタッチセンサが押されるまで、モータAとモータCを回転させてみよう。



(作成手順)

ダイアグラムウィンドウにコマンドアイコンを貼り付けます。

[ウィンドウ]から[ツールパレット表示]をクリックし、ストリングツールを用いて、コマンドどうしをつなぎます。

間違いが一つでもあると、 (壊れた矢印) が表示され実行できません。



コマンドアイコン



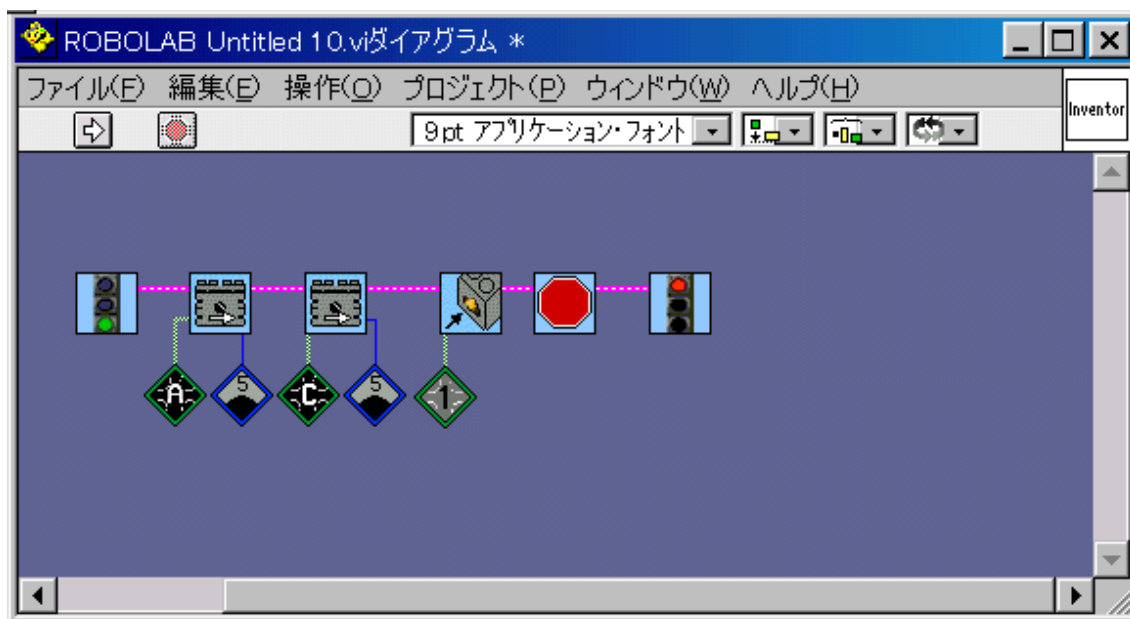
ストリング  
ツール

ツールパレット



## 3・2 モディファイの設定を試みよう:Inventor 2

汎用コマンドアイコンにモディファイのメニューパレットの設定をし、きめ細かい制御プログラミングを試みよう。



(モディファイの設定)

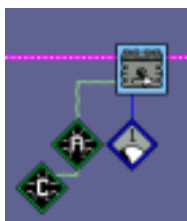
ポートの位置、パワーレベル、および定数を指定するときに、モディファイを使用します。モディファイは、コマンドのサブメニューに入っています。



やってみよう

モータ A と C のパワーレベルを 1 秒ごとに上げていくプログラムを作成してみよう。

<ヒント>



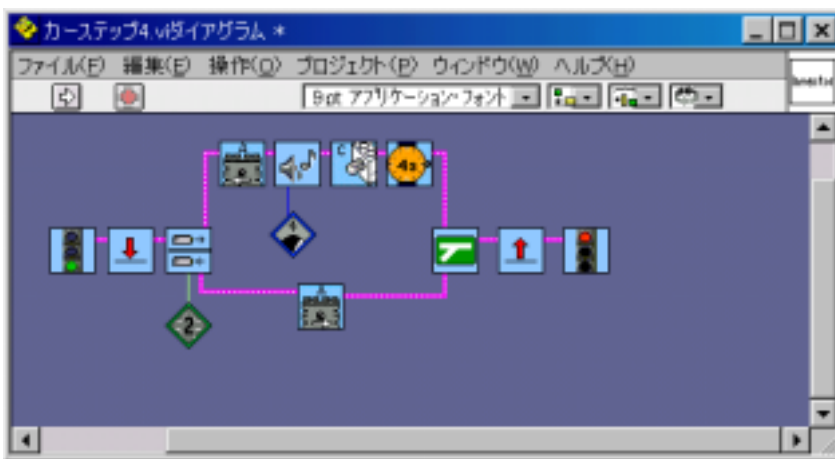
ポート A とポート C にパワーレベル 1 で接続されているモータ



### 3・3 制御構造を考えてみよう:Inventor 3

さまざまな制御構造やミュージックコマンドを用いて、高度なプログラムに挑戦してみよう。

(制御構造の例)



やってみよう

タッチセンサの状態（押す、放す）に対してどのような動作をするのか調べてみよう。

(Music の例)

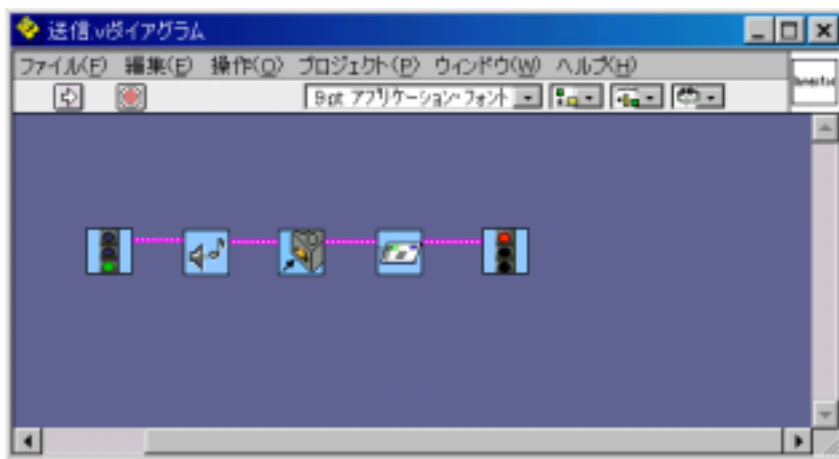
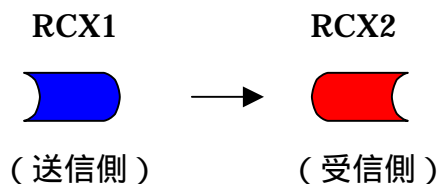


やってみよう

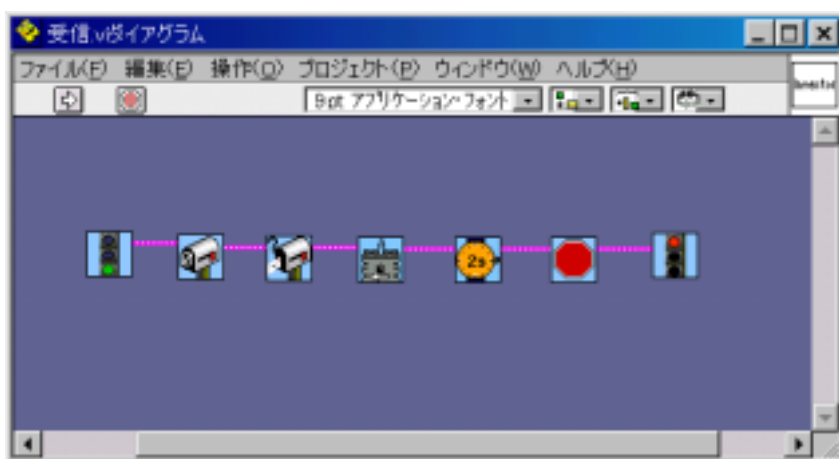
ミュージックコマンドを用いて、自分の好きな曲を演奏させてみよう。

### 3・4 RCX 間の通信を試みよう:Inventor 4

RCXごとにそれぞれのプログラムをダウンロードして、2つのRCXを用いて通信を試みよう



送信側( RCX 1 )



受信側( RCX 2 )

#### やってみよう

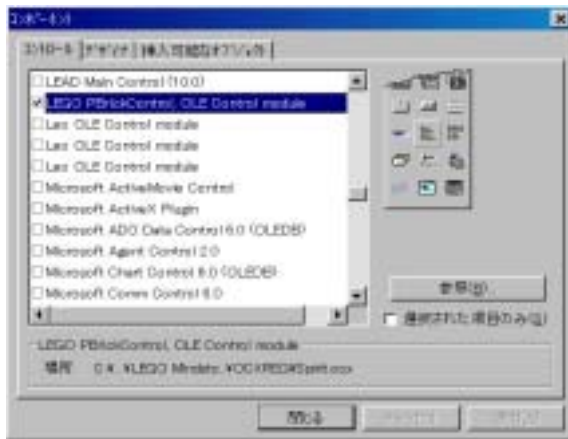
上記のプログラムを用いて、他のチームと通信させてみよう。

実行時は、必ず両方のRCXの赤外線レシーバーが向き合っていること。

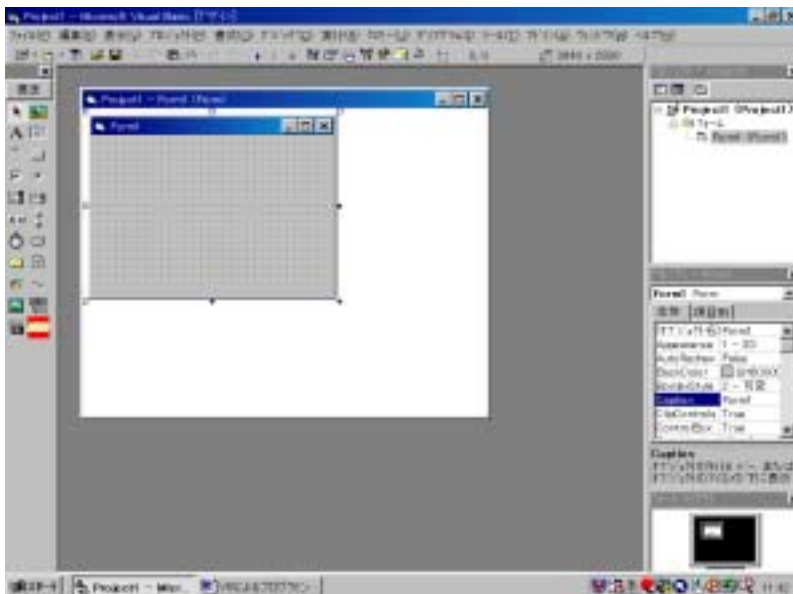




3. 次に、コンポーネントダイアログの中の“LEGO PBrickControl,OLE Contol module”がチェックされていることを確認してから、[OK]ボタンを押します。



これで VisualBasic から Sprit.ocx が利用可能になります。ツールボックスパレットに赤い“LEGO”のアイコンが追加されているはずですが。



## 4・2 最も簡単なプログラムを実行してみよう

Sprit.ocx が利用可能になったら、いよいよ VisualBasic によるプログラミングの開始です。次の例題に挑戦してみましょう。

フォーム上の左のボタンが押されたらモーターを On にし、右のボタンを押されたらモーターを Off にするプログラム

<プログラムリスト>

```
Private Sub cmdOff_Click()
    RCX.Off"0"
End Sub
```

} Aポートに接続されたモーターを Off

```
Private Sub cmdON_Click()
    RCX.On "0"
End Sub
```

} Aポートに接続されたモーターを On

```
Private Sub Form_Load()
    RCX.InitComm
End Sub
```

} RCX とのコミュニケーションの初期化

```
Private Sub Form_Unload(Cancel As Integer)
    RCX.CloseComm
End Sub
```

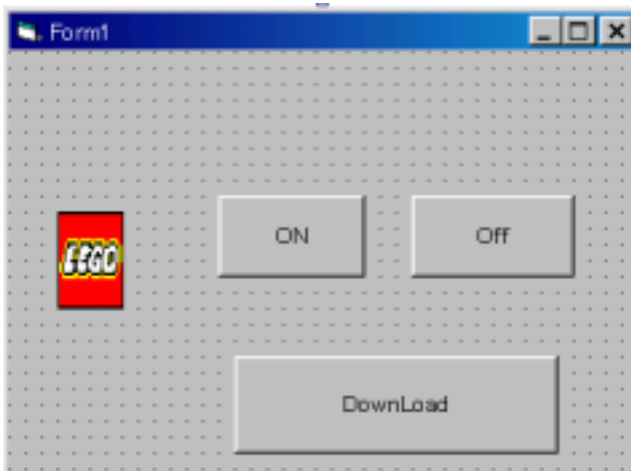
} RCX とのコミュニケーションの終了

モーター関連のメソッド

メソッド	説明
On	モーターを回す
Off	モーターを止め、ブレーキのかかった状態にする
Float	モーターを止め、ブレーキのかかっていない状態にする
SetFwd	回転方向を正転に設定する
SetRwd	回転方向を逆転に設定する
AlterDir	回転方向を逆にする
SetPower	パワーを指定する

## 4・3 ダウンロード可能なプログラムに挑戦

最も簡単なプログラムを改造して、RCX にプログラムをダウンロードしてからモーターを回すようにしてみよう。



<プログラムリスト> ...ダウンロードの部分

Private Sub cmdDownLoad\_Click()

RCX.SelectPrgm 4           ...プログラムを選択

RCX.BeginOfTask 0         ...ダウンロードされるタスクの始まり

RCX.On "0"                 ...A ポートに接続されたモーターを回す

RCX.Wait 2, 100           ... 1 秒間待つ

RCX.Off "0"                ...A ポートに接続されたモーターを止める

RCX.EndOfTask             ...ダウンロードをする

RCX.PlaySystemSound 5    ...ダウンロードが終了したと知らせる音を出す

End Sub

タスク関連のメソッド

SelectPrgm	RCX のプログラムスロットを選択する
BeginOfTask	どのタスクに対して、プログラムをダウンロードするかの初期化を行う
EndOfTask	ここまでが RCX にダウンロードされる
StartTask	引数を 1 つ持ち、引数で指定されたタスクを実行する
StopTask	引数を 1 つ持ち、引数で指定されたタスクを停止する
StopAllTask	実行中のすべてのタスクを停止する
Wait	指定した時間だけ 1/100 秒単位で待つ
PlaySystemSound	即時メソッドとして、RCX からサウンドを鳴らす

## 第5章 オリジナル作品の製作

最後に、自分たちだけのオリジナルロボットをつくってみよう。各チームで協力しながら、アイデアに満ちた作品を期待しています。

### 【活動のフローチャート】

