

特集 「エンターテインメントとAI」

LEGO Mindstorms の世界

LEGO Mindstorms World

山本 昌弘
Masahiro Yamamoto
法政大学国際文化学部
Faculty of Intercultural Communication, Hosei University.
yamamoto@i.hosei.ac.jp

Keywords: LEGO, Mindstorms, robot, toy, LEGO block.

1. 概要

Mindstorms は米国、MIT のメディアラボのセイモア・パパート教授のところで開発されたものである。過去からある LEGO ブロックに新たにコンピュータを埋め込んだブロックをつくり、LEGO でつくられた玩具ロボットを、教育用の言語である LEGO 型言語でロボットの動作を作成するもので、自ら知識を獲得していく学習道具としてつくられたものである。

さまざまなブロックが用意されており、自分のアイデアの形状・行動のロボットを組み立てることができる。これによって、自らさまざまな形状を想像して、独自のロボットを形づくるのが可能になる。小さなコンピュータを搭載しており、ロボットがどのように行動するかを規定してプログラムとして作成し、ロボットに知恵を与えることができる。これでロボットの動作を形成することが可能になる。プログラム作成ツールが準備されているので、プログラミングの能力を養成することができる。プログラム環境としては、アセンブラ言語に対応するビジュアルなブロック型言語と Visual Basic などの ActiveX を使用しても作成できる。

このように、Mindstorms は物体を想像する力と、行動を規定する力、プログラミング力を養成する知的な教材であるといえる。

本論では、LEGO Mindstorms の LEGO ブロックの構造、プログラミングである RCX コード、および実行環境について紹介する。

2. LEGO ブロック

Mindstorms の RIS (Robotics Invention Systems) ではロボットをつくるためのパーツとして多くのブロックが用意されている。子供の頃に誰もがつくって遊んだことがあるプレートやブロックに加えて、RIS では、ビーム、コネクタ、シャフト、ギヤ、タイヤなど 700 種以上のブロックが用意されている。したがって、ロボット

の形体は自由に作成でき、ギヤとモータを使用して自由に走行でき、かつ、アームなどを動かすことが可能になり、動くことができるペットロボットをつくるが可能になる。これまでの動くことができない静止型の LEGO ブロックとは大きく異なる点である。さらに、大きな特色は頭脳に当たる RCX、動作をするためのモータや外界と交流するためのセンサが備えられている。RCX は動作を指示するプログラムを内蔵することができ、プログラムに従って自立して動作することが可能である。このプログラムを内蔵することで知的な行動をする玩具ロボットをつくることができ、外界と交流しながら動作する自立型のロボットの作成が可能になる。また、RCX は内蔵したプログラムの指示で動作すると同時に、外界の例えばパソコンからの指示で動作させることも可能で、遠隔制御型のロボットを作成することもできる。

2.1 ロボットを構成する基本ブロック

これまでの LEGO で使用したプレートやブロックのほかに、RIS では、ビーム、コネクタ、シャフト、ギヤ、タイヤなどのパーツがあり、ロボットの形は、さまざまなパーツを使うことによって、個人の好みの形のロボットをつくるのが可能である。これは、LEGO Mindstorms の特色である。

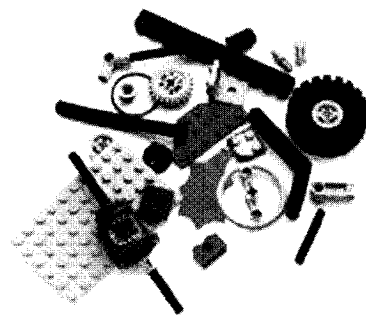


図1 LEGO ブロックのパーツ

2.2 RCX

RCX は RIS でロボットの頭脳になるブロックである。RCX には四つのボタンがあり、On/Off ボタンは電源の

切換え、View ボタンは RCX 画面の表示切換え、Prgm ボタンはプログラムの切換え、Run ボタンはプログラムの実行ボタンである。また、モータをつけるための三つの出力ポート、センサをつなぐための三つの入力ポートを持っており、モータやセンサを接続して動作を制御できる。

8 bit の CPU を搭載しており、16 MHz で動作する。CPU には 16 KByte の ROM を持ってあり、パソコンの OS に相当するファームウェアを RAM にロードするドライバを保持している。また、32 KByte の RAM には RCX を動作させるファームウェアとユーザのプログラムを保持できる。ユーザのプログラムは最大 5 個まで内蔵できる。

OS であるファームウェアは、パソコンの OS の働きをするものである。ユーザからのプログラムをダウンロードしたり、RCX の Prgm ボタンが押されたときに実行するプログラムを切り換える、Run ボタンで該当するプログラムを動かす、On/Off ボタンが押されたとき電源を切るなどのボタン操作に対する動作を RCX に指示する、パソコンにセンサの値を知らせる、などの制御を行う。マイクロプロセッサを内蔵していることでこれらの動作をすることができ、LEGO がこれまでの単なる組立て玩具から、知能を持った知的な玩具ロボットとして機能する。

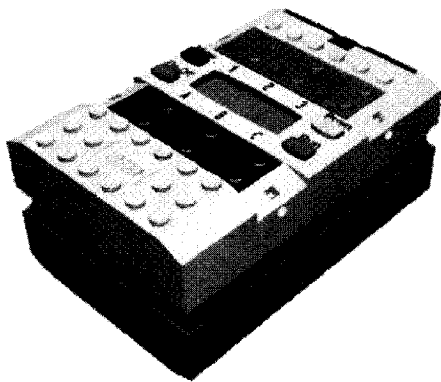


図2 RCXの外観

2.3 モーター

回転する軸の部分はシャフト状になっており、ギヤをつけたり、コネクタをつけたりしてロボットが持っているアームを動かしたり、タイヤをつけてロボット自体を動かすのに使用する。モータはパワーを変えることができ速度を変更できる。RCX は 3 個の出力ポートを持っており、モータを接続して、RCX コマンドで指示することができる。

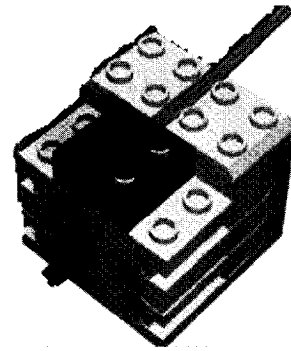
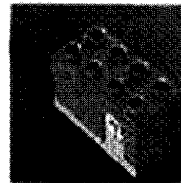


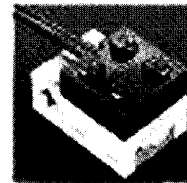
図3 モータ

2.4 センサ

センサにはタッチセンサとライトセンサなどが用意されている。タッチセンサはセンサブロックの先端の黄色い突起の部分が何か当たってへこんでいるか否かを調べることができる。ライトセンサは光を感知して、明るさを調べることができる。RCX は三つの入力ポートを持っておりこれらのセンサを接続して使用する。そのほかに、温度センサ、ローテーションセンサなどもある。



ライトセンサ



タッチセンサ

図4 センサ

2.5 IRタワー

RCX は赤外線インタフェースを持っており、外界から信号を受け取ったり、信号を送ったりできる。外部 PC の RS-232C インタフェースを使用して IR タワーを用いて赤外線と通信するための IR タワーが用意されている。これによって、外部のパソコンで作成したプログラムを受け取る（ダウンロードする）ことができる。プログラムは 5 個内蔵できる形になっている。また、このインタフェースを用いて、外部のパソコンなどから直接制御することで、遠隔制御型のロボットを作成できる。さらに、ほかの RCX へ信号を送ったり、受け取ったりすることも可能で、これによって RCX 間の交

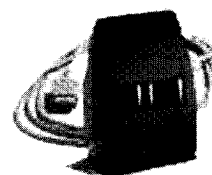


図5 IRタワー

信が可能で、RCX 間の協調型のロボットを作成できる。
これらのパーツを使って、さまざまなロボットを作成できる。図 6 にその一例を示す。

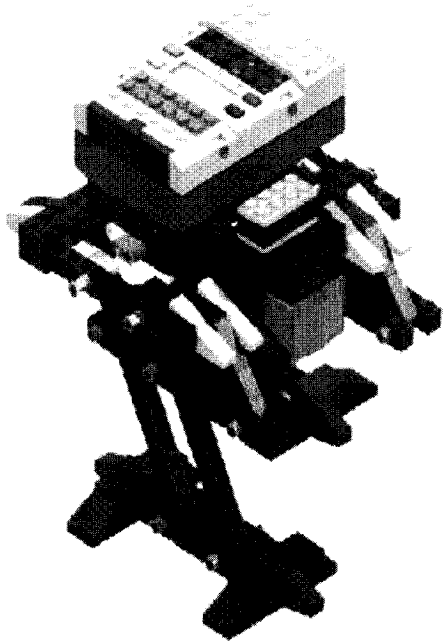


図 6 ロボットの例 (Jin Sato のプログラムより引用)

3. RCX コード

Mindstorms の頭脳は RCX であるが、プロセッサを備えていることで、プログラムを内蔵でき、それに従って、動作することができる。RIS では RCX コードでプログラムを作成するが、外部のパソコン画面状で作成し、IR タワーを用いて Mindstorms にダウンロードして実行する。

RCX コードを作成するプログラムは Windows や Macintosh で動作するソフトウェアがつくられており、容易に作成できるようになっている。

RCX コードは通常のコンピュータで言えばいわゆるアセンブラ言語である。その特徴はブロック型の言語になっており、パソコンの画面上で命令に対応するブロックを選んで LEGO のように組み合わせることでプログラムする形になっており、ビジュアルな言語である。RCX コードは LEGO 型言語といえる。

RCX コードは、プログラムブロック、コマンド、スタックコントローラ、センサウォッチャ、マイコマンドを備えており、アセンブラ言語のように単に制御手順を手続き形式で作成することに加え、高級言語が備えている関数表現や構造化も可能になっている。

3.1 プログラムブロック

RCX プログラムはこのプログラムブロックで開始し、コードは通常順番に実行される。プログラムブロックを右クリックすると、ブロックが拡大され、このプログラムを外部ファイルに保存しておくことや RCX にダウンロードを指示できる。

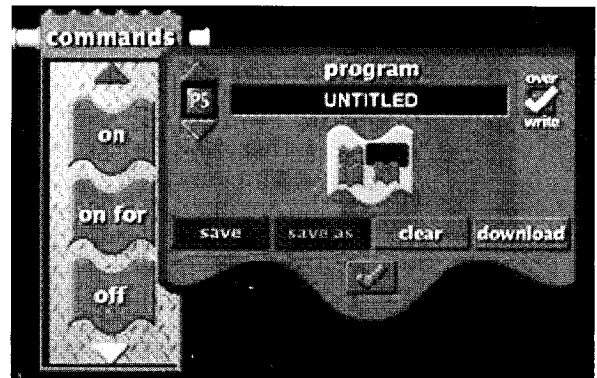


図 7 プログラムブロック

3.2 コマンド

コマンドは RCX に命令を伝えるためのブロックで、モータの制御、カウンタやタイマの制御、RCX とのやり取りなどを指定する。使用できるコマンドを表 1 に示す。これらのコマンドブロックをつなぎ合わせてプログラムする。コマンドブロックは右クリックすると、このブロックのポートの設定や数値の設定など、各種の属性を指定できる。

表 1 コマンドの仕様

名前	機能
on	モーターの電源をオンにする
on for	指定した時間モーターの電源をオンにする
off	モーターの電源をオフにする
set power	モーターのパワーを変える
set direction	モーターの回転方向を決める
reverse direction	モーターの回転方向を逆転させる
wait	指定した時間プログラムを止める
deep	ビーブ音を鳴らす
tone	音を鳴らす
add to counter	RCXに内蔵されているカウンタに1を加える
reset counter	カウンタをリセットして0にする
reset timer	タイマーをリセットして0にする
send to RCX	別のRCXへメッセージを送る
reset message	送られたメッセージをリセットして消去する

コマンドは LEGO ブロックのように組み合わせることでプログラムをしていく。図 8 にそのようすを示す。

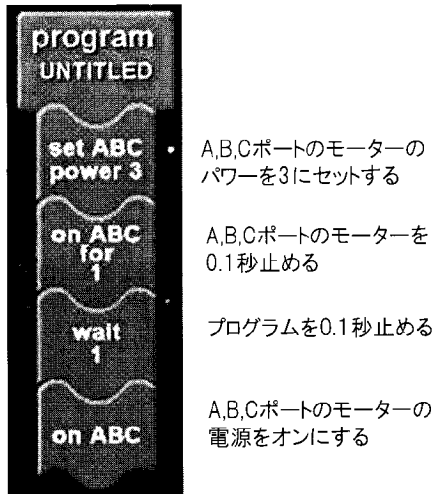


図8 コマンドの例

3.3 スタックコントローラ

スタックコントローラはプログラムの流れを制御するブロックで、例えば、センサの状態監視、カウンタ、タイマ、RCX メッセージの監視ができる。また、これらとは独立に、一般的なコマンドの繰り返し動作にも使用できる。

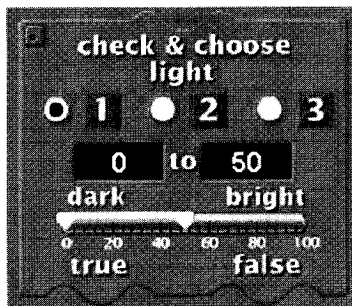


図9 スタックコントローラ

表2 スタックコントローラの仕様

名前	機能
choose a sensor	3つのスタックコントローラ(check & choose, repeat while, wait until) が使用するセンサをRCXに内蔵されているセンサから選ぶ
check & choose	センサの状態を調べ、プログラムの流れを2つに分岐させる
repeat	接続したコマンドを指定した回数繰り返し実行する。回数は2~255まで。また、サイコロでランダム回数を指定できる。いわゆるdo loopである
repeat while	選んだセンサの状態を調べ、trueである間、このスタックコントローラに接続したコマンド列を実行する。いわゆるwhile文である
repeat forever	接続したコマンド列を実行し続ける
wait until	選んだセンサの状態を調べ、trueになるまで接続したブロックを実行するのを待つ

3.4 マイコマンド

通常のプログラムでいえば、サブルーチンや関数のような働きをするもので、コマンドとスタックコントローラを使ってつくったブロック群であり、名前を付けて保存したり、別の場所で再利用できる。



図10 マイコマンド

3.5 センサウォッチャ

タッチセンサやライトセンサの状態を監視したり、RCX が内蔵しているカウンタやタイマの内容を調べたり、RCX メッセージなどの状態を監視するブロックである。表3にセンサウォッチャの仕様を示す。

表3 センサウォッチャの仕様

名前	機能
touch	タッチセンサが押されているかどうかを監視する
Light	ライトセンサの明るさを監視し、指定範囲に対応して実行する
RCX	別のRCXからのメッセージを監視し、メッセージの値が指定した範囲にあると、接続されているコマンドが実行される
counter	指定した範囲にあるかを監視し、範囲にあると実行される
Timer	タイマーの値を監視し、指定した範囲にあれば、コマンドを実行する

4. プログラミング開発環境

Mindstorms ではアセンブラ言語に対応する RCX コードを作成してプログラムをつくる環境に加えて、標準プログラミング環境として Sprit.ocx という ActiveX コントロールを使用できるようになっている。Sprit.ocx には RCX を制御する機能がまとめられており、Visual Basic などの ActiveX に対応するプログラミング言語から利用できる。Sprit.ocx を使用すると、ハードウェアの知識なしにプログラムが可能になる。Sprit.ocx には

表 4 タスク関連のメソッド

名前	機能
SelectPrgrm	RCXのプログラムスロットを選ぶ
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する
EndOfTask	ここまで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	実行中の全てのタスクを停止する
Wait	指定した時間だけ待つ
PlaySystemSound	即時メソッドとして、RCXからサウンドを鳴らす

表 5 モータ関連のメソッド

メソッド	機能
On	モーターを回す
Off	モーターを止め、ブレーキが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

RCX を動作させるメソッド以外に、タスク関連のメソッドも用意されている。

表 4 はタスク関連のメソッド、表 5 はモータ関連のメソッドを示す。

5. RCX の動作環境

Mindstorms の頭脳である RCX は、パソコンで作成した RCX プログラムをダウンロードしてそのプログラムを独自で動作させる方法と、ほかのパソコンを含めたモバイル機器からコマンドを RCX に送信することによって操作する方法である。ダウンロードして動作させるときは、RCX は自立型のロボットとして行動する。RCX プログラムを RCX にダウンロードするには、プログラムブロックから手動でする方法と Visual Basic からプログラムで送信する方法がある。

```
Private Sub cmdDownload_Click()
    Spirit1.InitComm 'RCXとの通信の初期化
    Spirit1.SelectPrgrm 4 'プログラム5を選択
    Spirit1.BeginOfTask 0 'ダウンロードされるタスク開始
    Spirit1.On "0" 'Aポートのモーターを回す
    Spirit1.Wait 2, 100 '1秒間待つ
    Spirit1.Off "0" 'Aポートのモーターを止める
    Spirit1.EndOfTask 'ダウンロードされるタスク終了
    Spirit1.PlaySystemSound 5 'ダウンロードの終了の通知
End Sub
```

図 11 Visual Basic によるダウンロードプログラム

図 11 は Visual Basic を用いたダウンロードプログラムの例を示す。

一方、モバイル機器から操作するには、パソコンなどのシリアルポートに接続した IR タワーを使用して、RCX へ直接コマンドを送信することで可能であり、このときには会話型の遠隔制御型のロボットになる。

図 12 は Visual Basic を用いて PC から制御するプログラムの例を示しており、RCX 上のプログラム番号を指定して、そのプログラムを実行させることが可能である。

```
Private Sub cmdDownload_Click()
    Spirit1.InitComm 'RCXとの通信の初期化
    Spirit1.SelectPrgrm 4 'プログラム5を選択
    Spirit1.BeginOfTask 0 'ダウンロードされるタスク開始
    Spirit1.On "0" 'Aポートのモーターを回す
    Spirit1.Wait 2, 100 '1秒間待つ
    Spirit1.Off "0" 'Aポートのモーターを止める
    Spirit1.EndOfTask 'ダウンロードされるタスク終了
    Spirit1.PlaySystemSound 5 'ダウンロードの終了の通知
End Sub

Private Sub cmdRun_Click()
    Spirit1.SelectPrgrm 4 '実行するプログラム5を選ぶ
    Spirit1.StartTask 0 'プログラムを実行する
End Sub

Private Sub cmdStop_Click()
    Spirit1.Off "012" 'すべてのモーターを止める
    Spirit1.StopAllTasks 'すべてのタスクを止める
End Sub
```

図 12 PC からの制御プログラム

6. Visual Basic を用いたプログラム

Visual Basic を用いた簡単なプログラムを図 13 に示す。このプログラムはダウンロードすることなくパソコンから制御してロボットを動作させることができる。

パソコンから、ロボットの前進する、停止する、後進するという三つの動作を指示することができる。

Visual Basic を用いてプログラムを作成することができることで、応用範囲が拡大する。例えば、音声認識 OCX と連動させることで、音声コマンドによる制御が

```
Private Sub cmdBackward_Click()
    Spirit1.SetVar 15, 2, 1 '変数15に定数1をセットする
    Spirit1.SetPower "02", 0, 15 'モータ0,2に変数15の値をセットする
    Spirit1.SetRwd "02" 'モータ0,2を逆転に設定
    Spirit1.On "02" 'モータ0,2を回す
End Sub

Private Sub cmdForward_Click()
    Spirit1.SetVar 15, 2, 1 '変数15に定数1をセットする
    Spirit1.SetPower "02", 0, 15 'モータ0,2に変数15の値をセットする
    Spirit1.SetFwd "02" 'モータ0,2を正転に設定
    Spirit1.On "02" 'モータ0,2を回す
End Sub

Private Sub cmdStop_Click()
    Spirit1.Off "02" 'モータ0,2を止める
End Sub

Private Sub Form_Load()
    Spirit1.InitComm 'RCXとの通信の初期化
End Sub

Private Sub Form_Unload(Cancel As Integer)
    Spirit1.CloseComm 'RCXとの通信を終了
End Sub
```

図 13 Visual Basic によるプログラム例

```

Private Sub back_Click()
  Spirit1.SetVar 15, 2, 1
  Spirit1.SetPower "0", 0, 15
  Spirit1.SetVa 15, 2, 6
  Spirit1.SetPower "2", 0, 15
  Spirit1.SetRwd "02"
  Spirit1.On "02"
End Sub

Private Sub Dic_Click()
  vhdic = NecSre321.LoadDictionary
    ("C:\Temp\Sample\Gram\Mind.grm")
  NecSre321.ActivateDictionary (vhdic)
End Sub

Private Sub Form_Load()
  Spirit1.InitComm
End Sub

Private Sub Form_Unload(Cancel As Integer)
  Spirit1.CloseComm
End Sub

Private Sub forward_Click()
  Spirit1.SetVar 15, 2, 1
  Spirit1.SetPower "0", 0, 15
  Spirit1.SetVar 15, 2, 6
  Spirit1.SetPower "2", 0, 15
  Spirit1.SetFwd "02"
  Spirit1.On "02"
End Sub

Private Sub mic_Click()
  Screen.MousePointer = 11 '砂時計
  If NecSre321.Microphone = 1 Then
    NecSre321.Microphone = 0
    mic.Caption = "Mic OFF"
  Else
    NecSre321.Microphone = 1
    mic.Caption = "Mic ON"
  End If
  Screen.MousePointer = 0 '通常
End Sub

Private Sub NecSre321_srRecognized(ByVal hresult
  As Long, ByVal ncandidates As Integer)
  Dim Str1 As String
  If hresult = 0 Then Exit Sub
  Text1.Text = ""
  num = NecSre321.ResultGetNum(hresult, 0)
  For i = 0 To num - 1
    vhword = NecSre321.ResultGetWord(hresult, 0, i)
    If vhword = 0 Then
      Text1.Text = "単語が取得できません!"
      Exit Sub
    End If
    NecSre321.GetWordExt vhword, Str1
    Text1.Text = Text1.Text + " " + Str1
  Next i
  If Str1 = "前進" Then
    forward_Click
  Exit Sub
  End If
  If Str1 = "後進" Then
    back_Click
  Exit Sub
  End If
  If Str1 = "止まれ" Then
    stop_Click
  Exit Sub
  End If
End Sub

Private Sub stop_Click()
  Spirit1.Off "02"
End Sub

```

図 14 音声コマンドを用いたロボット制御

可能になる。図 14 に示す例は音声コマンドでロボットを遠隔制御するものである。音声で、前進、止まれ、行進と発声することでロボットの動作を音声で指示できる。すなわち、音声コマンドで遠隔からロボットの動作を制御することが可能になる。ここでは、音声認識自身は外部のパソコンで行っているが、RCX の CPU の処理能力が向上すれば、ロボット自身で認識するようになれるであろう。ここでは音声認識エンジンは NEC 製のものを使用した。

7. 教育コース

LEGO Mindstorms には教育用のガイドモードが準備されており、RIS の教育コースが備えられている。RCX の使い方、IR タワーの使用法、RCX によるプログラミングの仕方などが用意されている。また、簡単な例題プログラムも作成されている。これによって、独自で学習していくことができる。

8. おわりに

LEGO Mindstorms は各種の多用なブロックが用意されており、非常に有用な玩具ロボット教材といえるが、LEGO Mindstorms をエンターテイメントの道具としてみたときには、音声入出力パーツや画像を扱うミニカメラパーツなどのパーツがあると人間とのインタフェースが増え、応用範囲が広がると思われる。また、LEGO の自立性を高めるには、プログラムの環境として、Java などの言語を使用できると一層有効になるとと思われる。これまで述べたように、LEGO Mindstorms は小型のコンピュータを内蔵することで、ロボットの行動を規定してプログラムとして作成し、ロボットに知恵を与えることができる。ロボットの行動はプログラムとして作成するので、プログラミングの能力を養成することができ、知的な玩具ロボットとして自ら知識を獲得して行く学習道具といえる。

◇ 参考文献 ◇

[Jin Sato] <http://www.mi-ra-i.com/JinSato/Mindstorms/> (Jin Sato 氏の「MindStorms 情報局」)
[MIND 99] LEGO MINDSTORMS パーフェクトガイド: (株) 翔泳社 (1999)
[MIND HP] <http://mindstorms.lego.com/> (LEGO Mindstorms のホームページ)
[MIT Media] <http://www.lcs.media.mit.edu/people/fredm/mindstorms/index.html> (LEGO Mindstorms Information)
[NEC] NEC Voice SDK: NEC

2001年3月14日 受理

著者紹介

山本 昌弘 (正会員)



1966年大阪大学工学部精密工学科卒業。同年、日本電気(株)に入社。同社C&C研究所、同パーソナルC&Cグループを経て1999年退社。現在、法政大学国際文化学部教授。計算法アーキテクチャ、プログラミング言語、マルチメディア応用、記号処理、知識処理に関する研究。情報処理学会、電子情報通信学会、知識処理学会各会員。