

## 第2章

# ロボティクス入門

自動化機械から知性を持った機械へ／プログラムとは？／コンパイルとリンクとは？／センサリー・モータ・コーディネーション／ロボットを学習させる

この章では、ロボティクスと呼ばれる研究分野の全体像を理解してもらうために、簡単なロボットを例にとり、その仕組みと中心的な課題について述べます。簡単とはいっても、ここに出てくるロボットは、たくさんのセンサやアクチュエータを持っていますので、一見複雑そうに見えますが、基本的には、本書で紹介するようなマイコンで制御する方式と同様の理屈で動いています。

## 2.1 自動化機械から知性を持った機械へ

ロボットを語る上で欠かせないのが、人工知能の存在です。皆さんがある種の機械を見て、それをロボットだと思うか、たんなる自動化機械だと思うか？ 一般的には、同じ動作の繰り返しを行う機械や、ある特定の仕事だけに専門化したり、限定された使い方しかできない機械は、自動化機械といい、ロボットとは区別されています。

これに対して、様々な環境に適応して、状況判断を行い、目的の仕事を達成してゆけるような自律性を持った機械をロボットと定義すると、わかりやすくなります。この「環境適応」「状況判断」「目的達成」が人工知能研究のめざすところであり、かつロボットのおもしろみの一つでもあります。どの程度、人工知能が優秀ならそこに知性を感じるかというのは、個人や状況によってまったく異なりますので一概にはいえませんが、多くの環境の状態を取り込み、それに応じて適切な行動がとれるロボットほど知性があるとみなすのは自然でしょう(図2.1)。

「適切な行動を作り出す」ことが人工知能の役割であり、優秀な人工知能と優秀な機械とが合体したものがロボットとなるわけです。また、そのようなロボットをどうしたら作り出せるかを研究する学問領域がロボティクスです。

## 2.2 プログラムとは？

プログラムは、コンピュータの行動規則をコンピュータ用の言語で書き直したものです。プログラムの詳細については、第5章で説明しますが、ここでは、LED実験3のプログラム `test3.c` (リスト2.1)を例にとり、その基本的な流れについて解説します。

`test3.c`は、本書で使う基本的な機能をすべてもっています。基本的な機能とは、以下の(a)(b)(c)の三つです。

## リスト2.1 test3.c

```
#include <3664f.h>
void init_serial(void);
unsigned char receive(void);
void send(unsigned char ch);

void main(void){
    unsigned char ch;
    int x0;
    IO.PMR1.BYTE = 0x00;    /* ポート1を汎用I/Oに設定 */
    IO.PCR1 = 0xff;        /* ポート1を出力に設定 */
    IO.PDR1.BYTE=0xff;

    AD.CSR.BIT.ADST=0;
    AD.CSR.BIT.SCAN=0;
    AD.CSR.BIT.CKS=1;
    AD.CSR.BIT.CH=0;

    IO.PDR1.BIT.B7=0;
    IO.PDR1.BIT.B6=0;
    IO.PDR1.BIT.B5=0;
    IO.PDR1.BIT.B4=0;

    init_serial ();        /* シリアル通信の設定 */
    send('H');
    send('e');
    send('l');
    send('l');
    send('o');
    send(' ');
    send('W');
    send('o');
    send('r');
    send('l');
    send('d');

    while(1){
        if (SCI3.SSR.BIT.RDRF == 1) {
            ch=0;
            ch=receive();
            if(ch=='1'){
            }
            else if(ch=='1'){
                IO.PDR1.BIT.B4=1;
            }
            else if(ch=='2'){
                IO.PDR1.BIT.B5=1;
            }
            else if(ch=='3'){
                IO.PDR1.BIT.B6=1;
            }
            else if(ch=='4'){
                IO.PDR1.BIT.B4=1;
                IO.PDR1.BIT.B5=1;
                IO.PDR1.BIT.B6=1;
            }
        }
    }
}
```

```

    }
    else if(ch=='0'){
        IO.PDR1.BIT.B4=0;
        IO.PDR1.BIT.B5=0;
        IO.PDR1.BIT.B6=0;
    }
}

AD.CSR.BIT.ADST=1;
while(AD.CSR.BIT.ADF==0){};
AD.CSR.BIT.ADF=0;
AD.CSR.BIT.ADST=0;
x0=(int)(AD.DRA>>8);
if(x0<128){
    IO.PDR1.BIT.B7=0;
}
else {
    IO.PDR1.BIT.B7=1;
}
}
}

void init_serial(void){
    int i;
    IO.PMR1.BIT.TXD = 1 ;    /* P22をTXD機能に設定          */
    SCI3.SCR3.BYTE = 0;
    SCI3.SMR.BYTE = 0;
    /* 調歩同期式, データ長8ビット, 偶数パリティ, 1ストップ・ビット */
    SCI3.BRR = 8;           /* ボーレート設定 57600bps          */
    for(i=0;i<100;i++){    /* 処理待ち時間                      */
        SCI3.SCR3.BIT.RIE = 0; /* 受信割り込み禁止                  */
        SCI3.SCR3.BIT.TE = 1; /* 送信可能                          */
        SCI3.SCR3.BIT.RE = 1; /* 受信可能                          */
        SCI3.SSR.BYTE &= 0x80; /* ステータス・レジスタの初期化     */
    }
}

unsigned char receive (void) {
    unsigned char ch;
    if (SCI3.SSR.BIT.RDRF == 1) {
        ch = SCI3.RDR;      /* データ取り出し */
        SCI3.SSR.BIT.RDRF = 0; /* 受信フラグ・クリア */
    }
    else {
        ch = 0;
        SCI3.SSR.BYTE &= 0x80; /* エラー・フラグ・クリア */
    }
    return (ch);
}

void send (unsigned char ch){
    while (SCI3.SSR.BIT.TDRE == 0) ;
    SCI3.TDR = ch;
}

```

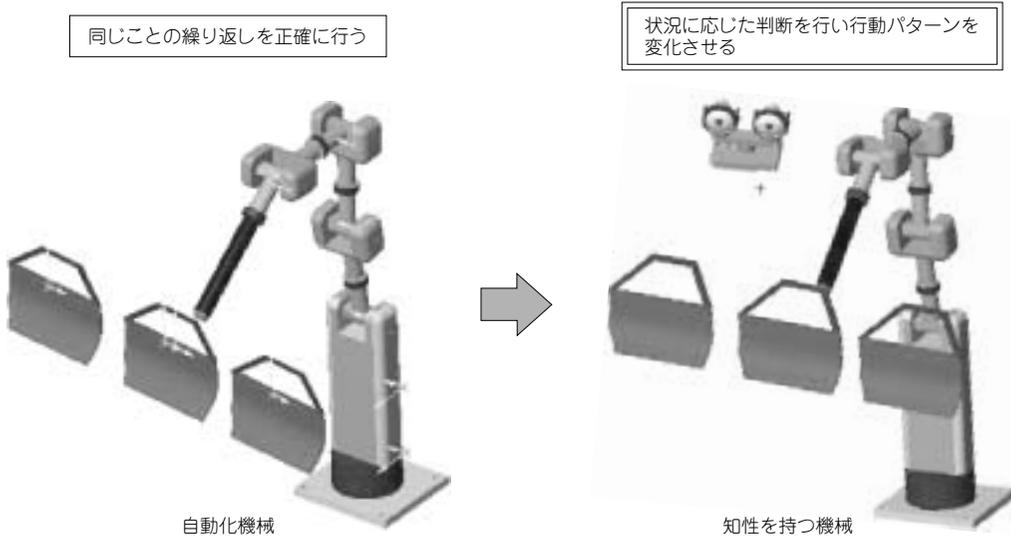


図2.1 自動化機械から知性を持った機械へ

- (a) I/Oアクセス：Port1の4番から7番までの入出力を制御
- (b) シリアル通信：パソコンとの通信を制御
- (c) A-Dコンバータ：外部入力(センサなど)からアナログ値を取得

これらの機能を順序よく並べたものがプログラムで、その流れの概略は、図2.2のフローチャートで示しました。フローチャートとは、プログラム内で行われる情報処理の流れにそって、その機能要素の関係を図的に表したもので、□は処理、◇は準備、◇は条件分岐、をそれぞれ意味しています。

test3.c(リスト2.1)はC言語で書かれています。この章ではH8Tinyを用いるために必要となる最低限のC言語の使い方について簡単にふれておきましょう。

C言語でプログラムを記述するとき最低限知っておかなければならないものは、

宣言、条件、関数、ポインタ

です。この他にもC言語の重要な機能はたくさんありますが、それらについては、巻末のAppendix 3を参照してください。

まず、test3.cのプログラムを見てみると、

#include <3664f.h>	ヘッダーの宣言
void init_serial (void);	サブルーチンの宣言
unsigned char receive (void);	サブルーチンの宣言
unsigned char receive_wait (void);	サブルーチンの宣言
void send (unsigned char ch);	サブルーチンの宣言
void main(void){	
unsigned char ch;	変数の宣言