

## 第5章 浮動小数点演算

MPC-2000 シリーズは浮動小数点演算にも対応していますが、一般の制御演算と区別するために、独立した FLOAT コマンドと以下のマクロコマンドの中でのみ、浮動小数点演算を実行します。また、MPC-2100 には、浮動小数点演算 IC が搭載されており、CPU に負担のかかる開平演算、三角関数を含んだ演算を高速に実行することができます。

### 5-1 浮動小数点演算マクロコマンド

マクロコマンドは MPC-684 との互換性のために移植された画像処理による位置補正に使用するコマンドです。AFFIN コマンドは 2 次元ベクトルを回転させることができるため、効率的に座標補正を行うことができます。詳細は、コマンドリファレンスを参照してください。

AFFIN	点データを回転変換
ATAN	ATAN で角度を得る
ATAN2	ATAN で角度を得る (汎用)
COS	COS 演算
SIN	SIN 演算
TAN	TAN 演算
GETDG	2 つの点を結ぶ直線と X 軸の成す角

### 5-2 浮動小数点演算

#### 倍精度配列変数 (FP(n))

FP(n) は特殊な配列変数です。あらかじめ、FP(0) ~ FP(7) の 8 個用意されており、浮動小数点演算対応変数として用いることができます。以下に例を示します。FP(n) に代入する式を記述すると、その式は、浮動小数点演算式となり、FP(n) には通常の変数と異なり浮動小数点形式でデータが保存されます。

#### ■ 数値を直接 E 形式で表示

```
#FP(1)=10/3
#pr FP(1)
3.333333E+00
#
```

#### ■ 10000 倍の上、整数化して表示

```
#FP(1)=10/3
#pr FP(10000,1)
33333
#
```

#### ■ ネイピア数の算出

```
10 FLOAT FP(2)=1
15 a=1
20 FOR i=1 TO 100
30 a=a*i
40 FLOAT FP(2)=FP(2)+1/a
50 NEXT
70 FORMAT "0.0000"
80 PRINT STR$(FP(10000,2))
#run

2.7182
#
```

#### ■ 2 次方程式 $x^2+4x+3=0$ の解

```
10 a=1 : b=4 : c=3
20 FLOAT FP(0)=(SQR(b*b-(4*a*c))-b)/2/a
30 FLOAT FP(1)=(SQR(b*b-(4*a*c))*-1-b)/2/a
40 PRINT FP(10000,0) FP(10000,1)
#run

-10000 -30000
#
```

## FLOAT コマンド

MPC-2000 シリーズの演算は通常、整数です。この演算と区別するために、FLOAT コマンドが用意されています。代入変数に FP(n) を指定すると、演算式には自動的に FLOAT コマンドが付加されます。なお、FLOAT コマンド中の演算は、通常の算術式と同様、加減算に対して \*、/ の乗除算が優先されます。

例:

FLOAT コマンドが付加される例	乗算が優先される例
<pre>#10 fp(2)=1/3 list 10   FLOAT  FP(2)=1/3 #</pre>	<pre>10   FLOAT  a=SQR(3*3+4*4) 20   PRINT  a #run  5 #</pre>

代入変数が通常の整数変数の場合で、FLOAT コマンドを付加すると、結果は整数として代入されますが内部演算は浮動小数点演算となります。これに伴い、開平関数 SQR() も FLOAT コマンド中では浮動小数点演算となります。

```
10   FLOAT  a=SQR(3)*10000000
20   PRINT  a
#run

17320508
#
```

## 浮動小数点对応関数

FLOAT コマンド中で使用できる算術関数は以下のとおりです。これらの関数は FLAOT コマンド中では、倍精度の浮動小数点関数として扱われます。通常の整数演算式中的ふるまいとは異なります。

SQR	開平計算	
SQ	自乗計算	
SIN	三角関数 SIN	入力はラジアン
COS	三角関数 COS	入力はラジアン
TAN	三角関数 TAN	入力はラジアン
ATAN	三角関数 ATAN	出力はラジアン
ACOS	三角関数 ACOS	出力はラジアン
RAD	度からラジアンに変換	出力はラジアン
DEG	ラジアンから度に変換	出力は度
VAL	文字列を浮動小数点として取得	

以下に任意の角度整数 i での SIN,COS の平方和が 1 になることを示す演算を例示します。SIN,COS の引数はラジアンであるために、100 度という整数値を関数 RAD() でラジアンに変換します。

```
10   FLOAT  a=SQR(SQ(SIN(RAD(i)))+SQ(COS(RAD(i))))*1000000
20   PRINT  a
#i=100
#run

1000000
#
```

さらに、ATAN を用いて $\pi$ を求めてみます。  
TAN(45 度)=1 であることから、ATAN(1) はラジアンとしては $\pi / 4$  の値となります。

```
10  FLOAT  FP(0)=ATAN(1)*4
20  PRINT  FP(10000,0)
#run

31415
#
```

なお、 $\pi$ の値そのものは、以下のように得るのが近道です。

```
10  FLOAT  FP(5)=RAD(180)
20  PRINT  FP(5)
#run

3.141593E+00
#
```

実際の応用では、外部機器からの EXP 表記の浮動小数点数を取り込む場合があります。これには VAL 関数を用います。

VAL 関数は、FLOAT コマンド中で、浮動小数点関数となり、 $\pm X.XXXXXX(e)YYY$  というタイプの文字列を読み取ることができます。

```
10  a$="C41$=Mx+9.7042e+002 C42$=My-6.3210e+002 "
20  FLOAT  a=VAL(a$) FP(0)=VAL(0) b=VAL(0) FP(1)=VAL(0)
30  PRINT  a FP(0) b FP(1)
#run

41 9.704200E+02 42 -6.321000E+02
#
```

FP(n) の文字列への変換は、FP\$(n) を用います。

以下に、FP\$(n) の表現とと整数化した固定小数点の様式を例示します。

```
10  FLOAT  FP(5)=RAD(180)
20  PRINT  FP$(5)
25  FLOAT  A=FP(5)*1000000
30  FORMAT "0.000000"
40  PRINT  STR$(A)
#run

3.141593E+00
3.141592
#
```

## 速度比較と使用上の注意

MPC-2000 シリーズでは、浮動小数点演算に対応していますが、MPC-1000、2000 ではソフトエミュレーション、2100 では、コプロセッサによって行うために実行速度と使用方法に差があります。

通常に加減乗除算では速度は同等となりますが、浮動小数点对応関数を含む演算を実行すると、MPC-2100 が圧倒的に高速となります。

下の例では、1 度から 180 度の SIN と COS の平方和を計算し、1 になることを確認し、速度を計測します。このプログラムを実行すると、MPC- 2 1 0 0 では 43msec、MPC-2000 では、148msec となります。

```
10  SYSCLK=0
20  FOR  i_=1 TO 180
30  FLOAT  a_ =SQR(SQ(SIN(RAD(i_)))+SQ(COS(RAD(i_))))
40  IF  a_!=1 THEN :PRINT "FL_NG_":END :END_IF
50  NEXT
60  PRINT  SYSCLK
```

MPC-2100 では浮動小数点对応関数で、演算 IC を使用するため、実際には、マルチタスク下で同時に演算を行うことができません。しかしながら、内部的にセマフォ処理を儲けることによって、処理衝突を回避し見かけ上マルチタスクの処理を可能としています。

なお、浮動小数点加減乗除算については、MPC-2100 でも演算 IC を使用せず、ソフトエミュレーションによって行われるためタスク間で同時に行っても不具合も非効率も生じません。