

# NCプログラムの基礎

# はじめに

このレポートは、Webサイト「NCプログラム基礎知識」の内容を加筆・修正等を行い、PDF版として作成したものです。

Webサイトは現在（2015年時点）でも更新を続けておりますので、そちらも合わせてご覧いただければと思います。

レポートの不備などがございましたら、下記メールアドレスかWebサイトからお願い致します。ご意見・ご感想などもお聞かせ頂けるとありがたいです。

メール：info2@s-projects.net

サイト：<https://nc-program.s-projects.net/>

## 【著作権について】

このレポートは著作権法で保護されている著作物です。下記の点にご注意戴きご利用下さい。

- ・ このレポートの著作権は作成者に属します。
- ・ 著作権者の許可なく、このレポートの全部又は一部をいかなる手段においても複製、転載、流用、転売等することを禁じます。
- ・ 著作権等違反の行為を行った時、その他不法行為に該当する行為を行った時は、関係法規に基づき損害賠償請求を行う等、民事・刑事を問わず法的手段による解決を行う場合があります。
- ・ このレポートに書かれた情報は、作成時点での著者の見解等です。著者は事前許可を得ずに誤りの訂正、情報の最新化、見解の変更等を行う権利を有します。
- ・ このレポートの作成には万全を期しておりますが、万一誤り、不正確な情報等がありましても、著者・パートナー等の業務提携者は、一切の責任を負わないことをご了承願います。
- ・ このレポートを利用することにより生じたいかなる結果につきましても、著者・パートナー等の業務提携者は、一切の責任を負わないことをご了承願います。

# 目次

## 第1部 基礎編

<b>第1章 NCプログラムとは</b>	<b>1</b>
制御装置	1
NCを使用するメリット	1
<b>第2章 NCプログラムの構成要素</b>	<b>3</b>
NCプログラムの全体像	3
座標系	7
演算子	8
数値の扱い方	10
変数	11
Mコード	14
Gコード	16
Gコード固定サイクル	21
工具径補正	26
関数	30
<b>第3章 制御構造</b>	<b>31</b>
無条件分岐   GOTO	31
条件分岐   IF	32
繰り返し   WHILE	33
サブプログラム	35
カスタムマクロ	36

## 第2部 実践編

<b>第4章 プログラミング</b>	<b>39</b>
プログラムの開始と終わり方	39
ヘリカル加工プログラムを作ってみる	41
変数を使用してみる	42
繰り返して変数の数値を変えてみる	44

変数で工具軌道を制御する	47
変数で穴位置を変える	49
マクロにして使い回す	50

<b>第5章 周辺機器</b>	<b>55</b>
-----------------	-----------

パソコンでのプログラミング	55
NCプログラムの入出力	57
テープ長さとバイトの関係	58

<b>第6章 NCプログラマの為の数学</b>	<b>59</b>
-------------------------	-----------

三角関数	59
三平方の定理	60

<b>第7章 NC旋盤</b>	<b>61</b>
-----------------	-----------

NC旋盤の特徴と機能	61
他の設備とのプログラミングの違い	62
刃先R補正	64
Gコード固定サイクル	65
ねじ切り	67

<b>第8章 OSP（オークマ）の特徴</b>	<b>68</b>
-------------------------	-----------

ファナックとの主な違い	68
プログラムの種類	70
スケジュールプログラム	72
サブプログラム呼び出し	73
変数	76

## 第3部 附録

<b>JIS6315-2 : 2003</b>	<b>77</b>
-------------------------	-----------

規定されているGコード	77
規定されているMコード	78

<b>カスタムマクロA</b>	<b>79</b>
-----------------	-----------

# 第1部 基礎編

## 第1章 NCプログラムとは

---

NCプログラムの「NC」とは数値制御（Numerical Control）の略称で、数値情報で指令することで工作機械などを制御する方式です。

このNCとコンピュータを組み合わせたものはコンピュータ数値制御（CNC）と呼ばれます。ただし、近年ではCNCではないNC工作機械というのはほとんどないので、単に「NC」「NC工作機械」という場合には、「CNC」あるいは「CNC工作機械」のことを指します。また、本書でもそのように扱います。

NCプログラムとは、NC工作機械を動作させる（制御する）ためのプログラムということになります。

### 制御装置

---

NCプログラムを使用するためには、それを読み込むためのシステムが必要になります。このシステムは「数値制御装置」あるいは、単に「制御装置」と呼ばれ、これがついていない工作機械ではNCプログラムを使うことができません。

制御装置は、工作機械に限定すると日本に本社があるファナックが世界トップのシェアを持っています。この他にも代表的なものとして、OSP（オークマ）、メルダス（三菱電機）、ヤスナック（安川電機）、マザトロール（ヤマザキマザック）、トスナック（東芝機械）、NECなどがあります。

本書ではファナックのプログラム形式で解説していますが、プログラミングの考え方は他の制御装置でも同じですので、ここで得られた知識はそのまま使うことができます。なお、OSP（オークマ）に関しては第8章でファナックとの違いなどを解説しています。

### NCを使用するメリット

---

NCはもともと、人がハンドルやレバーなどを操作して加工していたものを、自動化する目的で使用されるようになりました。工作機械に自動で加工を行わせることで、様々なメリットが得られるからです。

### 生産性の向上

---

加工が自動で行われるということは、人がその機械につききりになる必要がありません。すると、機械が加工を行っている間に、次に加工を行う製品の治工具やプログラムの準備などの外段取りを済ませることができます。

また、加工中に他の工作機械の加工準備から起動までを行えるので、一人で複数の機械を同時に動かすこともできます。

## 品質の安定

---

手動操作で加工を行うと、どうしても人によって面状態や寸法に違いが出てしまいます。これを自動で行うと同じ条件で加工ができるので、人による仕上がりの違いが出なくなり品質を安定させることができます。これは同じ製品を同じ品質で作る量産加工を行うのに適しており、自動で加工を行うので、人の操作ミスによる不具合の低減にもつながります。

## 管理が容易

---

自動で加工を行うということは加工速度を同じにできるので、自ずと加工時間も決まります。すると、生産計画が立てやすくなる上に、原価の管理もやりやすくなります。

## 安全性の向上

---

手動操作では加工を行っているすぐ側にいなければなりませんので、キリコや油などが飛んで来ることもあります。最近のNC工作機械には必ず扉が付いているのでこのような危険は少なく、作業環境が油まみれになるといったことも防げます。

## 複雑な加工が容易になる

---

NCを使用する最大のメリットは、手動操作ではできない複雑な加工を実現できることです。2軸や3軸以上の軸を同時に制御できるので、斜めの形状やR加工なども簡単に行えて、ワークのセット方向を変えたり傾けたりする必要もありません。

## 第2章 NCプログラムの構成要素

---

### NCプログラムの全体像

---

まずはNCプログラムの全体的な構成について見ていきましょう。下記のプログラムを見て下さい。（サンプルですので意味のあるプログラムではありません）

% ;	…データスタート
O1000 (SAMPLE) ;	…プログラム番号 （括弧内注釈）
#100 = 10. ;	…変数
N100 ;	…シーケンス番号
S2000 ;	…主軸回転数
M3 ;	…Mコード
G01 X50. Y-50. Z-10. F500 ;	…Gコード 座標軸 送り速度
M5 ;	
M02 ;	
% ;	…データエンド

上から順に見ていきましょう。

#### % | データスタート/データエンド

---

プログラムの先頭にある「%」はデータスタートを表す記号で、ここからプログラムが始まりますよ、という意味があります。プログラムの最後にも同じ「%」がありますが、こちらはデータエンドを表す記号で、プログラムはここまでですよ、という意味になります。

つまり最初に読み込んだ「%」はプログラムの始まり、2つ目に読み込んだ「%」はプログラムの終わりになり、これが一つのプログラムとなります。各設備にプログラムを読み込ませる場合に必要になりますので、パソコンでプログラムを作る際には必ず付けましょう。

※設備によっては「M02」「M30」「M99」などでプログラムの終了と判断し、次のブロックから別のプログラムとして読み込むものもあります。古い設備にこのタイプが多いようです。

#### ; | EOB（エンドオブブロック）

---

上のプログラムを見ると、すべての行の終わりに「;」（セミコロン）が付いています。この「;」をEOB（イー・オー・ビー）と呼びます。End Of Block（エンドオブブロック）の頭文字です。行の終わりには必ず付けることになっており、行の先頭から「;」までをブロックと呼びます。

ただし、パソコンなどでプログラムを作成する場合は、設備に読み込ませるときに改行が「;」に変換されるので、つける必要はありません。

## O | プログラム番号

---

データスタートの次のブロックに来るのがプログラム番号です。「O」の後に続く数字がプログラム番号で、ファナックの多くは最大で4桁になりますが、1桁～3桁で入力することも可能です（O0012 → O12 等）。

プログラム番号は1つのプログラムに1つ、データスタートのすぐ後のブロックに記述する必要があります。

## () | 注釈（コメント）

---

「O1000」の後に「(SAMPLE)」と入っていますが、「()」の中身はNCでは読み込まれません。自分の好きなように注釈を入れることができます。

あまり使いすぎるのも考え物ですが、少なくともこのプログラムは何を行うものなのか、という注釈は入れておいた方が良いでしょう。自分で作ったプログラムでも、時間が経てば何に使うプログラムなのか分からなくなってしまうこともあります。

「()」の中は読み込まれませんので基本的には何を記述しても良いのですが、全角文字や一部の記号は使用できません。NCには全角文字などを表示できる機能はついていないので、設備に読み込ませたときに別の文字に変換されます。こうなると何が記述されているのか分からなくなってしまうです。

設備の操作パネルのキーに無い文字は、使用できないと考えた方が良いでしょう。

## # | 変数

---

3ブロック目に「#100=10.」とありますが、ファナックなどの制御装置では「#」+「数値」で変数を表します。変数については後ほど詳しく説明しますが、数値を保存する箱だと思っておいてください。

ここでは変数の後に「=10.」とありますが、NCで使用する「=」は等しいという意味ではなく、代入を意味します。つまり「#100」という箱に「10.」という数値を入れて保存しておき、後でこの「10.」という数値を取り出して使用します。

## N | シーケンス番号

---

4ブロック目にある「N100」はシーケンス番号と呼ばれ、「N」+「数値」で表します。プログラムを見やすくするための目印で、単独で記述しても動作に影響はありません。「GOTO」の飛び先としても使用されます。

「N」の後に続く数値は自分の好きな数値を入れることができます。数値の番号も順番に入れなければならないわけではなく、同じ番号を付けることも可能ですが、上のブロックから順番に数値が大きくなっていくように番号を付けた方が、より見やすくなります。



## アドレス + 数値 → ワード

---

5～9ブロックにはアルファベットと数値が並んでいます。NCプログラムは「%」や「#」などの特殊な場合を除くと、ほとんどが「アルファベット」+「数値」の形になっています。アルファベットの部分をアドレスと呼び、アドレスと数値を合わせてワードと呼びます。

アドレスの後の数値には変数や計算式を使用することも可能です。計算式の場合は式全体を[ ]で括る必要があります。

NCプログラムではアルファベットと数値の組み合わせによって、様々な機能を実現しています。

## S | 主軸回転数

---

5ブロック目にある「S2000」は主軸の回転数を設定するもので、「S」+「数値」で表します。回転数の単位はrpm(revolution/minute = 一分間で何回転するか)です。「S2000」だけで主軸が回転するわけではありません。

## Mコード | 補助機能

---

6ブロック目にある「M3」の「M」はMコードと呼ばれ、補助機能を行うものです。主軸の回転や停止などを行います。

Mコードについては、別ページで詳しく解説しています。

## Gコード | 準備機能

---

7ブロック目の先頭にある「G01」の「G」はGコードと呼ばれ、準備機能を行うものです。直線補間や円弧補間などを行います。

Gコードについては、別ページで詳しく解説しています。

## X,Y,Z | 座標軸

---

7ブロック目の真ん中にある「X50. Y-50. Z-10.」は、それぞれX軸、Y軸、Z軸のことで、後に続く数値は座標、又は距離になります。

## F | 送り速度

---

7ブロック目の最後にある「F500」は送り速度を設定するもので、「F」+「数値」で表します。送り速度の単位はmm/min(一分間に何ミリ移動するか)です。ただし、旋盤などの場合は初期設定でmm/rev(主軸が一回転する間に何ミリ移動するか)になっています。これらはGコードで変更可能です。

## T | 次工具選択

---

冒頭のプログラムには記述されていませんが、ATC（オートツールチェンジャ）を装備した設備であれば、「T」+「数値」でその数値のツール番号に入っている工具を呼び出すことができます。呼び出しただけでは工具交換はされず、「M06」を記述することで工具交換が行われます。

## NCプログラムの全体像 まとめ

---

- NCプログラムの先頭と末尾には「%」（データスタート/データエンド）を入れる。
- データスタートのすぐ後のブロックには、プログラム番号を記述しなければならない。
- 「()」内は読み込まれないので注釈として使用できるが、設備の操作パネルにない文字は使用しない。
- NCプログラムは複数のブロックからできており、1つのブロックはいくつかのワードによって構成されている。
- ワードはアドレスと数値に分かれており、このアドレスと数値の組み合わせによって様々な機能を実現している。

NCプログラムは特定の記述がない限り、上から下に向かって1ブロックずつ読み込んでいきます。NCにはブロックを先読みする機能が付いており、設備にもよりますが、現在のブロックから3～5ブロックを先読みしています。プログラムエラーでアラームが出た場合は、先読みされたブロックにエラーがある場合もありますので、なかなかエラーが見つからないときにはこのことを思い出してみてください。

## 本書の注意事項

---

本書では、読みやすくするためにワード間にスペースを入れています。ファナックなどの制御装置ではスペースを入れる必要はありません。設備側で自動的にワードを認識して分割されます。

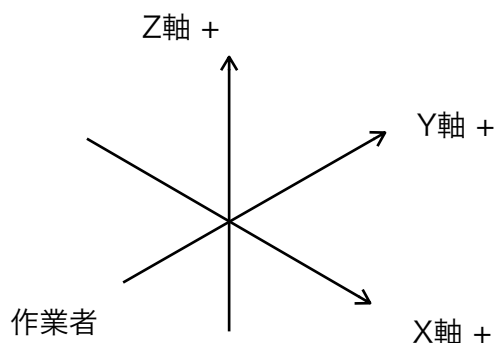
また、この章の冒頭のプログラムではEOB「;」を記載しましたが、それ以外は本書ではEOBを省略しています。

## 座標系

---

NCで使用される座標系は、X軸、Y軸、Z軸の3軸の直交座標系です。これに回転軸を加えた4軸、または5軸を制御できる設備もありますが、ここでは分かりやすく3軸での解説をします。

各軸の方向は設備の種類によって異なりますが、フライス盤や立形マシニングセンタでは、X軸は横方向で右が+、Y軸は前後方向で奥が+、Z軸は上下方向で上が+、となります。



また、円弧補間などの回転方向や工具径補正の方向を見る場合には、平面に垂直な軸のプラス側から見たときに、時計回りなのか反時計回りなのか、あるいは左なのか右なのかを判断します。XY平面（G17）であれば垂直な軸はZ軸になり、Z軸のプラス側は上になりますから、上から見たときの方向ということになります。

NC設備の座標には主に機械座標、ワーク座標（絶対座標）、相対座標の3つの座標があります。

### 機械座標

---

機械座標は設備自体が持っている固有の座標で、変更することはできません。フライス盤や立形マシニングセンタの多くは、右奥が機械原点となりX座標、Y座標は0になります。X軸はテーブルの中心が機械原点になっている設備もあります。Z軸は主軸が一番上に上がったところが機械原点になります。

### ワーク座標（絶対座標）

---

ワーク座標は設定したワーク座標系（G54～59）を原点とした座標です。いわゆる加工原点になります。

ワーク座標を使用するには、加工前にあらかじめワーク座標系に機械座標を入力しておきます。そしてプログラムの中で、どのワーク座標系を使用するのかを設定すると、ワーク座標系に入力されている機械座標がワーク座標の原点（X0, Y0）となります。

### 相対座標

---

オペレーターが自由に変更できる座標です。確認用に使うことが多いようです。

## 演算子

---

NCプログラムでは数学と同じように、四則演算（+,-,x,÷）や比較（<,<=,>）、論理演算（AND,OR）などを行うことができます。数学とは少し異なった記号を使用する場合もあるので、間違えないように注意しましょう。

### 算術演算子

---

NCコード	意味
+	加算（+）
-	減算（-）
*	乗算（×）
/	除算（÷）
=	代入

他のページでも書きましたが、NCプログラムでは「=」は等しいという意味ではなく代入です。「#100 = 10.」のように代入は変数に行い、それ以外には使用しません。

四則演算では「#100 = 10. + 5. \* 10.」となっていれば、右辺側の計算結果が#100に代入されます。計算される順番は数学と同じで基本的には左側から計算され、乗算、除算がある場合は、加算、減算よりも先に計算されます。この場合は「60.」が「#100」に代入されます。

また、「[ ]」を使用することで計算が行われる順番を変えることもできます。「#100 = [10. + 5.] \* 10.」とした場合は、まず「[ ]」の中の「10. + 5.」が先に計算されてから、「\* 10.」の計算が行われます。この場合「#100」には「150.」が代入されます。

四則演算は変数への代入だけでなく、アドレスの後に続く数値にも使用することができます。この場合は「X[10. + 20.] Y[5. - 2.]」のように、計算式を「[ ]」で括る必要があります。

除算を行う時の注意点として「#100 = 0 / 1」では「#100」には「0」が代入されます。「0 / 1」は「0」になりますから、当然と言えば当然なのですが、逆に「#100 = 1 / 0」は、「0で除算されました」というアラームが出て計算できません。これは「1 / 0」は「0」にはならないからです。変数を使用する場合にも、割る数が「0」になるとアラームが出ますので注意が必要です。

## 比較演算子

---

NCコード	意味
EQ	左辺と右辺が等しい (=)
NE	左辺と右辺が等しくない (≠)
LT	左辺が右辺より小さい (<)
LE	左辺が右辺以下である (≤)
GT	左辺が右辺より大きい (>)
GE	左辺が右辺以上である (≥)

比較演算子は主に条件分岐 (IF) や繰り返し (WHILE) などで使用されます。

「IF [#100 EQ 1] GOTO200」のように、比較演算を行っている部分は[ ]で括る必要があります。

「IF [[#100+1] EQ #1] GOTO300」のように、計算式を入れることもできます。この場合も計算式を[ ]で括る必要があります。

## 論理演算子

---

NCコード	意味
OR	どちらか一方でも成立すれば真
XOR	両方が成立しなければ真
AND	両方が成立すれば真