

# TB-DECO,

## Funktionsweise des Rechners

Der in Ausgabe 23 erschienene Artikel beschrieb die Möglichkeit zur Programmierung einer Teileparametrierung mithilfe der erweiterten Programmierfunktionen. Die optimale Nutzung dieser Programmierfunktionen setzt allerdings das Verständnis der Funktionsweise des Rechners voraus.



Die verwendete Programmiersprache umfasst eine Reihe von Funktionen, die im Kapitel "Erweiterte Programmierung" der HILFE ZU TB-DECO beschrieben werden.

Die Mehrzahl dieser Funktionen wird nicht benötigt, wenn lediglich einfache mathematische Operationen in einem Ablauf realisiert werden sollen.

Daher werden im Folgenden nur diese beiden Funktionen behandelt:

**@Move**  
**@Key**



### Rechner

Der im System TB-DECO enthaltene Rechner verwendet eine Methode namens „inverse logische Verkettung“, die sich dadurch auszeichnet, dass in der Transkription mathematischer Ausdrücke von links nach rechts keine Klammern verwendet werden. Stattdessen übernimmt ein separates Modul die Verwaltung aller Klammern.

### Beschreibung dieser Methode

Einige unter Ihnen, insbesondere Benutzer von Rechnern (Taschenrechnern) des Herstellers Hewlett Packard, sind möglicherweise bereits mit inverser Verkettung vertraut. Bei einem „normalen“

Rechner wird die Rechenoperation "3 \* 4" beispielsweise wie folgt eingegeben:

$$3 * 4 =$$

Bei einer Maschine mit inverser Verkettung lautet die Eingabe hingegen wie folgt:

**3 EINGABE 4 \***

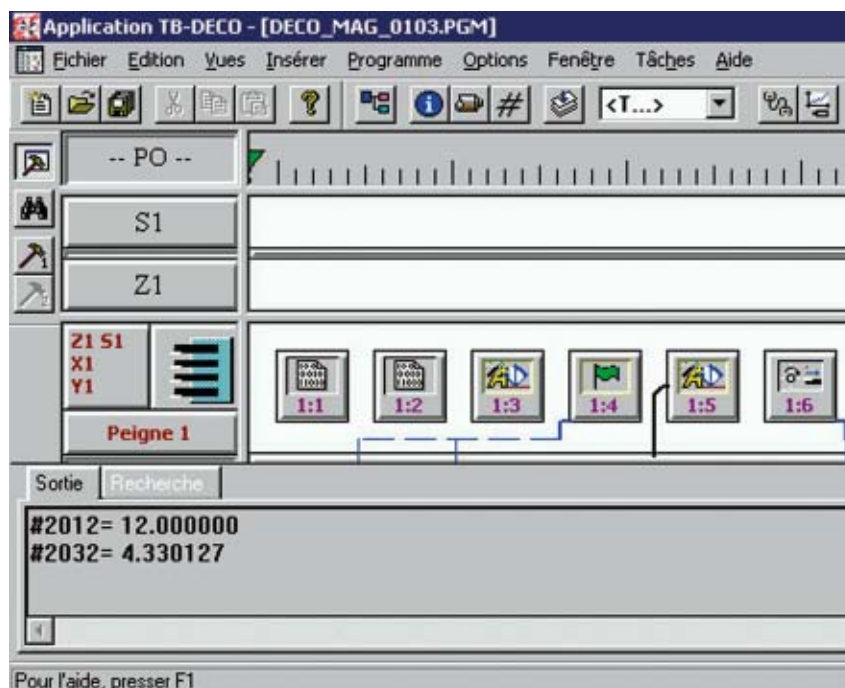
Durch Drücken der Tasten 3 und EINGABE, wird die Zahl 3 im Register des Rechners abgelegt. Anschließend wird die Zahl 4 gefolgt vom Operator \* eingegeben. Der Rechner greift nun auf die Zahl im Register (3) und auf die aktuelle Zahl (4) zu, multipliziert diese und gibt das Ergebnis (12) aus. Auch wenn diese Methode die Eingabe von Gleichungen verlängert, bietet sie den Vorteil, dass keine Klammern eingegeben werden müssen und dass die Berechnung schneller erfolgt. Der Grund hierfür ist Folgender: Angenommen, das Ergebnis der Rechnung  $(4+5)*3$  soll ermittelt werden. Die hierzu erforderliche Eingabe lautet bei einem normalen Rechner:

$$(4 + 5) * 3 =$$

Zum Berechnen des Ergebnisses müssen also acht Tasten gedrückt werden. Bei inverser Verkettung lautet die Eingabe wie folgt:

**4 EINGABE 5 + 3 \***

Es werden also zwei Tasten weniger benötigt, um zum Ergebnis zu gelangen.



### TB-DECO – Besonderheiten

Der Rechner des Systems TB-DECO umfasst mehrere Register, beginnend mit dem Register DX. Die Funktion EINGABE, mit deren Hilfe eingegebene Zahlen in den Registern abgelegt werden, wird durch die Funktion @Move ersetzt.

#### Funktion @Move

@Move ermöglicht das Ablegen eines Werts an einem definierten Ziel. Beispiele:

a) Register des Rechners (DX)	@Move	DX	8
b) Ausgabefenster (OUT)	@Move	OUT	#2012
c) Variable	@Move	#2012	10
d) Indirekter Wert	@Move	#2012	#2013
e) Übergabe an das Register DX	@Move	#2013	DX

#### Syntax

@Move <Ziel> <Quelle>

Das <Ziel> ist die Variable, in die der Wert eingegeben wird. Die <Quelle> ist entweder ein direkter Wert, der in die Variable <Ziel> eingegeben wird, (c) der Wert einer Variable, der in die Variable <Ziel> kopiert wird oder (d) der Inhalt des Registers DX, der an eine Variable (e) übergeben wird.

#### Details

@Move	DX	8	Speichert den Wert 8 in DX
@Move	OUT	#2012	Zeigt den Wert der Variable #2012 im Ausgabefenster an
@Move	#2012	10	Speichert den Wert 10 in #2012
@Move	#2012	#2013	Kopiert den Wert von #2013 nach #2012
@Move	#2013	DX	Übergibt den Wert in DX an eine Variable

#### Funktion @Key

Die Funktion @Key ermöglicht das Anwenden aller mathematischen Operationen auf die Register des Rechners.

#### Syntax

@Key <Operator>

Zulässig sind die Operatoren +, -, \*, / bzw. die trigonometrischen Operatoren TAN, SIN oder COS. Eine vollständige Liste aller verfügbaren Operatoren finden Sie in der HILFE zum System TB-DECO.

# TB-DECO,

## Funktionsweise des Rechners

**Beispiele:**

Führen Sie die nachfolgende Berechnung  $3 * 4$  aus, und übergeben Sie das Ergebnis an die Variable #2012.

Codezeile	Registerstatus				
@Move DX 3	<table border="1"><tr><td></td></tr><tr><td></td></tr><tr><td>3</td></tr></table> DX			3	Speichern des Werts 3 im Register DX
3					
@Move DX 4	<table border="1"><tr><td>3</td></tr><tr><td>4</td></tr></table> DX	3	4	Der Wert 3 wird in das nächste Register verschoben Speichern des Werts 4 im Register DX	
3					
4					
@Key *	<table border="1"><tr><td></td></tr><tr><td></td></tr><tr><td>12</td></tr></table> DX			12	@Key führt die Multiplikation beider Register aus Das Ergebnis der Operation wird in DX abgelegt
12					
@Move #2012 DX	<table border="1"><tr><td></td></tr><tr><td></td></tr><tr><td>12</td></tr></table> DX			12	Der Wert in DX wird an die Variable #2012 übergeben
12					

Führen Sie die nachfolgende Berechnung  $(\#3048 / 2) / \text{TAN } 30^\circ$  aus, und übergeben Sie das Ergebnis an die Variable #2032.  
Hinweis: #3048 = 5

Codezeile	Registerstatus				
@Move DX #3048	<table border="1"><tr><td></td></tr><tr><td></td></tr><tr><td>5</td></tr></table> DX			5	Speichern des Werts von #3048 (5) in DX
5					
@Move DX 2	<table border="1"><tr><td>5</td></tr><tr><td>2</td></tr></table> DX	5	2	Der Wert 5 wird in einem anderen Register abgelegt Speichern des Werts 2 in DX	
5					
2					
@Key /	<table border="1"><tr><td></td></tr><tr><td></td></tr><tr><td>2.5</td></tr></table> DX			2.5	@Key führt die Division beider Register aus Das Ergebnis wird in DX abgelegt
2.5					
@Move DX 30	<table border="1"><tr><td>2.5</td></tr><tr><td>30</td></tr></table> DX	2.5	30	Der Wert 2.5 wird in das folgende Register verschoben Speichern des Werts 30 in DX	
2.5					
30					
@Key TAN	<table border="1"><tr><td>2.5</td></tr><tr><td>0.577</td></tr></table> DX	2.5	0.577	@Key TAN berechnet den Tangens von 30 Das Ergebnis wird in DX abgelegt	
2.5					
0.577					
@Key /	<table border="1"><tr><td></td></tr><tr><td></td></tr><tr><td>4.330</td></tr></table> DX			4.330	@Key führt die Division beider Register aus Das Ergebnis wird in DX abgelegt
4.330					
@Move #2032 DX	<table border="1"><tr><td></td></tr><tr><td></td></tr><tr><td>4.330</td></tr></table> DX			4.330	Übergabe des Werts in DX an die Variable #2032
4.330					

**Wichtiger Hinweis:**

Sollen alle erweiterten Funktionen in einer Operation programmiert werden, muss vor dem Funktionsaufruf eine geöffnete eckige Klammer hinzugefügt werden.

Beispiel: [ @Move ..., [ @Key .... etc