Inhaltsverzeichnis

xCAM G-Code 1
- Datentypen 1
- Programmkopf (H) 1
- Kommentar (COM) 2
- Arbeitsseite (F) 2
- Kontur Startpunkt (G0) 4
- Linie (G1) 5
- Bogen (G5) 5
- Fräsung (GIN) 6
- Tasche (PO) 8
- Schnitt (CUT) 10
- Bohrung (B) 12
- Nut (GROOVE) 13
- Falz (RABBET) 14
- Universal Makro (UNIMACRO) 15

xCAM G-Code

Der G-Code ist das Dateiformat, in welchem xCAM die Informationen speichert. Jeder G-Code speichert die Informationen in einer Zeile, welche mit CR LF abgeschlossen wird. Die Datei hat die Endung **.NCC**. Bei Zweitbearbeitungen, z.B. Bearbeitungen von oben und unten, muss, sofern die Maschine nicht eigens dafür ausgelegt ist, eine Datei für die Bearbeitung oben und eine für unten ausgegeben werden. Prinzipiell können beliebig viele Dateien für ein Bauteil ausgegeben werden.

Jede Zeile muss mit dem Befehlaufruf des jeweiligen G-Codes beginnen. Dieser ist bei jedem Befehl in der Überschrift in Klammern vermerkt.

Soll der G-Code für ein Nesting in xPress verwendet werden, muss jedes Teil eine geschlossene Kontur enthalten. Diese Kontur darf keinerlei Zuweisungen von Fräsungen, Taschen oder Schnitten haben und muss als Konturtyp eine Nestingkontur sein.

Eine Beispieldatei können Sie hier herunterladen, hier als Grafik. Um die erzeugten Dateien grafisch darstellen zu können, benötigen Sie eine Installation von xCAM. Kontaktieren Sie uns hierfür bitte per Email.

Datentypen

Der G-Code kann folgende Datentypen verarbeiten:

- **string:**
  - Alle strings (Textvariablen) müssen in Anführungszeichen angegeben werden. Zeilenumbrüche in Werten von Variablen, z.B. bei Beschreibungen, werden mit der Zeichenfolge **_C_N_L_** dargestellt.
- **double:**
  - Zahlen haben als Dezimaltrennzeichen den Punkt und werden ohne Tausendertrennzeichen definiert.
- **bool:**
  - Bei Variablen des Typs bool wird 0 für false und 1 für true verwendet.

Jede Variable muss, falls angegeben, einen Wert zugewiesen haben. Der Standardwert für Texte ist ein leerer String (**„„**) und für Zahlen eine **0**.

Programmkopf (H)

Der Programmkopf, auch **Header** genannt, definiert das Fertigmaß des Bauteils und muss zwingend an erster Stelle angeführt werden.

Variablen

Variable	Beschreibung	Datentyp	Pflichtfeld
DX	Länge des Teils.	double	x
DY	Breite des Teils.	double	x
DZ	Dicke des Teils	double	x
OX	Definiert den Versatz des Teiles in X.	double	
OY	Definiert den Versatz des Teiles in Y.	double	
OZ	Definiert den Versatz des Teiles in Z.	double	
H_COUNT	Menge gleicher Bauteile	int	
PART_NAME	Name des Bauteils	string	
PART_NO	Teilenummer	string	
FIELD	Gibt das Arbeitsfeld der Maschine an. Dieser Wert definiert sich aus dem Arbeitsfeld, wie dieses in xCAM bei der Maschine definiert worden ist.	string	x
C_OPT_W_SIDES	Gibt an, ob die Arbeitsseiten optimiert werden sollen. Ist diese Option deaktiviert, werden alle Bearbeitungen in ein CNC-Programm an die Maschine übergeben. Andernfalls werden laut Maschineneinstellungen in xCAM mehrere CNC-Programme erzeugt. (z.B. ein CNC-Programm für die Bearbeitungen oben und ein zweites für die Bearbeitungen von unten.	bool	
TEXT	Freie Beschreibung	string	

Kommentar (COM)

Der Kommentar hat keine Auswirkung auf das fertige CNC-Programm. Dieser kann verwendet werden, um Informationen im G-Code festzuhalten.

Variablen

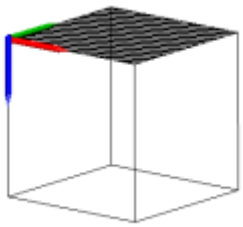
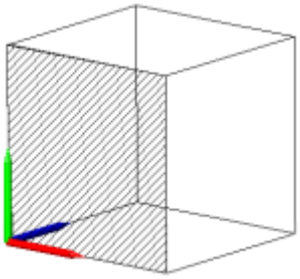
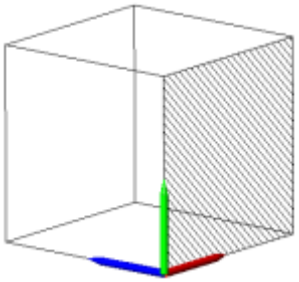
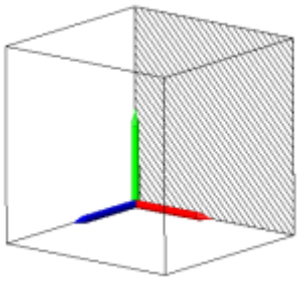
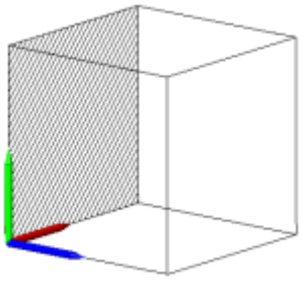
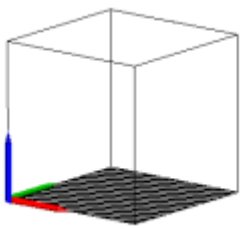
Variable	Beschreibung	Datentyp	Pflichtfeld
MSGSTATUS	Gibt den Typ des Kommentars an. Aktuell hat dieser immer den Wert 0.	int	x
TEXT	Freie Beschreibung	string	x

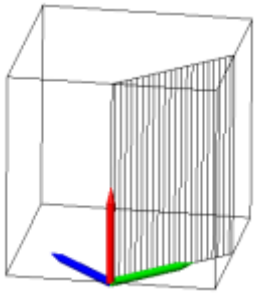
Arbeitsseite (F)

Die Arbeitsseite definiert die Ebene, auf welcher sich alle nachfolgenden Befehle beziehen. Die Achsen des UCS sind folgendermaßen definiert:

- Rote Achse: X
- Grüne Achse: Y
- Blaue Achse: Z

xCAM kann folgende Arbeitsseiten verwalten:

Arbeitsseite	Beschreibung	Nullpunkt
1	Arbeitsseite oben	
2	Arbeitsseite vorne	
3	Arbeitsseite rechts	
4	Arbeitsseite hinten	
5	Arbeitsseite links	
6	Arbeitsseite unten	

Arbeitsseite	Beschreibung	Nullpunkt
0	Diese Arbeitsseite definiert eine freie Ebene.	

Variablen

Variable	Beschreibung	Datentyp	Pflichtfeld
F	Gibt die Arbeitsseite an.	int	x
POSX	Gibt den Ursprung der Arbeitsseite in X an.	double	nur bei Arbeitsseite F0
POSY	Gibt den Ursprung der Arbeitsseite in Y an.	double	nur bei Arbeitsseite F0
POSZ	Gibt den Ursprung der Arbeitsseite in Z an.	double	nur bei Arbeitsseite F0
ANGLEZ	Gibt die Drehung in Grad in um die Z-Achse an.	double	nur bei Arbeitsseite F0
ANGLEX	Gibt die Drehung in Grad um die **neue** X-Achse an.	double	nur bei Arbeitsseite F0
ANGLEZ2	Gibt die Drehung in Grad um die **neue** Z-Achse an.	double	nur bei Arbeitsseite F0
TEXT	Freie Beschreibung	string	

Kontur Startpunkt (G0)

Jede Kontur beginnt mit einem G0. Konturen werden bei verschiedenen Technologiedaten benötigt:

- Fräsung
- Tasche (nur bei Freiformtaschen)
- Schnitt (nicht zwingend)
- Als Nestingkontur

Variablen

Variable	Beschreibung	Datentyp	Pflichtfeld
X	Startpunkt der Kontur in X.	double	x
Y	Startpunkt der Kontur in Y.	double	x
Z	Startpunkt der Kontur in Z.	double	x
N	Name der Kontur, welcher innerhalb eines CNC-Programmes eindeutig sein muss. Über diesen Namen geschieht die Zuweisung an die Technologiedaten.	string	x
CTYPE	Gibt den Typ der Kontur an. Folgende Typen können definiert werden:	int	x
	0 = Standardkontur.		
	1 = Hilfskontur. Diese Kontur wird nur für visuelle Darstellungen benötigt und hat keine Auswirkung auf das CNC-Programm.		
	2 = Nestingkontur. Anhand der Nestingkontur werden die Bauteile bei einem eventuellen Nesting geschachtelt.		
ANGA	Drehwinkel in Grad um die Z-Achse.	double	

Variable	Beschreibung	Datentyp	Pflichtfeld
ANGB	Drehwinkel in Grad um die neue X-Achse	double	
TEXT	Freie Beschreibung	string	

Linie (G1)

Der Code **G1** definiert eine Linie. Als Startpunkt dient der Endpunkt des vorherigen Codes. Dies kann ein **G0, ein G1 oder ein G5** sein.

Variablen

Variable	Beschreibung	Datentyp	Pflichtfeld
X	Endpunkt der Linie in X.	double	x
Y	Endpunkt der Linie in Y.	double	x
Z	Endpunkt der Linie in Z.	double	x
G1_SPEED	Vorschubgeschwindigkeit.	double	
G1_ANGLE_C	Drehung in Grad um die Z-Achse. Dieser Winkel wird nur benötigt, wenn sich der Winkel im Laufe der Kontur ändern soll. Standardmäßig wird der Winkel im GIN definiert.	double	
G1_ANGLE_A	Drehung in Grad um die neue X-Achse. Dieser Winkel wird nur benötigt, wenn sich der Winkel im Laufe der Kontur ändern soll. Standardmäßig wird der Winkel im GIN definiert.	double	
TEXT	Freie Beschreibung	string	

Bogen (G5)

Der Code **G5** definiert einen Bogen. Als Startpunkt dient der Endpunkt des vorherigen Codes. Dies kann ein **G0, ein G1 oder ein G5** sein.

Variablen

Variable	Beschreibung	Datentyp	Pflichtfeld
X	Endpunkt des Bogens in X.	double	x
Y	Endpunkt des Bogens in Y.	double	x
Z	Endpunkt des Bogens in Z.	double	x
I	Zentrum des Bogens in X.	double	x
J	Zentrum des Bogens in Y.	double	x
DIR	Drehrichtung des Bogens. -1 = Bogen gegen den Uhrzeigersinn 1 = Bogen im den Uhrzeigersinn	int	x
G5_SPEED	Vorschubgeschwindigkeit.	double	
G5_ANGLE_C	Drehung in Grad um die Z-Achse. Dieser Winkel wird nur benötigt, wenn sich der Winkel im Laufe der Kontur ändern soll. Standardmäßig wird der Winkel im GIN definiert.	double	

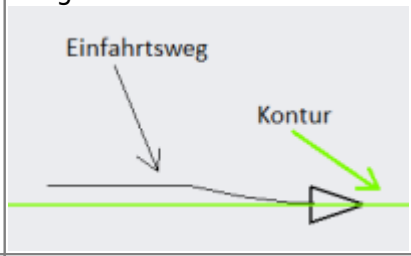
Variable	Beschreibung	Datentyp	Pflichtfeld
G5_ANGLE_A	Drehung in Grad um die neue X-Achse. Dieser Winkel wird nur benötigt, wenn sich der Winkel im Laufe der Kontur ändern soll. Standardmäßig wird der Winkel im GIN definiert.	double	
TEXT	Freie Beschreibung	string	

Fräsung (GIN)

Eine Fräsung erzeugt eine Bearbeitung im Werkstück, welche entlang der zugewiesenen Kontur verläuft.

Variablen

Variable	Beschreibung	Datentyp	Pflichtfeld
T	Gibt das Werkzeug an, mit welchem diese Bearbeitung abgearbeitet werden soll. Dieses Werkzeug muss mit dem Eintrag in der Werkzeugdatenbank von xCAM übereinstimmen.	string	x
TI	Definiert die Tiefe der Bearbeitung.	double	Wenn der Paramter **Tiefe aus Kontur (DFC)** den Wert **1** hat, ist dieser Wert irrelevant. Ansonsten ist dies ein Pflichtfeld.
N	Gibt den Namen der Kontur an, entlang welcher die Bearbeitung verläuft.	string	x
CS	Definiert das Startsegment, an welchem die Bearbeitung beginnen soll. Dieser Wert ist standardmäßig mit **0** zu füllen, um die gesamte Kontur zu bearbeiten.	string	x
CE	Definiert das Endsegment, an welchem die Bearbeitung enden soll. Dieser Wert ist standardmäßig mit **0** zu füllen, um die gesamte Kontur zu bearbeiten.	string	x
C	Definiert die Werkzeugkorrektur, d.h. auf welcher Seite der Kontur der Fräser läuft.	int	x
	0 = Keine Werkzeugkorrektur. Der Fräser läuft auf dem Mittelpunkt der Kontur.		
	1 = Werkzeugkorrektur rechts. Der Fräser wird um den Fräserradius nach rechts versetzt.		
	2 = Werkzeugkorrektur links. Der Fräser wird um den Fräserradius nach links versetzt.		

Variable	Beschreibung	Datentyp	Pflichtfeld
OF	Zusätzliches Offset zur Werkzeugkorrektur. Ist keine Werkzeugkorrektur eingestellt (**C=0**) ist dieser Wert irrelevant.	double	
GIN_DRIVE_IN_SPEED	Eintauchgeschwindigkeit	double	
GIN_DRIVE_SPEED	Bearbeitungsgeschwindigkeit	double	
GIN_ROTATION_SPEED	Drehzahl	double	
IN	Gibt den Einfahrtsmodus der Bearbeitung an. 0 = Auf Punkt: Der Fräser sticht direkt am Konturstartpunkt in das Werkstück ein. 1 = Gerade: Verlängert die Kontur am Startpunkt tangential zum ersten Segment. Die Länge des Einfahrtsweges berechnet sich aus dem **Fräserradius * Faktor Einstichart**. Zusätzlich kann der Einfahrtsweg am Beginn um den Wert **Abstand Einfahrtsweg** versetzt werden. 2 = Bogen: Verlängert die Kontur am Startpunkt tangential zum ersten Segment. Der Radius des Bogens berechnet sich aus dem **Fräserradius * Faktor Einstichart**. 3 = Ausschnitt: Dieser Einfahrtsweg eignet sich nur bei geschlossenen Innenkonturen. Dabei wird ein Einfahrtsweg berechnet, der sich um den Wert **Abstand Einfahrtsweg** nach innen versetzt und eine Länge aus **Fräserradius * Faktor Einstichart** hat. 	int	x
INFACTOR	Definiert den Faktor Einstichart. Dieser wird für die Berechnung des Einfahrtsweges benötigt.	double	
INCUTOUTDIST	Definiert den Abstand Einfahrtsweg. Dieser wird für die Berechnung des Einfahrtsweges benötigt.	double	
INFLYING	Gibt an, ob der Einfahrtsweg fliegend sein soll. Dabei startet der Fräser am Beginn des Einfahrtsweges oberhalb der Platte und dringt entlang des Einfahrtsweges auf die richtige Tiefe ein.	bool	
OUT	Gibt den Ausfahrtsmodus der Bearbeitung an. 0 = Auf Punkt: Der Fräser endet direkt am Konturende. 1 = Gerade: Verlängert die Kontur am Ende tangential zum letzen Segment. Die Länge des uUsfahrtsweges berechnet sich aus dem **Fräserradius * Faktor Ausstichart**. Zusätzlich kann der Ausfahrtsweg am Ende um den Wert **Abstand Ausfahrtsweg** versetzt werden. 2 = Bogen: Verlängert die Kontur am Ende tangential zum letzen Segment. Der Radius des Bogens berechnet sich aus dem **Fräserradius * Faktor Ausstichart**.	int	x

Variable	Beschreibung	Datentyp	Pflichtfeld
OUTFACTOR	Definiert den Faktor Ausstichart. Dieser wird für die Berechnung des Ausfahrtsweges benötigt.	double	
OUTCUTOUTDIST	Definiert den Abstand Ausfahrtsweg. Dieser wird für die Berechnung des Ausfahrtsweges benötigt.	double	
OVERLAY	Gibt die Überlappung an, um welche die Kontur am Endpunkt verlängert wird. Dieser Wert ist nur bei geschlossenen Konturen relevant.	double	
HPOS	Gibt die Haubenposition an. Es können die Werte 0 - 4 vergeben werden. Diese werden im Post-Prozessor dann für die Maschine übersetzt.	int	
ANGLE_Z	Gibt die Neigung des Werkzeuges an.	double	
ANGLE_XY	Gibt die Drehung des Werkzeuges um die Z-Achse an. Dieser Wert ist nur relevant, wenn die Variable **GIN_CALC_Z_ANGLE** den Wert **0** hat.	double	
VECTOR_MODE	Definiert den Modus, wie sich die C-Achse (Drehung um Z) verhalten soll.	int	
	0 = Fixer Winkel. Der Fräser wird am Beginn mit dem angegebenen Winkel zugestellt und ändert sich nicht.		
	1 = C-Achse mitdrehen. Der Fräser dreht sich relativ zum jeweiligen Kontursegment.		
VECTOR_CALC_MODE	Gibt an, welche Ebene in Z verwendet wird, um die Neigung des Werkzeuges zu berechnen.	int	
	0 = Oberkante. Als Z-Wert wird die Dicke der Platte verwendet.		
	1 = Unterkante. Als Z-Wert wird 0 verwendet.		
	2 = Z-Wert der Kontur. Als Z-Wert wird jener verwendet, welcher in der zugewiesenen Kontur eingestellt ist.		
PRIO	Definiert die Priorität der Fräsung. Je kleiner die Zahl der Priorität ist, desto früher wird diese abgearbeitet. Eine Fräsung mit Priorität 1 wird vor einer Fräsung mit Priorität 10 gemacht. Der Standardwert ist 1.	int	
TEXT	Freie Beschreibung	string	

Tasche (PO)

Eine Tasche setzt eine geschlossene Kontur voraus, sofern es sich nicht um eine Kreis- oder Rechtecktasche handelt.

Variablen

Variable	Beschreibung	Datentyp	Pflichtfeld
TYPE	Definiert den Typ der Tasche 0 = Kreistasche. Dieser Typ erfordert den Einfügepunkt und Durchmesser der zu fräsenden Tasche. Der Einfügepunkt bezieht sich auf das Zentrum der Tasche. 1 = Rechtecktasche. Dieser Typ erfordert den Einfügepunkt sowie Länge und Breite der zu fräsenden Tasche. Der Einfügepunkt bezieht sich auf die untere linke Ecke. 2 = Freiformtasche. Dieser Typ erfordert eine geschlossene Kontur.	int	x
X	Definiert den Einfügepunkt der Tasche in X.	double	Nur bei Taschen des Typs **TYPE=0** oder **TYPE=1**.
Y	Definiert den Einfügepunkt der Tasche in Y.	double	Nur bei Taschen des Typs **TYPE=0** oder **TYPE=1**.
Z	Definiert den Einfügepunkt der Tasche in Z.	double	Nur bei Taschen des Typs **TYPE=0** oder **TYPE=1**.
DIA	Gibt den Druchmesser einer Kreistasche an.	double	Nur wenn der **TYPE=0** ist.
LEN	Gibt die Länge einer Rechtecktasche an.	double	Nur wenn der **TYPE=1** ist.
WID	Gibt die Breite einer Rechtecktasche an.	double	Nur wenn der **TYPE=1** ist.
ANGRAD	Gibt den Radius einer Rechtecktasche (TYPE=1) an, mit welchem die Ecken abgerundet werden sollen	double	
ANG	Gibt den Winkel an, um welchen eine Rechtecktasche (TYPE=1) auf der XY-Ebene gedreht ist.	double	
T	Definiert das Werkzeug, mit welchem die Tasche ausgefräst werden soll	string	x
TI	Gibt die Tiefe der Tasche an.	double	Diese Variable ist nur dann kein Pflichtfeld, wenn der Wert **TICON=1** und **TYPE=2** ist.
TICON	Gibt an, dass die Tiefe der Tasche dem Z-Wert der Kontur entspricht. Dieser Wert ist nur bei einer Freiformtasche (**TYPE=2**) relevant.	bool	
CNAME	Gibt den Namen der Kontur an, welche als Tasche ausgefräst werden soll.	string	Wenn der **TYPE=2** ist.
OFF	Gibt ein Offset der Tasche an. Um diesen Wert wird die Tasche verkleinert.	double	
CLEAR	Gibt den Modus an, wie die Tasche ausgeräumt werden soll. 0 = Spiral 1 = Zig-Zag	int	x
INTOOUT	Gibt an, ob die Tasche von innen nach außen ausgeräumt werden soll oder nicht.	bool	
ROT_SPEED	Definiert die Drehzahl des Werkzeuges.	double	

Variable	Beschreibung	Datentyp	Pflichtfeld
IN_SPEED	Definiert die Eintauchgeschwindigkeit.	double	
SPEED	Definiert die Vorschubgeschwindigkeit.	double	
CLEANMODE	Bietet die Möglichkeit, die Tasche nach dem Ausräumen zusätzlich mit einem Werkzeug zu bearbeiten. 0 = Ohne Nachfräsen. 1 = Rand Schlichten. Umfräst die Außenkontur der Tasche mit dem Werkzeug, das unter **CLEANTOOL** angegeben wurde. Ist dieser Modus aktiv, hat man die Möglichkeit, direkt unter der Tasche einen Befehl **Fräsung (GIN)** zu definieren. Die Tiefe wird automatisch aus der Tasche übernommen. Ebenso ist die Werkzeugkorrektur irrelevant. Dieser Befehl die nur dazu, die Einfahrtswege dieser Zusaätzfräsung zu definieren. Dieser Befehl heißt allerdings nicht **GIN** sondern **POCKET_CLEANTOOL_GIN**.	int	
CLEANTOOL	Gibt das Werkzeug an, mit welchem die Zusatzfräsung erzeugt werden soll.	string	Nur wenn **CLEANMODE=1** ist.
P_PRIO	Definiert die Priorität der Tasche. Je höher die Priorität ist, desto früher wird diese abgearbeitet. Der Standardwert ist 1.		
TEXT	Freie Beschreibung	string	

Schnitt (CUT)

Definiert einen Schnitt anhand angegebener Koordinaten oder einer zugewiesenen Kontur.

Variablen

Variable	Beschreibung	Datentyp	Pflichtfeld
T	Gibt das Werkzeug an, mit welchem der Schnitt ausgeführt werden soll.	string	x
TI	Definiert die Tiefe des Schnittes	double	x
CR	Gibt die Anzahl der Arbeitsschritte an, in welchen die gesamte Tiefe erreicht werden soll.	int	
CALCMODE	Gibt an, nach welchem Modus der Schnitt berechnet werden soll 0 = Koordinaten. Der Schnitt definiert sich anhand des angegebenen Start- und Endpunktes. 1 = Kontur. Der Schnitt berechnet sich anhand der angegebenen Kontur.	int	x
C	Definiert die Werkzeugkorrektur, d.h. auf welcher Seite der Kontur die Säge läuft. 0 = Werkzeugkorrektur links. Die Säge wird um die halbe Sägeblattdicke nach links versetzt. 1 = Keine Werkzeugkorrektur. Der Fräser läuft auf dem Mittelpunkt der Kontur. 2 = Werkzeugkorrektur rechts. Die Säge wird um die halbe Sägeblattdicke nach rechts versetzt.	int	x

Variable	Beschreibung	Datentyp	Pflichtfeld
CUT_START_X	Definiert den Startpunkt des Schnittes in X.	double	Nur wenn **CALCMODE=0**.
CUT_START_Y	Definiert den Startpunkt des Schnittes in Y.	double	Nur wenn **CALCMODE=0**.
CUT_START_Z	Definiert den Startpunkt des Schnittes in Z.	double	Nur wenn **CALCMODE=0**.
CUT_END_X	Definiert den Endpunkt des Schnittes in X.	double	Nur wenn **CALCMODE=0**.
CUT_END_Y	Definiert den Endpunkt des Schnittes in Y.	double	Nur wenn **CALCMODE=0**.
CUT_END_Z	Definiert den Endpunkt des Schnittes in Z.	double	Nur wenn **CALCMODE=0**.
N	Gibt den Namen der Kontur an, aus der sich der Schnitt berechnen soll.	string	Nur wenn **CALCMODE=1**.
CS	Definiert das Startsegment, an welchem die Bearbeitung beginnen soll. Dieser Wert ist standardmäßig mit 0 zu füllen, um die gesamte Kontur zu bearbeiten. Dieser Wert ist nur wichtig, wenn der Parameter **CALCMODE=1** ist.	string	x
CE	Definiert das Endsegment, an welchem die Bearbeitung beginnen soll. Dieser Wert ist standardmäßig mit 0 zu füllen, um die gesamte Kontur zu bearbeiten. Dieser Wert ist nur wichtig, wenn der Parameter **CALCMODE=1** ist.	string	x
CUTMODE	Gibt den Modus an, wie sich der Schnitt am Beginn und Ende verhält. 0 = Auf Punkt. Die Säge sticht mit ihrem Mittelpunkt direkt am Startpunkt des Schnittes ein und geht bis zum Endpunkt des Schnittes. 1 = Auf Länge. Dieser Modus sorgt dafür, dass die Säge nicht über den Start- und Endpunkt hinaus geht. Anhand der Tiefe des Schnittes und dem Sägeblattradius werden die genauen Koordinaten berechnet. 2 = Verlängerung über Maßangabe. Verlängert den Schnitt am Startpunkt um den Wert **ADD_LENGTH_START** und am Ende um den Wert **ADD_LENGTH_END**. 3 = Verlängerung über Faktor der Sehnenlänge. Verlängert den Schnitt am Startpunkt um den **Sägeblattradius * CUT_START_FACTOR** und am Ende um **Sägeblattradius * CUT_END_FACTOR**. 4 = Automatisch. Dieser Modus kommt nur dann zum Tragen, wenn **CALCMODE=1** definiert ist. In diesem Fall werden die einzelnen Schnitte anhand der angegebenen Kontur berechnet. Bei Innenecken wird der Schnitt automatisch verkürzt, bei Außenecken verlängert sich dieser um **Sägeblattradius * CUT_START_FACTOR**.	int	x

Variable	Beschreibung	Datentyp	Pflichtfeld
ADD_LENGTH_START	Gibt an, um wie viel der Schnitt am Startpunkt verlängert werden soll. Dieser Wert ist nur bei **CUTMODE=2** relevant.	double	
ADD_LENGTH_END	Gibt an, um wie viel der Schnitt am Endpunkt verlängert werden soll. Dieser Wert ist nur bei **CUTMODE=2** relevant.	double	
CUT_START_FACTOR	Gibt den Faktor an, um welchen sich, verrechnet mit dem Sägeblattraduis, der Schnitt am Startpunkt verlängern soll. Dieser Wert ist nur bei **CUTMODE=3** oder **CUTMODE=4** relevant.	double	
CUT_END_FACTOR	Gibt den Faktor an, um welchen sich, verrechnet mit dem Sägeblattraduis, der Schnitt am Endpunkt verlängern soll. Dieser Wert ist nur bei **CUTMODE=3** relevant.	double	
DO_SCRATCH	Gibt an, ob vor dem effektiven Schnitt vorgeritzt werden soll.	bool	
SCRATCH_DEPTH	Gibt die Vorritztiefe an.	double	
IS_GG	Gibt an, ob im Gegenlauf geschnitten werden soll.	bool	
CUT_ANGLE	Gibt den Schnittwinkel. Bei normalen, senkrechten Schnitten ist der Winkel = 0 zu definieren.	double	
CUT_ANGLE_MODE	Gibt den Referenzpunkt in Z an, von welchem sich ein schräger Schnitt berechnet.	int	Nur wenn ein schräger Schnitt ausgeführt werden soll.
	0 = Oberkante.		
	1 = Unterkante.		
HPOS	Gibt die Position der Haube an. Zulässig sind die Werte 0-4, welche im Post-Prozessor für die jeweilige Maschine übersetzt werden.	int	
CUT_DRIVE_IN_SPEED	Definiert die Eintauchgeschwindigkeit der Säge.	double	
CUT_DRIVE_SPEED	Definiert die Vorschubgeschwindigkeit der Säge.	double	
PRIO	Definiert die Priorität des Schnittes. Je höher die Priorität ist, desto früher wird dieser abgearbeitet. Der Standardwert ist 1.	int	
TEXT	Freie Beschreibung	string	

Bohrung (B)

Führt eine Bohrung aus.

Variablen

Variable	Beschreibung	Datentyp	Pflichtfeld
X	Startpunkt der Bohrung in X	double	x
Y	Startpunkt der Bohrung in Y	double	x
Z	Startpunkt der Bohrung in Z	double	x
DU	Gibt den Durchmesser der Bohrung an.	double	x

Variable	Beschreibung	Datentyp	Pflichtfeld
TI	Definierte die Tiefe der Bohrung.	double	Immer, außer **B_IS_TH=1**.
TOOL	Gibt das Werkzeug an, mit welchem die Bohrung erzeugt werden soll.	string	Immer, außer **B_MODE=0**
B_IS_TH	Gibt an, ob die Bohrung eine Durchgangsbohrung ist.	bool	
B_THV	Gibt an, um wie viel die Bohrung bei einer Durchgangsbohrung durchgebohrt werden soll.	double	
B_MODE	Gibt den Modus der Bohrung an. 0 = Durchmesser. Die Bohrung wird immer mit dem Durchmesser an die Maschine übergeben. 1 = Werkzeug. Die Bohrung wird immer mit dem angegebenen Werkzeug an die Maschine übergeben. 2 = Automatisch. Wird ein Bohrer mit dem angegebenen Durchmesser gefunden, wird die Bohrung mit Durchmesser ausgegeben. Andernfalls wird versucht, die Bohrung mit dem angegebenen Werkzeug zu erzeugen. 3 = Bohrung fräsen. Fräst die Bohrung als Spirale.	int	x
B_COUNT_HOLE	Gibt die Anzahl der Bohrungen an.	int	Dieser Wert ist nur relevant, wenn **B_RASTER_X** oder **B_RASTER_Y** größer als 0 sind.
B_RASTER_X	Gibt den Raster der Bohrung in X an.	double	
B_RASTER_Y	Gibt den Raster der Bohrung in Y an.	double	
AZ	Definiert den Winkel um die Z-Achse.	double	
AX	Definiert die Neigung der Bohrung (neue X-Achse).	double	
B_ROTATION	Gibt die Drehzahl der Bohrspindel an.	double	
B_WORKSPEED	Gibt die Vorschubgeschwindigkeit an.	double	
B_COUNT_STEPS	Gibt an, in wie vielen Arbeitsschritten die angegebene Tiefe erreicht werden soll.	int	
B_PRIORITY	Definiert die Priorität der Bohrung. Je höher die Priorität ist, desto früher wird diese abgearbeitet. Der Standardwert ist 1.	int	
TEXT	Freie Beschreibung	string	

Nut (GROOVE)

Definiert eine Nut. Ob die Nut gesägt oder gefräst wird, wird durch das angegebene Werkzeug definiert. Sollen die Daten an eine Nestingmaschine übergeben werden, ist dieser Befehl nicht zulässig. In diesem Fall muss die Nut mit dem Befehl GIN und der entsprechenden Kontur zum Ausräumen der Nut oder mit dem Befehl PO ausgegeben werden.

Variablen

Variable	Beschreibung	Datentyp	Pflichtfeld
SX	Definiert den Startpunkt der Nut in X.	double	x
SY	Definiert den Startpunkt der Nut in Y.	double	x
EX	Definiert den Endpunkt der Nut in X.	double	x
EY	Definiert den Endpunkt der Nut in Y.	double	x
WID	Gibt die Breite der Nut an.	double	x
DEP	Gibt die Tiefe der Nut an.	double	x
T	Definiert das Werkzeug welches verwendet werden soll, um die Nut auszuräumen.	string	x
TX	Gibt das Werkzeug an, welches explizit für die Nut verwendet werden soll, wenn diese in X-Richtung geht. Ist hier kein Werkzeug angegeben, wird das Hauptwerkzeug der Nut **T** verwendet.	string	
TY	Gibt das Werkzeug an, welches explizit für die Nut verwendet werden soll, wenn diese in Y-Richtung geht. Ist hier kein Werkzeug angegeben, wird das Hauptwerkzeug der Nut **T** verwendet.	string	
TCL	Gibt das Werkzeug an, das bei nicht durchgehenden Nuten verwendet wird, um die Ecken nachzufräsen.	string	
DIRECTION	Gibt an, ob die Nut um Gleichlauf oder im Gegenlauf geschnitten werden soll.	bool	
	0 = Nut wird im Gleichlauf geschnitten.		
	1 = Nut wird im Gegenlauf geschnitten.		
DRIVEINSPEED	Definiert die Eintauchgeschwindigkeit.	double	
DRIVESPEED	Definiert die Vorschubgeschwindigkeit.	double	
PRIORITY	Definiert die Priorität der Nut. Je höher die Priorität ist, desto früher wird diese abgearbeitet. Der Standardwert ist 1.	int	
TEXT	Freie Beschreibung	string	

Falz (RABBET)

Definiert einen Falz. Sollen die Daten an eine Nestingmaschine übergeben werden, ist dieser Befehl nicht zulässig. In diesem Fall muss der Falz mit dem Befehl GIN und der entsprechenden Kontur zum Ausräumen des Falzes oder mit dem Befehl PO ausgegeben werden.

Variablen

Variable	Beschreibung	Datentyp	Pflichtfeld
SX	Definiert den Startpunkt des Falzes in X.	double	x
SY	Definiert den Startpunkt des Falzes in Y.	double	x
EX	Definiert den Endpunkt des Falzes in X.	double	x
EY	Definiert den Endpunkt des Falzes in Y.	double	x
WID	Gibt die Breite des Falzes an.	double	x
DEP	Gibt die Tiefe des Falzes an.	double	x

Variable	Beschreibung	Datentyp	Pflichtfeld
T	Gibt das Werkzeug an, mit welchem der Falz gefräst werden soll.	string	x
TCL	Gibt das Werkzeug an, das verwendet werden soll, um bei einem nicht durchgehenden Falz die Ecken nachzufräsen.	string	
DRIVEINSPEED	Gibt die Eintauchgeschwindigkeit des Fräsers an.	double	
DRIVESPEED	Gibt die Vorschubgeschwindigkeit des Fräsers an.	double	
PRIORITY	Definiert die Priorität des Falzes. Je höher die Priorität ist, desto früher wird dieser abgearbeitet. Der Standardwert ist 1.	int	
TEXT	Freie Beschreibung	string	

Universal Makro (UNIMACRO)

Das Universal Makro bietet die Möglichkeit, fixe Komponenten zu definieren. Diese werden im Post-Prozessor den entsprechenden Anforderungen angepasst. Bezieht sich das Makro auf eine spezielle Arbeitsseite, ist diese vor dem Makro zu definieren.

Variablen

Variable	Beschreibung	Datentyp	Pflichtfeld
NAME	Gibt den Namen des Makros an, über welchen dieses im Post-Prozessor identifiziert werden kann.	string	x
POSX	Definiert den Einfügepunkt des Makros in X.	double	x
POSY	Definiert den Einfügepunkt des Makros in Y.	double	x
POSZ	Definiert den Einfügepunkt des Makros in Z.	double	
T	Gibt das Werkzeug an, welches für dieses Makro eventuell relevant ist.	string	
DEPTH	Definiert die Tiefe des Makros.	double	
ANGLEX	Definiert den Neigungswinkel des Makros.	double	
ANGLEZ	Definiert den Drehwinkel des Makros.	double	
VALUE1	Definiert eine freie Variable, die im Post-Prozessor ausgewertet werden kann.	double	
VALUE2	Definiert eine freie Variable, die im Post-Prozessor ausgewertet werden kann.	double	
VALUE3	Definiert eine freie Variable, die im Post-Prozessor ausgewertet werden kann.	double	
VALUE4	Definiert eine freie Variable, die im Post-Prozessor ausgewertet werden kann.	double	
VALUE5	Definiert eine freie Variable, die im Post-Prozessor ausgewertet werden kann.	double	
PRIORITY	Definiert die Priorität des Unimakros. Je höher die Priorität ist, desto früher wird dieses abgearbeitet. Der Standardwert ist 1.	int	
TEXT	Freie Beschreibung	string	

From:
https://wiki.camid.eu/ -

Permanent link:
https://wiki.camid.eu/doku.php?id=camid:cam:gcode

Last update: **12.10.2023 07:58**