G-Code Kurs

Wichtiger Hinweis!

Dieser G-Code-Kurs verwendet die Beamicon2-Software im Demo-Modus.

Eine echte Maschine darf man nicht anschließen!

Dies kann zu Schäden an der Maschine führen und die Maschine zerstören. Wenn Sie den Kurs mit einer angeschlossenen Maschine benutzen wollen, müssen Sie zuerst ein Konfigurations-Backup machen, da die Konfiguration vom Kurs überschrieben wird. Wenn Sie Fragen haben, lesen Sie das Benutzerhandbuch und/oder fragen Sie Ihren Händler.

Vorbereitungen

Zum Starten benötigen wir das Programm Beamicon2, das auf einem Computer installiert sein muss. Es kann unter Windows oder Linux Debian sein. Für die Installation der Beamicon2 Software lesen Sie bitte das Beamicon2 Benutzerhandbuch.

Wir müssen auch eine Konfigurationsdatei für den Kurs herunterladen:

(**http://tecno-world.com/Forum/cursilloGCode/foro_metal.ini)**

Die Installation ist einfach

Nach der Installation fragt das Programm nach der Sprache und dem Modell einer Maschine. Es spielt keine Rolle, welche wir wählen. Gehen Sie bei geöffnetem Programm auf Menü->Datei->Konfiguration importieren und importieren Sie die Konfigurationsdatei, die Sie zuvor heruntergeladen haben.

Der Name dieser Datei lautet forum_metall.ini

Wir haben jetzt eine virtuelle Maschine mit den folgenden Spezifikationen:

- - Typ: 3-Achsen-Fräsmaschine,
- -X: von 0mm - 500mm
- -Y: von 0mm - 300mm
- -Z: von 0mm - 200mm, (200mm ist oben, 0mm = der Tisch)

Da wir keine echte Maschine haben, können wir die Maschine nicht referenzieren, in unserem Fall sind nach dem Programmstart alle Achsen auf "0".

Wir brauchen nur die ersten Bildschirmseite "Programm", der Rest spielt momentan keine Rolle und wir können ihn (für den Kurs) vergessen.

Wir haben immer noch das Logo im mittleren Fenster oben. Auf dem Ziffernblock drücken wir nun die Taste "6" (Pfeil nach rechts), das Logo verschwindet und die Koordinaten zeigen in X etwas anderes als "0" an. Dies ist die manuelle Bewegung, damit wir unsere virtuelle Maschine bewegen können.

Mit einem Klick der linken Maustaste auf das Grafikfenster können wir die Grafik verschieben. Mit der rechten Maustaste können wir die Grafik drehen. Mit dem Mausrad können wir zoomen.

Die äußeren Linien zeigen den nutzbaren Raum unserer virtuellen Maschine. Das dreifarbige Pfeilsymbol zeigt den Nullpunkt der Maschine an.

Der Fräser befindet sich mit seiner Spitze an dem Punkt, der durch die Koordinaten im Koordinatenfenster rechts angezeigt wird. (Wenn wir den Fräser nicht sehen, liegt es daran, dass wir noch kein Werkzeug ausgewählt haben, wir werden ihn später noch sehen)

Jetzt können wir probieren die Maschine zu jedem beliebigen Punkt im Arbeitsraum zu fahren. Jetzt sind wir bereit, das erste Programm zu schreiben.

Beginn des Kurses

Erzeugen wir eine neue Datei: Menü->Datei->Neue NC-Datei erstellen...

Es öffnet sich ein Dialog, in dem wir den Namen einer Datei schreiben, zum Beispiel "mein_erster_Test.nc", und wir speichern diese.

Wir befinden uns nun bereits im Code-Editor. Die ersten Zeilen wurden vom System automatisch hinzugefügt.

Der G-Code-Editor und die Syntax

Wir spielen ein wenig mit dem Editor, um zu verstehen, wie er funktioniert. Die ersten beiden Zeilen in Grau sind ein Kommentar, der nichts bewirkt - er ist nur zur Dokumentation.In der dritten Zeile steht ein %-Symbol, dies zeigt an, dass das CNC-Programm dort beginnt. Alle Zeilen vor diesem Symbol werden als Kommentare behandelt. Dies kann sehr nützlich für die Dokumentation von Programmen und Unterprogrammen sein, z.B. wenn wir nach einem Jahr das gleiche Teil fertigen wollen, wir Daten kennen, z.B. welche Fräsmaschine wir verwendet haben, usw.

Gehen wir mit dem Cursor auf die Zeile unter dem %-Symbol. Beginnen wir, etwas zu schreiben:

```
G0 X100
```

Wenn wir den Befehl schreiben, ändert er seine Farbe, jeder Befehlstyp hat eine bestimmte Farbe. Dies erleichtert die Fehlersuche beim Schreiben des Programms.

Beispiel:
`G77 X20`

Die Farbe von `G77` ändert sich nicht - dies weist darauf hin, dass es sich nicht um einen korrekten Befehl handelt. Okay, probieren wir noch mehr:

```
G0 G43 M99 M3 X100.0 #33=5
```

Wir können bereits sehen, dass jedes spezifische Kommando seine Farbe hat. Zur Dokumentation können wir zu jeder Zeile in Klammern einen Kommentar schreiben: `G0 X100 (meine erste Bewegung)`

Okay, wir haben genug gespielt, und ich denke, es ist jedem mehr oder weniger klar, wie es funktioniert. Um das Fenster des Editors zu schliessen, drücken wir nun auf die Taste "Abbrechen".

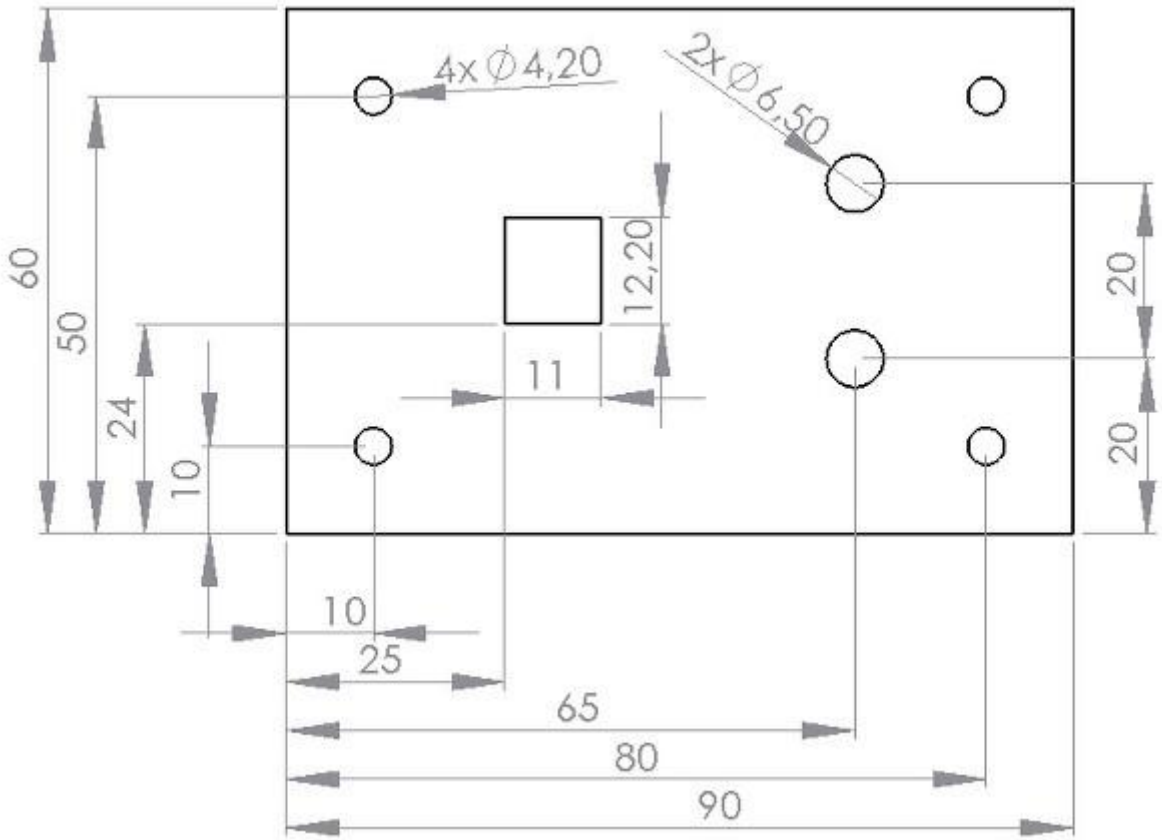
Ich denke, wir können jetzt das erste Programm schreiben.

Mein erstes Programm

Es ist immer besser, ein Beispiel für Erklärungen zu verwenden. Auf dem Bild sehen Sie einen Schalter und zwei LEDs mit ihrer Halterung. Wir werden als Beispiel die Herstellung einer Frontplatte mit einem Schalter, 2 LEDs und 4 Bohrungen in den Ecken zur Befestigung verwenden. Zu diesem Zweck steht uns eine 1,5 mm dicke Dur-Aluminiumplatte zur Verfügung. Da wir beschlossen haben, ohne CAD/CAM zu arbeiten, zeichnen wir schnell auf ein kleines Stück Papier die gewünschte Form. Das gibt uns eine Vorstellung davon, wie das Werkstück später aussieht.

Teile, die wir haben

Frontplattenskizze



Um es besser zu erklären und leichter verständlich zu machen, berücksichtigen wir im Moment NICHT den Durchmesser des Werkzeugs, ich sage dies nur, damit die Fortgeschrittenen nicht denken, dass uns etwas Wichtiges entgeht. Auch dieses Problem werden wir später lösen.

Werkstückoffset (Absolute Nullpunktverschiebung)

Wie wir sehen können, hat die Maschine ihren Ursprung in der linken unteren Ecke. Wenn wir das zu bearbeitende Stück dort ablegen, können wir es nicht richtig auf dem Tisch fixieren. Außerdem sind wir möglicherweise nicht in der Lage, den Umfang ordnungsgemäß zu fräsen. Wir müssen das Stück mehr zentriert auf den Tisch legen, wo wir es bequem und sicher befestigen können. Auch wenn wir eine virtuelle Maschine verwenden, können wir dies simulieren. Mit den Pfeilen auf dem numerischen Tastenfeld fahren wir die Maschine auf einen Punkt in der Nähe von X=50, Y=50. Dort befindet sich die linke untere Ecke des Teils, das wir in die Maschine gelegt haben.

Wenn wir uns die Zeichnung ansehen, die wir zuvor angefertigt haben, sehen wir, dass alle Messungen vom selben Punkt aus vorgenommen wurden (linke untere Ecke). Daher wäre es der richtige Ort, den Nullpunkt dort zu setzen.

Dazu wählen wir aus dem Dropdown-Menü unter den Koordinaten, **G54** (Werkstückoffset). Einmal ausgewählt, klicken wir auf die Schaltfläche "Reset" der "X"- und "Y"-Koordinaten.

Das Symbol der drei farbigen Pfeile wird auf der Spitze des Fräsers platziert.

Wenn wir in der Liste der Werkstückoffsets zwischen **G53 und** G54 hin- und herwechseln sehen wir, dass auch das Pfeilsymbol seine Position ändert.

Das erste haben wir bereits über den G-Code gelernt: den Nullpunkt oder den Werkstückoffset. Wenn wir die Koordinaten der Maschine verwenden wollen, schreiben wir in eine Zeile:

G53 (verwenden der tatsächlichen Maschinenkoordinaten)

oder die Koordinaten des Werkstücks:

G54 (verwenden der Koordinaten des Werkstücks)

Diese Einstellungen werden gespeichert, wenn Sie die Maschine aus- und wieder einschalten. Wenn Sie also z.B. einen Schraubstock auf der Maschine angebracht haben und Sie die Ecke davon finden, kann diese in **G54** gespeichert werden und man kennt immer die Position.

Es gibt mehrere Speicher für die Offsets um Positionen zu speichern. Zum Beispiel **G55, G56** - bis **G59**. Die Koordinaten, die wir in einem Programm programmieren, beziehen sich immer auf diesen von uns gewählten Nullpunkt.

Lassen Sie uns sicherstellen, dass jeder versteht, was ich meine. Öffnen Sie ein neues Programm mit Menü->Datei->Neue NC-Datei erstellen...

Wir schreiben unter das %-Symbol diese Zeilen:

```
G53
G0 X0 Y0
G0 X100
G54
G0 X0 Y0
G0 X100
M30
```

Nach dem Speichern sehen wir das Programm im unteren Fenster und oben im Grafikfenster sehen wir einige rote Linien.

Nun starten wir das Programm mit der Schaltfläche "Programm starten". Der Fräser bewegt sich entsprechend den von uns programmierten Koordinaten. Nachdem das Programm

beendet ist, wird es zurückgespult, und wir können mehrmals auf "Programmstart" klicken, um es gut zu sehen.

Aber was haben wir wirklich getan?

Okay, im Programm haben wir der Maschine gesagt, dass sie die tatsächlichen Koordinaten der Maschine verwenden muss.

Dies sagt der Befehl **G53.** Dann mit dem Befehl **G0** wir haben der Maschine gesagt, dass sie sich mit maximaler Geschwindigkeit bewegen muss, schnell vorwärts, auf den Punkt **X0Y0** und dann auf den Punkt **X100.** Die Maschine bewegt nur die von uns angegebenen Achsen. In der Z**eile G0 X100** haben wir kein Y verwendet, dewegen bleibt diese Koordinate auf 0. Dann sagen wir der Maschine, dass sie die Koordinaten des Werkstücks, **G54** benutzen soll, welche wir vorher gespeichert haben.

```
G54
G0 X0Y0
```

Die Maschine bewegt sich nicht zu ihrem wirklichen Punkt von X=0, Y=0, sondern bewegt sie sich zu dem Punkt ungefähr X50, Y50, das ist der Punkt, wo wir die Ecke des Materials eingestellt haben. Aber die Koordinaten zeigen bereits X=0, Y=0 und den folgenden Befehl

```
G0 X100
```

Bewegen Sie das Werkzeug in Bezug auf diesen Punkt 100 mm nach rechts. Das Werkzeug befindet sich bereits in Position ungefähr X=150, Y=50, aber in den Koordinaten finden wir genau X=100, Y=0.

Dies zu verstehen ist sehr, sehr wichtig, denn es ist die Grundlage aller Bewegungsbefehle an der Maschine.

Die ersten Bewegungen

Unten links haben wir eine Schaltfläche "Programm bearbeiten". Damit öffnen wir den Editor, löschen alle Zeilen hinter dem %-Symbol und schreiben:

```
G54
G0 Z20
X0 Y0
Z2
G1 Z0 F100
X90 F800
Y60
X0
Y0
G0 Z20
M30
```

Jetzt speichern wir das Programm und starten es erneut. Wir können es mehrmals starten um zu sehen, was passiert. Als erstes bewegt sich der Fräser nach oben. Dies ist wichtig, damit es zu keiner Kollision mit dem Material auf dem Tisch kommt. Auf der zweiten Zeile stehen nur Koordinaten.

Warum weiß die Maschine, was zu tun ist?

Der G0-Befehl, den wir vorhin gesagt haben (und den wir bereits kennen), ist so lange aktiv, bis wir ihn ändern. Die Maschine merkt sich den letzten Befehl, und es ist nicht nötig, ihn erneut einzugeben. Viele Befehle werden auch gespeichert, wenn wir die Steuerung aus- und wieder einschalten. Aber passen Sie auf, denn die Maschine merkt sich zwar den letzten Befehl, aber wir sind vielleicht nicht sicher, was das war. Zum Beispiel wissen wir bei einem Programm zu Beginn nicht sicher, welchen Status es hat und welcher G-Code der letzte im vorherigen Programm war. Wir müssen also die Befehler wiederherstellen, nur um ganz sicherzugehen.

Wir werden später mehr über diese Dinge sprechen. Aber kommen wir zur Linie:

Der Fräser bewegt sich auf die Position X=0, Y=0, das ist die Ecke des Materials. Mit dem Z2-Befehl bleibt der Fräser 0,5 mm über dem Material (weil wir gesagt haben, dass wir 1,5 mm dickes Material haben).

Jetzt haben wir einen neuen Befehl, G1. Mit der Zeile `G1 Z0 F100` dringt der Fräser in das Material ein, aber langsam. Die Vorschubgeschwindigkeit beträgt 100mm/min und es dauert eine Weile, bis der Tisch bei Z=0mm erreicht wird.

In der nächsten Zeile `X90 F800` ist der Befehl `G1` noch aktiv und die Maschine fährt nur mit der X-Achse auf die Position 90mm mit einer Geschwindigkeit von 800mm/min, die wir bereits mit Parameter F angegeben haben. Dann folgen die Koordinaten des Rechtecks,

immer mit dem gleichen Befehl `G1` und der gleiche Geschwindikeit von 800mm/min. Wenn der Punkt X=0, Y=0 erreicht ist, wird der Befehl auf Eilgang geändert mit `G0` und wir heben den Fräser auf die sichere Position von 20-mm an.

In der letzten Zeile finden wir einen neuen Befehl: `M30`. Die Befehle `Mxx` sind Makros.Ein Makro ist ein im System gespeichertes Programm. Es ist ein Programm aus ein paar Codezeilen, die innerhalb des Maschinensystems etwas ausführen. Eines dieser Makros sind `M2` y `M30`. Mit diesen beiden Makros beenden wir das Programm. M2 beendet einfach das Programm und `M30` spult das Programm zusätzlich auf Anfang zurück. Es ist nicht unbedingt notwendig, ein Programm zu beenden mit `M2` o `M30`. Wenn es keine Codezeilen mehr gibt, stoppt die Maschine. Aber es ist ein guter Programmierstil, in der letzten Zeile immer ein `M30` zu setzen. Einige Maschinen schalten möglicherweise nicht automatisch alle Ausgaben ab. Und in Zukunft werden wir mit der Programmierung von Unterprogrammen beginnen. Und in diesem Fall ist es notwendig, die `M30` am Ende des Programms zu schreiben. Es ist also besser, es von Anfang an aufzuschreiben, auch wenn es nicht immer notwendig ist.

Zusammenfassung des ersten Kapitels

Nun wissen wir bereits, wie das Werkstück auf der Maschine auszurichten ist, und wir wissen, wie wir den Nullpunkt durch den Werkstücks-Offset definieren, die Geschwindigkeit der Bewegung ändern, im Eilgang und im Arbeitsvorschub fahren, wir können Löcher mittels eines Zyklus herstellen und wir wissen, wie wir das Programm beenden.

Das wissen wir schon:

- `G0` - Eilgang
- `G1` - Arbeitsvorschub, definiert durch Parameter F
- `F` - Vorschubgeschwindigkeit in mm/min
- `G53` - Koordinaten im Maschinensystem (Koordinaten in Bezug auf den Maschinennullpunkt)
- `G54-G59` - Koordinaten des Werkstücks (Koordinaten in Bezug auf den Werkstücknullpunkt)
- `M2` - Ende des Programms
- `M30` - Programmende mit Rückspulen.

Korrektur des Werkzeugdurchmessers

Gehen wir zurück zu dem Programm mit dem Rechteck von oben. Wir haben das Material tatsächlich direkt "auf der Linie" geschnitten. Angenommen, wir haben einen Fräser mit 3 mm Durchmesser, dann wird das verbleibende Stück kleiner sein, als wir gezeichnet haben. Natürlich schneiden wir mit dem Fräser entlang der Linie, und die Hälfte des Fräsers befindet sich im Material. Es gibt zwei Möglichkeiten, dies zu lösen. Wir können das Programm unter Berücksichtigung des bekannten Durchmessers des Fräsers "umprogrammieren" und schreiben:

```
G54
G0 Z20
X-1.5 Y-1.5
Z2
G1 Z0 F100
X91.5 F800
Y61.5
X-1.5
Y-1.5
G0 Z20
M30
```

Jetzt fräsen wir das Werkstück korrekt. Aber auf diese Weise müssen wir eine Menge Berechnungen anstellen, bei denen wir Fehler machen können, und wir sehen den Fehler nicht direkt bei so vielen seltsamen Koordinaten.

Welche andere Lösung gibt es?

Der G-Code hat die Möglichkeit, den Werkzeugdurchmesser automatisch zu korrigieren. Wir öffnen das Programm erneut im Editor und löschen alle Zeilen hinter dem %-Symbol. Und wir schreiben:

```
N100 G54
N110 G40
N120 G0 Z20
N130 G0 Y-1.6 X0
N140 Z2
N150 G1 Z0 F100
N160 G42
N170 G1 Y0
N180 X90 F800
N190 Y60
N200 X0
N210 Y0
N220 G40
N230 Y-1.6
```

```
N240 G0 Z20
N250 M30
```

Was sehen wir jetzt?

Zu Beginn haben wir in der zweiten Zeile **N110 G40**. N110 ist die Nummerierung der Zeilen, und wir können sie zur Organisation des Programms verwenden. Sie hat keinen Einfluss auf das Programm, sie dient nur dazu zu wissen, in welcher Position wir uns innerhalb eines Programms befinden, falls es länger ist.Und wir haben einen neuen Befehl gefunden: **G40.**

Dieser Befehl deaktiviert die automatische Werkzeugdurchmesserkorrektur. In diesem Fall sagen wir es, um deutlich zu machen, dass sie noch nicht für Anfahtsbewegungen aktiviert ist.

In Zeile N130 gehen wir mit der Maschine nicht bis zum Punkt X=0, Y=0, denn wenn wir dorthin gingen, würden wir das Material schneiden. Wir müssen etwas weiter als die Hälfte des Fräsers entfernt bleiben. Der Abstand ist nicht wichtig, er kann auch -10 mm betragen. Aber immer mehr als die Hälfte des Durchmessers.

Sobald wir im Material sind, machen wir eine andere Bewegung:

```
N160 G42
N170 G1 Y0.
```

Mit **G42** aktivieren wir die automatische Fräserdurchmesserkompensation. Mit der Bewegung **G1 Y0 i**n unserem Beispiel bewegt sich die Maschine nur 0,1 mm in eine positive Y-Richtung. Da sind wir genau am Rand des Werkstückes.
Im Grafikfenster sehen wir eine graue Linie. Dies ist das eigentliche Werkstück, und die grüne Linie ist der Werkzeugweg. Wir befinden uns jetzt in einem virtuellen Koordinatensystem und können die realen Koordinaten des Werkstücks unabhängig vom Werkzeugdurchmesser verwenden.

Spielen wir ein bisschen mit den Zeilen, und ändern diese so ab:

N130 G0Y-10X0

Und versuchen wir es noch einmal. Jetzt ändern wir:

N160 G41

und probieren es erneut.

Okay - wenn wir mit diesen Parametern spielen, könnten wir eine Sache seltsam finden:

Woher kennt das Programm den Durchmesser des Fräsers?

In keiner Zeile haben wir es gesagt! Gehen Sie zu Menü->Einstellungen->Werkzeuge. Dort finden wir eine Liste der bekannten Werkzeuge. Sie können die Liste bearbeiten und den Durchmesser des Werkzeugs 1 ändern. Wenn wir den Dialog verlassen schauen wir auf die linke Seite des Hauptbildschirms. Dort steht auf einem blauen Feld mit gelben Buchstaben T1. Dies ist das Werkzeug, das derzeit verwendet wird.

Aber es gibt eine interessante Sache:

CAM-Programme führen die Fräserkompensation in der Regel direkt an den Koordinaten durch. Wenn wir das Stück gefräst haben und feststellen, dass das Maß nicht genau 90 mm, sondern 90,2 mm beträgt, müssten wir das G-Code-Programm mit dem CAM erneut berechnen und an die Maschine senden.Aber wenn wir zur Werkzeugliste gehen und den Durchmesser des Fräsers von 3 mm auf 2,6 mm ändern, schneidet er um 2 Zehntel näher am Material und das Maß ist korrekt.Auch wichtig, wenn wir ein einmal erstelltes und gespeichertes Programm nochmals verwenden und wir nicht den passenden Fräser haben.

Eine andere Sache ist der Unterschied zwischen `G41` y `G42`. Mit diesen Befehlen ändern wir die Seite des Fräsers in Bezug auf die Richtung der Bewegung. Mit `G42` der Fräser immer auf der rechten Seite ist, in `G41` immer auf der linken Seite.

Wir wissen bereits, wie man den Durchmesser eines Fräsers korrigiert, und wir verwenden es direkt, um mit dem Teil zu beginnen, das wir herstellen wollen.

Wieder öffnen wir das Programm im Editor und löschen alle Zeilen hinter dem %-Symbol. Schreiben wir folgendes:

```
G54
G40 (Radiuskorrektur aus)
G0 Z20
(kleine Bohrung)
G0 Y26 X27 (innerhalb der Bohrung)
Z2
G1 Z0 F100
G41 (Korrektur links)
G1 Y24 X25 F800
X=X+11
Y=Y+12.2
X25
Y24
G40
G1 Y26 X27
G0 Z20
(Kontur aussen)
G0 X0 Y-2
G0 Z2
G1 Z0 F100
```

```
G42
G1 Y0 F800
X90
Y60
X0
Y0
G40
G1 Y-1.6
G0 Z20
M30
```

Wenn wir uns das Grafikfenster ansehen, sehen wir graue und grüne Linien. Die grauen sind die Kontur des Werkstücks, und die grünen zeigen den Werkzeugweg an.

Aber was geht in diesen Zeilen vor sich?

X=X+11
Y=Y+12.2

Dies ist eine G-Code-Berechnung. Ja! G-Code kann "-,+,*,/,sin,cos,sqrt" berechnen. Doch vorerst sehen wir uns nur diese beiden Zeilen an: In unserer Zeichnung haben wir das Rechteck 11x12,2 mm.

Natürlich könnten wir schreiben:

```
X36
```

```
Y36.2
```

Aber wenn wir es so machen, ist es schwerer verständlich und später auch nicht so einfach abzuändern, wenn wir die Maße mal anpassen müssen. Auf diese Weise ist es viel übersichtlicher und auch sehr viel einfacher, nachträglich die Abmessungen zu verändern.

Die ersten Löcher: G81

Mit dem, was wir gelernt haben, können wir die Maschine bereits an einen bestimmten Punkt bewegen, so dass wir die Löcher in den Ecken bereits herstellen können. Wieder öffnen wir das Programm und bearbeiten es, wir löschen alle Zeilen nach dem %-Symbol.

Schreiben wir folgendes:

```
G54
G0 Z20
X10 Y10
Z2
G1 Z0 F100
G0 Z2
X80
Z2
G1 Z0 F100
G0 Z2
Y50
Z2
G1 Z0 F100
G0Z2
X10
Z2
G1 Z0 F100
G0 Z20
M30
```

Okay, mit dem Wissen, das wir bis jetzt haben, könnten wir es so schreiben und es funktioniert. Das Werkzeug fährt zu den 4 Ecken, fährt schnell über das Material und bohrt langsam bis auf den Tisch, die Koordinate Z=0. Aber wenn wir viele Löcher haben, wird die Sache kompliziert, und es ist nicht einfach, so zu programmieren.

Dafür gibt es ein neues Kommando `G81`. G81 ist ein Bohrzyklus.

Was ist ein Zyklus?

Es ist ein Programm, das in G-Code geschrieben und im System der Maschine gespeichert ist. Es hilft uns sehr, wir müssen nur den Befehl G81 und die Lochkoordinaten sagen. Wir öffnen das Programm erneut im Editor und löschen alle Zeilen hinter dem %-Symbol.

Schreiben wir folgendes:

```
G54
G0 Z20
X10 Y10
G81 Z0 R2 F100
```

```
X80
Y50
X10
G0 Z20
M30
```

Viel kürzer und leichter zu schreiben, aber was passiert im Detail:

In der dritten Zeile stehen wir über dem ersten Loch. Der Zyklus `G81` hat einige Parameter auf seiner Seite. Der erste ist die Tiefe Z, die wir bohren wollen, in unserem Fall Z=0. Der zweite Parameter ist die Höhe, auf der wir das Werkzeug nach der Herstellung der Bohrung zurückziehen wollen, und der Parameter F, den wir bereits kennen, ist die Vorschubgeschwindigkeit, mit der wir das Werkzeug in das Material absenken.

Wie jeder G-Code-Befehl ist auch dieser Befehl modal, d.h. er bleibt so lange aktiv, bis wir ihn ändern. In Zeile X80 fährt die Maschine also auf diese Position, bleibt aber dort nicht stehen - der Befehl G81 ist immer noch aktiv und die Maschine wiederholt diesen Befehl und macht ein Loch und das Werkzeug kehrt auf eine Höhe von 2 mm zurück.

Dies wird so lange wiederholt, bis wir den Befehl zurücksetzen. Wir können dies zum Beispiel tun mit `G0Z20`. Damit schalten wir auf Eilgang und heben das Werkzeug auf eine sichere Höhe.

Für tiefere Bohrungen gibt es andere Zyklen, zum Beispiel mit Spanbrechen. Aber das kommt vorerst nicht in Frage, wir werden das später besprechen.

Machen wir die ersten (virtuellen) Späne

Bisher haben wir der Maschine noch nicht gesagt, dass sie Späne erzeugen muss. Jede Maschine hat eine andere Spindel, aber im G-Code gibt es einige Befehle, die immer gleich sind.

- `M3` normaler Start (Rechtslauf)
- `M4` Gegen den Uhrzeigersinn einschalten
- `M5` Ausschalten der Spindel (Spindelstopp)
- `S` Geschwindigkeit, mit der die Spindel dreht.

Die Befehle `M3`-`M5` sind Makros, die in der Maschine gespeichert sind. Diese Makros können vom Maschinenhersteller oder einem fortgeschrittenen Benutzer geändert werden. Es kann nützlich sein, wenn z.B. ein automatischer Wechsler vorhanden ist und die Spindel ohne ein Werkzeug im Inneren nicht eingeschaltet werden darf.

Aber schauen wir mal, wie wir dies im Programm benutzen.

Wir öffnen das Programm mit dem Editor und ändern ein paar Zeilen:

Schreiben wir folgendes:

```
G54
G40 (Radiuskompensation aus)
G0 Z20
(kleine Bohrung)
G0 Y26 X27 (innerhalb der Bohrung)
Z2
M3 S1000 (Spindel mit normaler Drehrichtung, 1000
U/min)
G1 Z0 F100
G41 (Korrektur links)
G1 Y24 F800 (Bewegung > halber Bohrerdurchmesser)
X36
Y36.2
X25
Y24
X27
G40 (Radiuskompensation aus)
G1 Y26 (Bewegung > halber Bohrerdurchmesser)
G0 Z20
M5
(Außenkontur)
G0 X0Y-2
G0 Z2
M3 S1000
G1 Z0 F100
G42 (Korrektur rechts)
G1 Y0 F800
X90
Y60
X0
Y0
G40
G1 Y-2 (Bewegung > halber Bohrerdurchmesser)
G0 Z20
M5
M30
```

Finden Sie die Zeilen, die wir zum Ein- und Ausschalten der Spindel hinzufügen mussten? Wir werden es deutlicher sehen, wenn wir das Programm starten und uns das Grafikfenster ansehen. Immer, wenn wir schreiben `M3` rotiert das Werkzeug, mit einem `M5` steht er still. Wenn die Simulation zu schnell ist, können Sie die Geschwindigkeit mit dem Vorschub-Override F% von 150% bis 0% anpassen.

Am Ende des Programms war es nicht notwendig, ein **M5 zu schreiben,** weil ein **M2** o **M30** schaltet immer alles aus. Aber es ist ein guter Schreibstil, also ist es besser, vor allem, wenn jemand anderes es lesen muss.

Kreise und Kurven G2/G3

Bislang können wir nur lineare Bewegungen ausführen. Das bedeutet, dass wir nicht um eine Ecke fräsen oder eine Zirkularfräsung durchführen können. In unserem Beispiel der Frontplatte wollen wir die 4,2 mm und 6,5 mm Löcher machen.

Wie wir wissen, können wir Löcher mit dem Bohrzyklus herstellen, aber dafür müssen wir einen Bohrer mit genau dem Durchmesser als Werkzeug einsetzen. Und das für jedes Loch, das wir machen wollen.

In unserem Fall gäbe es 2 verschiedene Bohrer, 4,2 und 6,5 mm. Es gibt Fräsmaschinen, die nicht über einen automatischen Werkzeugwechsler verfügen, es gibt auch Spindeln, in die keine Bohrer passen, die größer als 6 mm sind.

Wie können wir das also beheben?

Wir machen die Löcher mit dem Fräser.

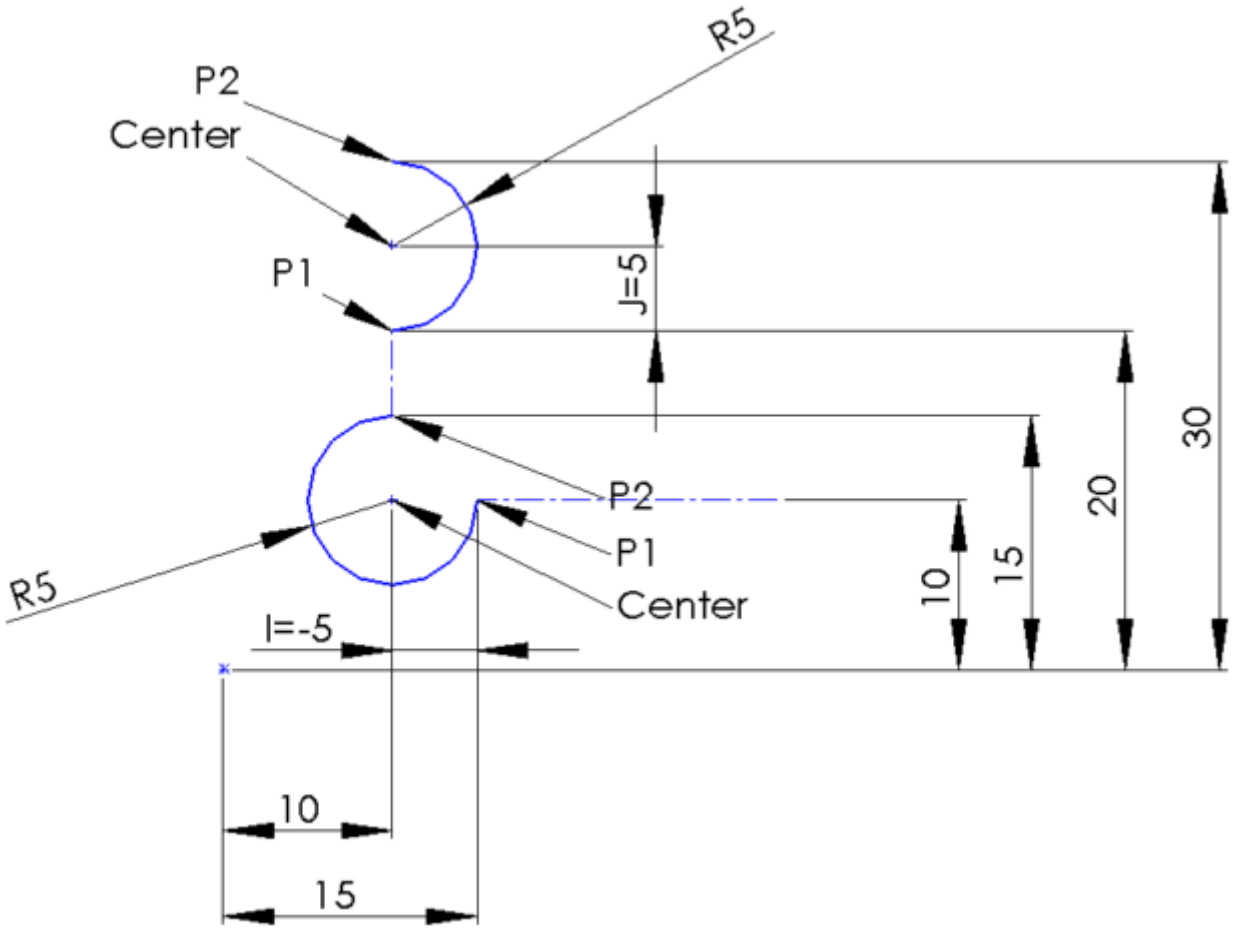
Bevor wir mit der Frontplatte weitermachen, sollten wir ein paar Konzepte erstellen. Wieder öffnen wir das Programm im Editor und löschen alle Zeilen hinter dem %-Symbol.

Schreiben wir folgendes:

```
N100 G54
N110 G0 Z20
N120 G0 X15 Y10
N130 Z2
N140 M3
N150 G1 Z0 F100
N160 G2 X10 Y15 I-5 J0 F800
N170 G0 Z20
N180 Y20
N190 Z2
N200 G1 Z0 F100
N210 G3 X10 Y30 I0 J5 F800
N220 G0 Z20
N230 M5
N240 M30
```

Das Grafikfenster des Programms zeigt zwei Bögen in grüner Linie. Wenn wir das Programm starten, folgt der Fräser diesen Kurven, indem er diejenige von ¾ im Uhrzeigersinn und die 180º-Kurve in umgekehrter Richtung ausführt. Schauen wir genauer hin, was passiert, dazu haben wir hier eine Skizze:

Kurven



In der Zeile N120 gehen wir zu den Koordinaten X15, Y10. Dies ist der Startpunkt P1 des ersten Bogens. In Zeile N160 haben wir den ersten Befehl für Bögen, G2. Wie wir gesehen haben, G2 macht immer Bögen im Uhrzeigersinn. G3 aber in der entgegengesetzten Richtung. Mal sehen, wie es funktioniert:

1. Fahren mit einem beliebigen Befehl zum Startpunkt P1. Das Wichtigste ist, dass wir da sind.
2. Wir schreiben: G2 Koordinaten des Endpunkts P2, Koordinaten des Mittelpunkts der Kurve (Center point)
3. Wir beendeten den ersten Bogen am Punkt P2.

Es gibt eine Sache zu beachten: Die Koordinaten des Zentrums müssen in dem von der Maschine gewünschten Format eingegeben werden. Es gibt zwei Möglichkeiten. Der Mittelpunkt kann in absoluter oder relativer Form angegeben werden.

Unser Fall ist die relative Form, also nicht die absoluten Realkoordinaten, die in diesem Beispiel ja X10 Y10 wären. Wir setzen die relativen Koordinaten in Bezug auf den Punkt P1, den Startpunkt.

```
P1=X15 Y10
Mitte=X10 Y10
I=X=(10-15)=-5
J=Y=(10-10)=0
```

Der Buchstabe I ist immer die X-Koordinate und der Buchstabe J ist immer die Y-Koordinate. Wir müssen zwei verschiedene Buchstaben verwenden, weil wir X nicht zweimal schreiben können, für den Punkt P2 und für die Mitte. Im G-Code für das Zentrum ist also X=I, Y=J.

Okay, machen wir mit dem nächsten Bogen weiter. Betrachten wir die Skizze, dann sehen wir:

1. Wir befinden uns am Startpunkt P1, der X=10, Y=20 ist.

2. Wir wollen zum Endpunkt P2 gelangen, der X=10, Y=30 ist.

3. Der Mittelpunkt ist X=10, Y=25.

4. Wir berechnen die I(X)- und J(Y)-Koordinaten des Zentrums:

P1=X10 Y20

C= X10 Y25 => I=0, J=5

Eine Sache, die sehr, sehr wichtig zu wissen ist:

Der Bogenradius muss am Startpunkt P1 und am Austrittspunkt P2 gleich sein. Das heißt, der Radius kann nicht verändert werden.

Verändern wir die Zeile

```
N210 G3 X10 Y30 I0 J5 F800 durch
N210 G3 X10 Y30 I0 J4 F800
```

so dass der Radius am Startpunkt 4 mm und am Ende 6 mm betragen würde. Wenn wir die Änderungen speichern, kann das Programm nicht mehr ausgeführt werden, da ein Fehler vorliegt. Die Beamicon2-Steuerung hat eine Einstellung, bei der eine Abweichung (Toleranz) noch akzeptiert wird, normalerweise ist das 0,001 mm.

CAM-Programme von schlechter Qualität können G-Code mit diesen Fehlern erzeugen, und in einigen Fällen kann es notwendig sein, diese Toleranz zu erhöhen.

Okay - zum Üben zeichnen wir einige Kreise auf Papier und berechnen die Koordinaten für `G2` y `G3`.

Exakter Stopp G61 und konstante Geschwindigkeit (CV) G62/G64

Wenn eine Maschinenachse in Bewegung ist, ist es klar, dass sie einige Zeit braucht, bis sie anhält. Es ist eine Brems-Rampe, die nicht Null sein kann. Eine Maschine kann nicht in 0 Sekunden bremsen. Die Bremszeit hängt von vielen Dingen ab, z.B. vom bewegten Gewicht, von der Geschwindigkeit, von der Struktur der Maschine usw. Aus diesem Grund hat der Maschinenhersteller für jede Achse einen Beschleunigungswert festgelegt.

Sehen wir uns noch einmal die Skizze unserer Frontplatte an. Außen haben wir ein Rechteck. Die Ecken sind 90° - und die Maschine kann die Richtung nicht in 0 Sekunden ändern. Es ist wie bei einem Auto, wenn wir eine gerade Straße haben, können wir sehr schnell fahren, aber vor der nächsten Kurve müssen wir bremsen, bis wir die Kurve sicher nehmen können. Dasselbe gilt für die Maschine. Wenn wir wollen, dass die Kurve 90° beträgt und die Messungen 100% genau sind, müssen wir anhalten und in der anderen Richtung wieder beschleunigen. Dies erzeugt einige Vibrationen in der Maschine und dauert sehr lange. Aber der Werkzeugweg ist 100% wie in der NC-Datei. Dies ist der Modus "Exakter Stopp" und wir aktivieren ihn mit dem Befehl G61.

Für den Fall, dass wir nicht so viel Perfektion benötigen, z.B. wenn wir mit Holz oder Materialien/Stücken arbeiten, die keine so hohe Präzision benötigen, können wir auf den Modus mit konstanter Geschwindigkeit umschalten. Dies wird aktiviert durch den Befehl `G62` o `G64` (wegen der Kompatibilität zwischen verschiedenen Maschinen gibt es zwei Befehle, die das Gleiche tun, sie sind exakt gleich).

Aber was passiert bei diesem Befehl?

In diesem Modus stoppt die Maschine nicht. Es ist wie in einem Auto - oder besser mit einem Motorrad. In sehr engen Kurven bremsen wir nicht auf Null - wir schneiden die Kurve auf der Innenseite ab, damit wir schneller fahren können. Die Maschine macht dasselbe. Auf einer Kurve oder einer Ecke verläuft sie nicht zu 100% entlang der gezeichneten Bahn. Die Maschine schneidet den Weg, so dass sie nicht so viel bremsen muss. Auf diese Weise können wir so viele Bremsungen und Pausen vermeiden, die Maschine läuft viel ruhiger und ist schneller fertig. Aber wir werden einen Unterschied zwischen dem Pfad in der NC-Datei und dem tatsächlichen Teil haben. Dieser Unterschied, den wir zulassen, kann in der Konfiguration der Maschine angegeben werden.

Absolut-Modus/ Inkremental-Modus G90/G91

Als wir Bögen/Kreise mit den Befehlen `G2/G3` gemacht haben, haben wir zum ersten Mal relative Koordinaten verwendet.

Wir arbeiten normalerweise in absoluten Koordinaten `G90`, die Punkte beziehen sich auf den Ursprung der Maschine oder des Teils, wie wir festgestellt haben. Aber wir können auch im inkrementellen Modus arbeiten `G91` und in diesem Fall beziehen sich die Koordinaten auf den Punkt, an dem wir uns befinden. Um zu einem neuen Punkt zu gelangen, geben wir also die X-, Y- und Z-Werte dieses Punktes in Bezug auf den aktuellen Punkt ein.

Ein Beispiel:

```
G90
G0 X20 Y20
G91
G0 X5 Y10
```

Wo stehen wir nach diesen Befehlen?

Ja, genau. Wir sind bei X25, Y30.

Die Befehle `G90` Y `G91` sind modal, d.h. wenn sie einmal programmiert sind, bleiben sie aktiv, bis der umgekehrte Befehl gegeben wird. Der inkrementelle Modus `G91` wird oft in Unterprogrammen verwendet, später werden wir es auch verwenden.

Der Start eines NC-Programms

Wie wir gelernt haben, ist der G-Code eine modale Sprache, d.h. der Status der Befehle ändert sich nicht, bis wir einen neuen Befehl ausführen. Dies zu wissen ist sehr wichtig, denn zu Beginn eines Programms kennen wir den Zustand der Maschine nicht. Deshalb ist es immer sehr wichtig, einen Kopf mit den wichtigsten Befehlen zu schreiben, um einen bekannten Zustand zu erzeugen.

Zum Beispiel ist es besser, am Anfang zu wissen, dass wir uns im Absolut-Modus befinden und dass alle Korrekturen deaktiviert sind.

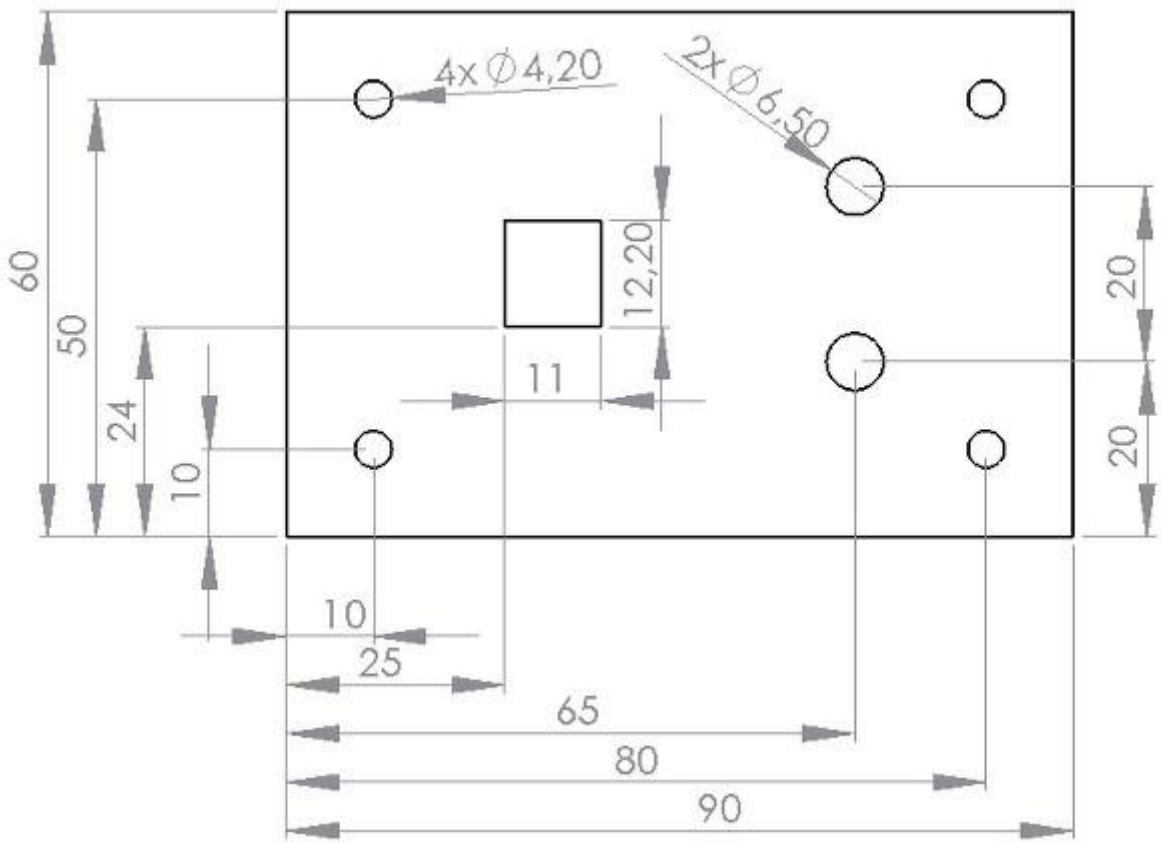
(Meine Startdefinition) => ein paar Kommentare vorneweg helfen Ihnen, wenn Sie das Programm später wieder benötigen.

- % => das Startsymbol
- `G54` => Der Offset, den wir verwenden (kann `G53` bis `G59` `sein`)
- `G40` `G80` => Modus `G41/G42` `abschalten und Zyklen deaktivieren.`
- `G90` =>absoluter Modus
- `G61` =>exakter Stopp

Damit haben wir die Maschine "konfiguriert", und wir kennen den Zustand der wichtigsten Befehle. Es ist immer besser, diese Zeilen vor einem Programm einzufügen. Natürlich müssen wir die Zeilen entsprechend der gewünschten Konfiguration ändern. Wer zum Beispiel mit einem Laser arbeitet, verwendet normalerweise `G62` und je nach dem von uns verwendetem Offset ändern wir `G54`.

Unser Gesamtwerk - die Frontplatte

Skizze der Frontplatte



Wenn wir diese Frontplatte herstellen wollen, müssen wir zuerst ein wenig nachdenken:

Welche Werkzeuge haben wir?

-Welches Element fräsen wir zuerst, welches zuletzt?

Okay - ich sage jetzt mal, wir haben nur einen Fräser mit 3 mm Durchmesser, sonst nichts. Und wenn wir die Platte auf dem Frästisch befestigt haben, müssen wir natürlich zuerst die inneren Elemente schneiden und dann die äußeren.

Wir beginnen mit dem NC-Programm, Schritt für Schritt:

```
(Frontplatte)
%
G54
G40 G80 G90 G61
G0 Z40
(Werkzeugwechsel)
M6 T1
(Einschalten der Spindel mit 3000 U/min)
M3 S3000
```

```
(wir gehen auf die sichere Z-Höhe)
G0 Z40
(wir fräsen das innere Rechteck)
G0 X27 Y26
G0 Z2
G1 Z0 F30
G41 G1 Y24 F800
G1 X36
G1 Y36.2
G1 X25
G1 Y24
G1 X27
G40 G1 Y26
G0 Z40
(zur Mitte der ersten Bohrung gehen)
G0 X65 Y20
G0 Z2
G1 Z0 F30
G42 G1 X68.25 F800
G2 X68.25 Y20 I-3.25 J0
G40 G1 X65
G0 Z40
G0 X65 Y40
G0 Z2
G1 Z0 F30
G42 G1 X68.25 F800
G2 X68.25 Y40 I-3.25 J0
G40 G1 X65
G0 Z40
(wir fräsen die 4 Löcher außen)
G0 X80 Y10
G0 Z2
G1 Z0 F30
G42 G1 X82.1 F800
G2 X82.1 Y10 I-2.1 J0
G40 G1 X80
G0 Z40
G0 X80 Y50
G0 Z2
G1 Z0 F30
G42 G1 X82.1 F800
G2 X82.1 Y50 I-2.1 J0
G40 G1 X80
G0 Z40
G0 X10 Y50
G0 Z2
G1 Z0 F30
```

```
G42 G1 X12.1 F800
G2 X12.1 Y50 I-2.1 J0
G40 G1 X10
G0 Z40
G0 X10 Y10
G0 Z2
G1 Z0 F30
G42 G1 X12.1 F800
G2 X12.1 Y10 I-2.1 J0
G40 G1 X10
G0 Z40
(schließlich fräsen wir die Außenkontour)
G0 X-2 Y-2
G0 Z2
G1 Z0 F30
G42 G1 X0 Y0 F800
G1 X90
G1 Y60
G1 X0
G1 Y0
G40 G1 X-2 Y-2
G0 Z40
M5
M2
```

Mit dem, was wir bisher gelernt haben, sind wir in der Lage, G-Code-Programme zu schreiben, um fast jede Form herzustellen. Mit diesem Programm können wir ein bisschen spielen. Fügen Sie weitere Löcher hinzu, ändern Sie die Form der Platte usw.

Erweiterte Programmierung

Schauen wir uns das geschriebene Programm noch einmal genauer an:

(Fräsen der 4 äußeren Löcher)

```
G0 X80 Y10
G0 Z2
G1 Z0 F30
G42 G1 X82.1 F800
G2 X82.1 Y10 I-2.1 J0
G40 G1 X80
G0 Z40
G0 X80 Y50
G0 Z2
G1 Z0 F30
G42 G1 X82.1 F800
G2 X82.1 Y50 I-2.1 J0
G40 G1 X80
G0 Z40
```

Diese Zeilen werden immer in ähnlicher Weise wiederholt, sie sind sehr ähnlich. Und wir haben nur 4 Löcher, stellen Sie sich vor, es sind 100 Löcher - wir müssten viele dieser Zeilen immer gleich schreiben. Es ist ein langweiliger Job, und bei so vielen Zahlen können wir Fehler machen.

Auch in anderen Teilen des Programms können wir die Dinge einfacher machen. Aber wir fangen bei Null an. Die G-Code-Sprache ermöglicht uns Berechnungen.

Probieren wir den folgenden G-Code aus:

```
G0 X20 Y20
G0 X=X+10 Y=Y +20
```

Ganz genau. Wir können mit allen Werten rechnen. Wir können +,-,*,/ und noch mehr, wie z.b. SIN(), COS() SQRT() verwenden. Wir können Formeln schreiben wie `G1 X=SIN(Y*20)` zum Beispiel.

Um dies weiter zu vervollständigen, haben wir auch Variablen.

Und was ist eine Variable?

Es ist ein Speicher zum Speichern eines numerischen Wertes. Variablen haben einen Namen, zum Beispiel #5. Wenn wir etwas in einer Variablen speichern wollen, schreiben wir #5=10. Von nun an und bis zur Änderung hat die Variable #5 den Wert 10.

Kommen wir zum Beispiel zu den Löchern von oben. Schreiben wir mal etwas in der Art:

```
%
#0=80
#1=10
#2=4.2
#3=0
G0 X=#0 Y=#1
G0 Z=#3
G1 Z=#3 F30
G42 G1 X=X (#2/2) F800
G2 X=#0 (#2/2) Y=#1 I=-1*(#2/2) J0
G40 G1 X=#0
G0 Z40
```

Auf den ersten Blick viel komplizierter als das obere Programm. Aber nur auf den ersten Blick. Erklären wir das mal Schritt für Schritt:

Für einen Kreis sind der Punkt P1 und der Punkt P2 eines Bogens gleich, Startpunkt und Endpunkt. Wir speichern die X-Koordinate in Variable #0 und die Y-Koordinate dieses Punktes in Variable #1. Der Radius ist nicht sehr einfach zu handhaben, daher halten wir den Durchmesser des Kreises in der Variablen #2. Und am Ende Variable #3, die Tiefe, in der wir das Loch machen wollen.

In der ersten Zeile gehen wir zum Mittelpunkt des Kreises, `G0 X=#0 Y=#1` ist dasselbe wie `G0 X80 Y10.` In der nächsten Zeile gehen wir auf die Höhe 2 mm höher als die Tiefe, um das Material nicht zu berühren, machen wir 1 mm daraus. In dieser Zeile gehen wir dann in das Material:

`G1 Z=#3 F30` ist dasselbe wie `G1 Z0 F30`

Die Zeile

`G42 G1 X=X (#2/2) F800` ist dasselbe wie

`G42 G1 X82.1 F800.`

Und die Zeile

`G2 X=#0 (#2/2) Y=#1 I=-1*(#2/2) J0` ist dasselbe wie

`G2 X82.1 Y10 I-2.1 J0`

Zum Schluss noch `G40 G1 X80`- und wir sehen, es ist der gleiche Code wie im vorherigen Beispiel.

Aber warum brauchen wir eine so komplizierte Sache?

Schreiben wir mal folgendes Programm:

```
%
G40 G54
#2=4.2
#3=0
#0=10
#1=10
M6 T1
M3 S30
M98 P1000
#0=80
M98 P1000
#1=50
M98 P1000
#0=10
M98 P1000
M5
M2

O1000 (Löcher)
(in: #0=X, #1=Y)
( #2=Durchmesser)
( #3=Tiefe)
G0 X=#0 Y=#1
G0 Z=#3
G1 Z=#3 F30
G42 G1 X=X (#2/2) F800
G2 X=#0 (#2/2) Y=#1 I=-1*(#2/2) J0
G40 G1 X=#0
G0 Z40
M99
```

Wir speichern und schauen auf den Bildschirm. Wir sehen 4 Löcher, mit Radiuskorrektur, perfekt in den Ecken. Das Programm ist bereits viel klarer. Wir können noch viele weitere Löcher hinzufügen, und es ist immer noch sehr klar geschrieben.

Im Einzelnen:

Zu Beginn des Programms geben wir in Variable #2 den Durchmesser der Bohrungen und in Variable #3 die Tiefe an. Da alle Löcher gleich sind, ändern wir diese Werte im Programm nicht.

Wir beginnen mit dem ersten Loch in der linken unteren Ecke. X=10, Y=10 und deshalb setzen wir die Variable #0=10 und #1=10. Wir haben wieder einen neuen Befehl. Das

Unterprogramm. Mit `M98 P1000` rufen wir das Unterprogramm mit der Nummer 1000 auf.

Das Unterprogramm ist genau das Programm, das wir zuvor geschrieben haben. Nur, dass es nicht mehr notwendig ist, den Variablen Werte zuzuweisen, weil wir dies bereits im Hauptprogramm getan haben.

Ein Unterprogramm hat diese Form:

`O1000` => O sagt, dass es sich um ein Unterprogramm handelt, 1000 ist eine Identifikationsnummer, die wir nach Belieben wählen können.

`M99` => Es ist der Rücksprung zum Hauptprogramm, das Ende des Unterprogramms.

In Zeile `M98 O1000` springen wir direkt in die erste Zeile des Unterprogramms `G0 X=#0 Y=#1` und wenn wir zur Zeile `M99` kommen, springen wir zum Hauptprogramm in der Zeile nach dem Unterprogrammaufruf, die `#0=80`.

Es ist wichtig, dass wir das Hauptprogramm abschließen mit `M2` o `M30`, **`damit wir nicht ungewollter Weise in das Unterprogramm geraten nachdem wir das Hauptprogramm beendet haben.`**

Was sind die Vorteile dieser Art der Programmierung?

1. Wie wir sehen können, ist es im Hauptprogramm viel einfacher zu programmieren.
2. Die Unterprogramme können für das nächste Mal gespeichert werden.
3. Wenn Sie das Werkzeug wechseln, passiert nichts - die Radiuskorrektur wird automatisch durchgeführt.
4. Wenn wir größere/kleinere Löcher benötigen, ändern wir nur eine Zeile: #2=4.2 => #2=5.0 zum Beispiel.
5. Wenn wir mehr Löcher brauchen, müssen wir nur die Koordinaten der Zentren zum Hauptprogramm hinzufügen.

Index - Kompendium

G-Code-Sprachelemente

- % Start des Programms. Alles, was vorhergeht, wird als Kommentar behandelt
- (öffnet einen Kommentar
-) einen Kommentar schließen
- "" String
- 0.1234 Zahl
- N Zeilennummer
- O Nummer eines Unterprogramms
- P Programm- oder Parameternummer
- F Arbeitsvorschub
- S Spindeldrehzahl
- T Werkzeugnummer
- D Radius-Korrekturparameter
- H Wartezeit
- A Drehachse parallel zu X
- B Drehachse parallel zu Y
- C Drehachse parallel zu Z
- U zweite X-Achse
- V zweite Y-Achse
- W zweite Achse Z
- X X-Achse
- Y Y-Achse
- Z Z-Achse
- I Parameter
- J Parameter
- K Parameter
- R Parameter

//Code G ohne Bewegung

- G17-19 Auswahl der Arbeitsebene (G17=XY, G18=XZ, G19=YZ)
- G21 Metrische Koordinaten (mm)
- G40 Radiuskorrektur deaktivieren
- G41 Radiuskorrektur (rechts)
- G42 Radiuskorrektur (linke Hand)
- G43 Längenausgleich aktivieren
- G49 Werkzeuglängenkorrektur deaktivieren
- G50 Skalierung deaktivieren
- G53 Maschinenkoordinaten aktivieren
- G54-59 aktiviere Offset G54-G59)

- G61 Exakter Stop
- G62,G64 Modus konstante Geschwindigkeit (CV)
- G69 Koordinatensystem drehen
- G92

//C Bewegung G-Code

- G0 schnelle Positionierung (Eilgang)
- G1 Lineare Interpolation (Arbeitsvorschub)
- G2 Kreisinterpolation im Uhrzeigersinn
- G3 Kreisinterpolation gegen den Uhrzeigersinn
- G4 Pause (in s)
- G52 Zum Referenzpunkt fahren (wie G74)
- G73 Bohren mit Spanbrecher
- G74 Zum Referenzpunkt fahren (wie G52)
- G79 Werkzeuglängenmessung
- G81 Einfaches Bohren
- G82 Bohren mit Verweilzeit am Boden
- G83-Tiefbohrungen (in mehreren Schritten)
- G84-Gewinde (mit Gewindebohrer)
- G85 Reiben
- G86-Bohrzyklus
- G87 Festzyklus Rechtecktasche
- G88 Zyklus Nr. 8
- G89 Zyklus Nr. 9

//M Steuern des Programmablaufs

- M1 Programmierter Stopp (warten, fortsetzen)
- M2 Ende des Programms (alles aus)
- M30 Programmende (alles aus und zurückspulen)
- M98 Unterprogramm (P=Zahl)
- M99 Rückkehr aus Unterprogramm

//M Hilfsfunktionen, um etwas in der Maschine zu aktivieren

- M3 Spindel im Uhrzeigersinn drehen
- M4 Spindel gegen den Uhrzeigersinn drehen
- M5 Spindel stoppen
- M6 Automatischer Werkzeugwechsel
- M7 aktivieren Nebelkühlung
- M8 Flüssigkeitskühlung aktivieren
- M9 Kühlung deaktivieren
- M10 offene Klemm- oder Sperrspindel oder Drehachse fixieren/sperren
- M11 Schraubstock schließen
- M14 Tangentialmesser einstechen (Pen down)

- M15 Tangentialmesser ausheben (Pen up)
- M41 Getriebe Stufe 1
- M42 Getriebe Stufe 2
- M43 Getriebe Stufe 3
- M44 Getriebe Stufe 4
- M60 Teilewechsler (automatischer Teilewechsler)
- M66 Manueller Werkzeugwechsel
- +-*/= Operationen
- < Operator "kleiner als"
- > Operator "größer als"
- & Operator "Und"
- ! Operator negieren
- | Operator "oder"
- (
-)
- # Variable
- #I Eingang
- #O Ausgang
- L51
- L52
- L53

//Makro-Sprache Schlüsselwörter

- SQRT
- SIN
- COS
- IF
- THEN
- ELSE
- ENDIF
- PRINT
- RETURN
- CALL
- QUITE
- SKIP
- REWIND
- REPEAT
- NEXT
- UNTIL
- ASKBOOL
- ASKINT

- ASKFLT

Inhaltsverzeichnis