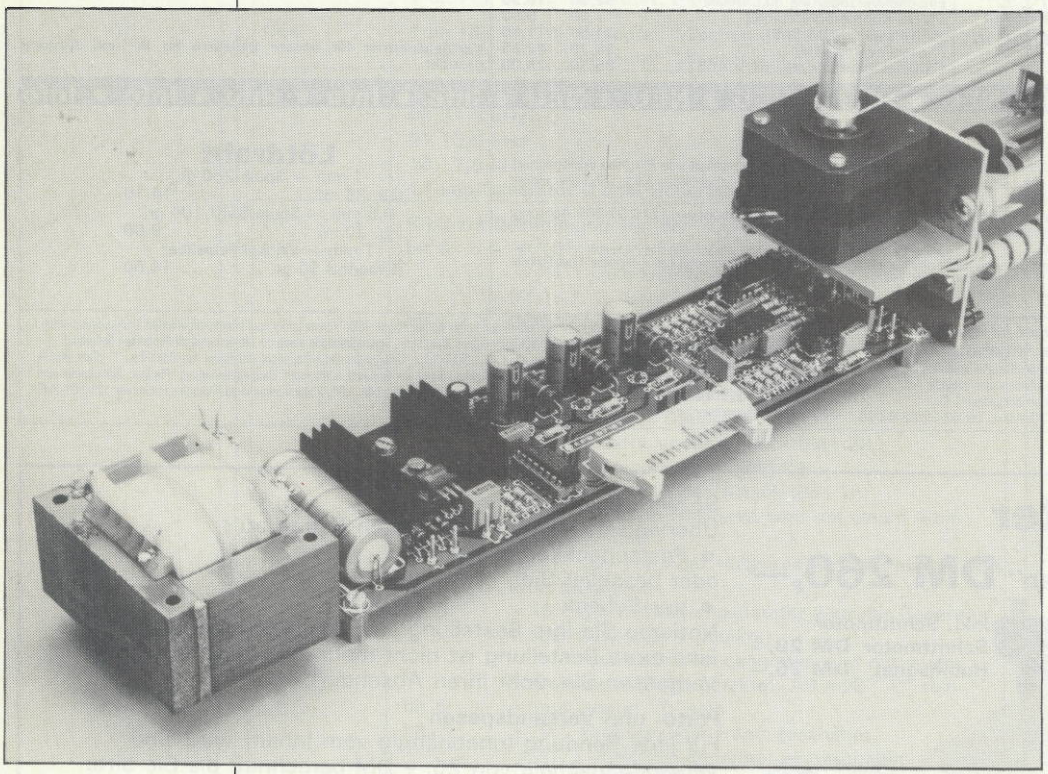
SELBSTBAUPLOTTER MONDRIAN

Programmierung

Zum Plotter gehört natürlich eine Steuerung, die je nach Computer unterschiedlich sein wird. Damit Sie den Plotter MONDRIAN mit Ihrem Computer zum Leben erwecken können, haben wir hier die Algorithmen und Flußdiagramme angegeben, die zur Programmierung notwendig sind.



10

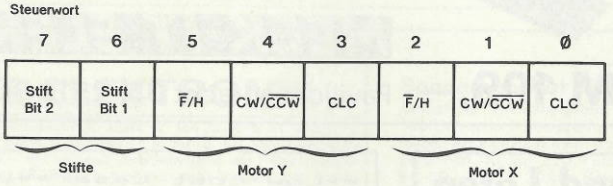


Bild 10. Aufbau des Steuerworts, mit dem der Plotter bedient werden muß: drei Bits pro Schrittmotor und zwei Bits für die Stiftewahl.

Ein fix und fertiges Rezept für die Plotter-Steuerung, etwa in Form eines Programms für den Computer Stamiga von Commotari, gibt es nicht. Der Plotter ist dumm, er muß bitweise vom Computer gefüttert werden, um zu wissen, was er tun soll. Damit die Sache ausreichend schnell abläuft, muß ein Teil des Programms in Maschinensprache geschrieben werden, denn die meisten Heimcomputer arbeiten mit einem BASIC-Interpreter, der Zeile für Zeile übersetzt.

Das Steuerwort ist so, wie in Bild 10 skizziert, aufgebaut. Ein Schrittmotor führt bei jeder positiven Flanke des Signals am Takteingang einen oder einen halben Schritt aus, je nach dem Pegel am entsprechenden Eingang. Der Taktimpuls muß mindestens 10 μs lang ''1'' bleiben. Wenn nur der X-Motor *oder* der Y-Motor in Betrieb ist, werden gerade Linien gezeichnet. Laufen beide Motoren zur gleichen Zeit, dann zeichnet der Plotter Linien in einem Winkel von 45°. Man kann einen Motor auch im Halbschrittbetrieb arbeiten lassen, dann laufen die Linien unter den Winkeln 26°34' oder 63°27' (tan 26°34' = 0,5 beziehungsweise tan 63°27' = 2).

Wir geben als Anregung für ein Steuerprogramm einige Routinen und Algorithmen an. Sie sind die Grundbausteine eines allgemeinen Plotterprogramms. Andere Lösungen sind selbstverständlich möglich.

Basisroutine

Wie der Name schon sagt, sorgt dieses Programm für die Grundfunktionen des Plotters. Je nach Steuerwort (Bild 10), das als 8-bit-Information am Ausgang liegt, wird ein ganzer oder ein halber Schritt in X- und/oder Y-Richtung ausgeführt, und/oder ein bestimmter Stift gewählt. Welcher Motor sich bewegt, hängt von der Information in Bit 0 und in Bit 3 ab. Der Motor führt einen Schritt aus, wenn das zugehörige Bit ''0'' ist. Drehrichtung und Voll- oder Halbschrittbetrieb werden mit den anderen vier Bits eingestellt.

Im Flußdiagramm (Bild 12) erkennt man, daß das Steuerwort zunächst am Ausgangs-Port liegt, an den der Plotter angeschlossen ist. Damit werden Voll- oder Halbschrittbetrieb und die gewünschte Drehrichtung eingestellt, und der Takteingang wird auf Null gelegt. Anschließend schaut der Plotter in den Bits 0 und 3 nach, welcher Motor einen Schritt ausführen soll.

Die aktuellen Koordinaten, also die aktuellen Motor-Positionen, werden in zwei 16-bit-Zählern gespeichert, die auch die eingestellte Richtung und die Information über Voll- oder Halbschrittbetrieb enthalten.

Anschließend werden die Bits 0 und 3 auf ''1'' gesetzt, und das Steuerwort wird nochmal an den Ausgangs-Port gelegt. Danach füh-

ren die gewählten Motoren einen Schritt aus.
Mit den beiden höchstwertigen Bits können die Stifte gewählt und alle Stifte hochgestellt werden. Da in den meisten Fällen dabei kein Schritt ausgeführt werden soll, müssen die Bits 0 und 3 in diesem Fall auf ''1'' gesetzt werden.
Die Überprüfung eines Bits kommt in der Basisroutine oft vor. Der Z80 kennt bestimmte Anweisungen dafür. Bei anderen Prozessoren ist folgendes Verfahren möglich: Das Byte, das das zu prüfende Steuerwort-Bit enthält, wird auf ''1'' gesetzt und UND-verknüpft (Rechteck unten rechts in Bild 12). Das Ergebnis kann man im Nullzeiger ablesen.

Einfache Linien und Stiftewahl

Mit einer kleinen Erweiterung der Basisroutine kann man ganz einfach beliebig lange Linien unter einem bestimmten Winkel zeichnen. Erst wird das Steuerwort eingestellt. Dabei müssen gleichzeitig die Bits für die Stiftewahl gesetzt werden. Die gewünschte Linienlänge kann man eine bestimmte X- oder Y-Koordinate binden. Nach jedem Schritt wird geprüft, ob der Zielpunkt erreicht ist. Ist das nicht der Fall, dann wird nach kurzer Verzögerung der folgende Schritt ausgeführt. Die Verzögerungszeit kann mit einer Programmschleife oder mit einem Timer im 6522 (VIA) oder im Z80-CTC erzeugt werden. Ein Timer hat den Vorteil, daß die resultierende Schrittfrequenz innerhalb gewisser Grenzen unabhängig vom Programmteil ist, der zwischen zwei Schritten abgearbeitet wird. Sie müssen also die Verzögerungszeit so austüfteln, daß die Motoren sowohl im Voll- als auch im Halbschrittbetrieb ruckfrei drehen. Als Verbesserung der Motorsteuerung kann man ein ''Feature'' einbauen, das die Motoren beim Starten und Stoppen gleichmäßig beschleunigt und bremst. Einerseits wird so die Gefahr vermindert, daß ein Motor Schritte ausläßt, und zum anderen werden Längsschwingungen des Wagens verhindert. Diese Längsschwingungen werden möglicherweise durch die Massenträgheit und die Dehnung der Antriebsschnur verursacht.
Dadurch daß ein Motor oder beide Motoren drehen, werden unterschiedliche Drehrichtungen und Voll- oder Halbschrittbetrieb miteinander verknüpft, und es ist mög-

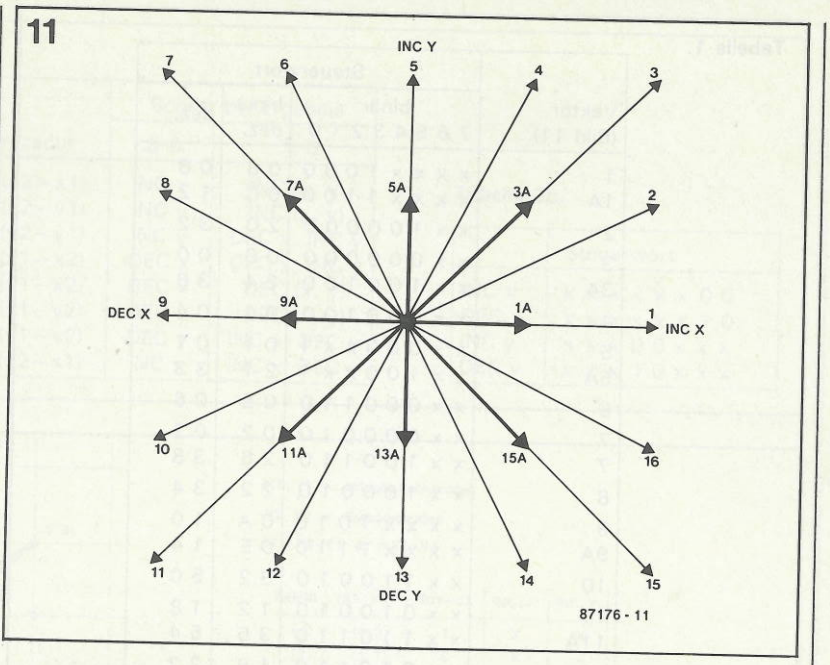


Bild 11. Gerade Linien unter bestimmten, festen Winkeln sind relativ leicht zu zeichnen. Die zugehörigen Steuerworte stehen in Tabelle 1.

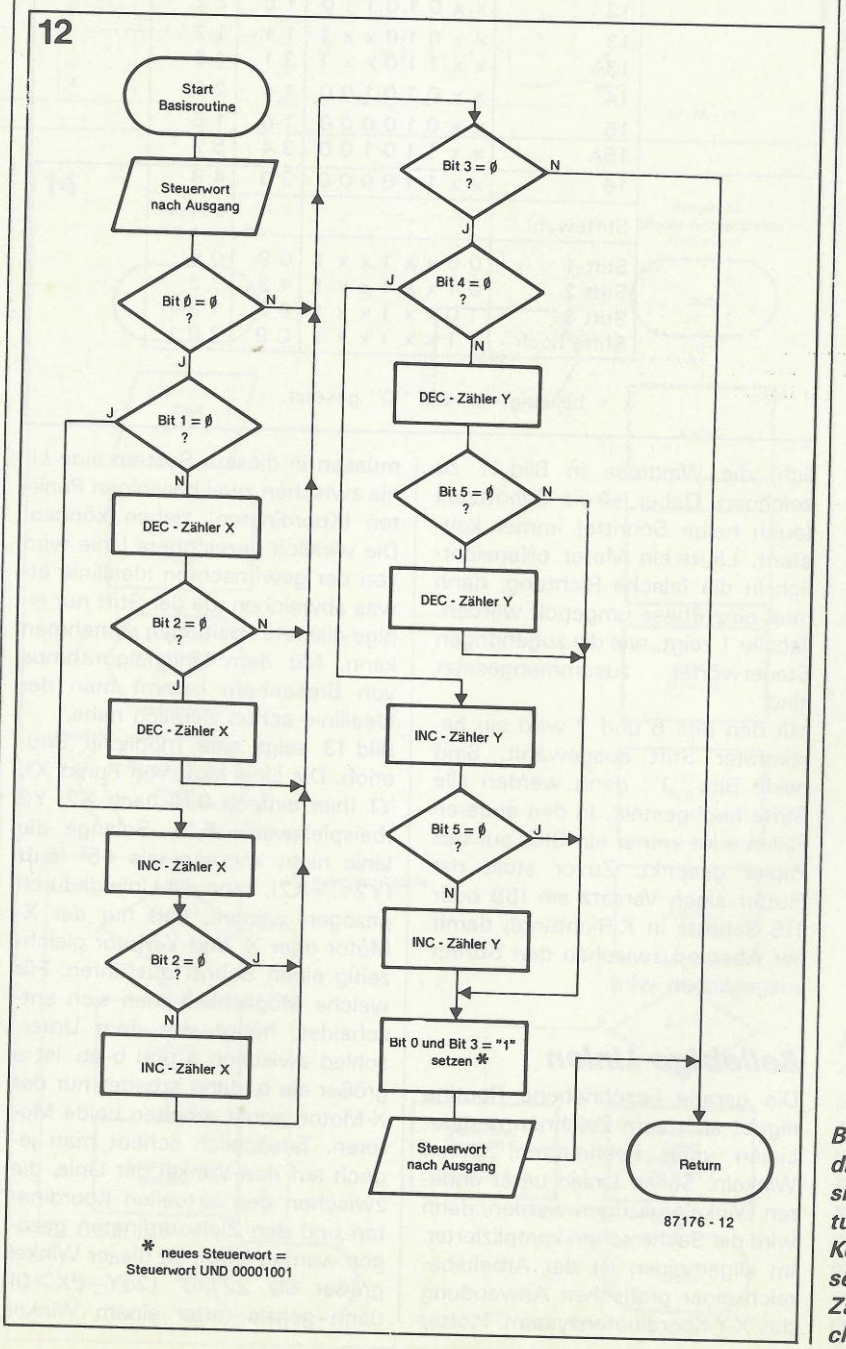


Bild 12. Das Flußdiagramm der Basisroutine. Die aktuellen X- und Y-Koordinaten müssen in 16-bit-Zählern gespeichert werden.

Tabelle 1.

Vektor (Bild 11)	Steuerwort binär 7 6 5 4 3 2 1 0	hexa-dez.	dez.
1	x x x x 1 0 0 0	0 8	0 8
1A	x x x x 1 1 0 0	0 C	1 2
2	x x 1 0 0 0 0 0	2 0	3 2
3	x x 0 0 0 0 0 0	0 0	0 0
3A	x x 1 0 0 1 0 0	2 4	3 6
4	x x 0 0 0 1 0 0	0 4	0 4
5	x x 0 0 0 x x 1	0 1	0 1
5A	x x 1 0 0 x x 1	2 1	3 3
6	x x 0 0 0 1 1 0	0 6	0 6
7	x x 0 0 0 0 1 0	0 2	0 2
7	x x 1 0 0 1 1 0	2 6	3 8
8	x x 1 0 0 0 1 0	2 2	3 4
9	x x x x 1 0 1 0	0 A	1 0
9A	x x x x 1 1 1 0	0 E	1 4
10	x x 1 1 0 0 1 0	3 2	5 0
11	x x 0 1 0 0 1 0	1 2	1 8
11A	x x 1 1 0 1 1 0	3 6	5 4
12	x x 0 1 0 1 1 0	1 6	2 2
13	x x 0 1 0 x x 1	1 1	1 7
13A	x x 1 1 0 x x 1	3 1	4 9
14	x x 0 1 0 1 0 0	1 4	2 0
15	x x 0 1 0 0 0 0	1 0	1 6
15A	x x 1 1 0 1 0 0	3 4	5 2
16	x x 1 1 0 0 0 0	3 0	4 8
Stiftewahl			
Stift 1	0 0 x x 1 x x 1	0 9	0 9
Stift 2	0 1 x x 1 x x 1	4 9	7 3
Stift 3	1 0 x x 1 x x 1	8 9	1 3 7
Stifte hoch	1 1 x x 1 x x 1	C 9	2 0 1

x = beliebig, hier auf ''0'' gesetzt.

Tabelle 1. Aufbau des Steuerworts für die Zeichnung in Bild 11.

lich, die Windrose in Bild 11 zu zeichnen. Dabei ist die Schrittzahl (auch halbe Schritte) immer konstant. Läuft ein Motor offensichtlich in die falsche Richtung, dann muß eine Phase umgepolt werden. Tabelle 1 zeigt, wie die zugehörigen Steuerwörter zusammengesetzt sind.

Mit den Bits 6 und 7 wird ein bestimmter Stift ausgewählt. Sind beide Bits ''1'', dann werden alle Stifte hochgestellt. In den anderen Fällen wird immer ein Stift auf das Papier gesenkt. Zuvor stellt der Plotter einen Versatz ein (58 oder 116 Schritte in X-Richtung), damit der Abstand zwischen den Stiften ausgeglichen wird.

Beliebige Linien

Die gerade beschriebene Routine eignet sich zum Zeichnen gerader Linien unter bestimmten, festen Winkeln. Sollen Linien unter anderen Winkeln gezogen werden, dann wird die Sache schon komplizierter. Im allgemeinen ist der Arbeitsbereich einer grafischen Anwendung das X-Y-Koordinatensystem. Plotter müssen in diesem System eine Linie zwischen zwei beliebigen Punkten (Koordinaten) ziehen können. Die wirklich gezeichnete Linie wird von der gewünschten Ideallinie etwas abweichen, da der Stift nur einige diskrete Positionen einnehmen kann. Mit dem Linienalgorithmus von Bresenham kommt man der Ideallinie schon ziemlich nahe.

Bild 13 zeigt eine mögliche Situation. Die Linie läuft von Punkt X1, Y1 (hier einfach 0,0) nach X2, Y2 (beispielsweise 5,3). Solange die Linie nicht schräger als 45° läuft ($Y2 <= X2$), kann die Linie dadurch gezogen werden, daß nur der X-Motor oder X- und Y-Motor gleichzeitig einen Schritt ausführen. Für welche Möglichkeit man sich entscheidet, hängt von dem Unterschied zwischen a und b ab. Ist a größer als b, dann arbeitet nur der X-Motor, sonst arbeiten beide Motoren. Tatsächlich schaut man jedoch auf den Winkel der Linie, die zwischen den aktuellen Koordinaten und den Zielkoordinaten gezogen werden kann. Ist dieser Winkel größer als 22°30' ($2dY - dX > 0$) dann geht's unter einem Winkel von 45° zum folgenden Punkt (X+1, Y+1). Im anderen Fall arbeitet nur der X-Motor. Das Angenehme an diesem Algorithmus ist die einfache Berechnung der Entscheidungsprozedur: dX und dY werden mit einer Subtraktion bestimmt, und die Multiplikation mit 2 geschieht auf der Maschinensprache-Ebene dadurch, daß die Zahl einmal nach links geschoben wird.

Für Linien unter Winkeln zwischen 45° und 90° gilt derselbe Algorithmus, nur müssen X und Y getauscht werden. Linien in den übrigen drei Quadranten können genau so einfach gezogen werden, allerdings muß vorher festgelegt werden, in welchem Oktanten (halber Quadrant) der Zielpunkt in Bezug auf den Startpunkt liegt.

Das Flußdiagramm in Bild 14 zeigt, wie Linien nach dem beschriebenen Verfahren zwischen beliebigen Punkten gezogen werden können. Zunächst wird also festgelegt, in welchem Oktanten der Zielpunkt in Bezug auf den Startpunkt liegt. Davon abhängig wird ein Zeiger gesetzt, der auf eine der zugehörigen acht Entscheidungsprozeduren zeigt (Tabelle 2). In der Entscheidungsprozedur wird immer das Steuerwort eingestellt, also welche Motoren oder welcher Motor in einem bestimmten Augenblick einen Schritt in eine bestimmte Richtung machen muß.

Zur endgültigen Ausführung der Schritte wird die Basisroutine aufgerufen. Nach jedem Schritt werden die aktuellen Koordinaten X1, Y1 mit den Zielkoordinaten X2, Y2 verglichen, und wenn sie gleich sind, wird der Algorithmus verlassen.

Kreise und Ellipsen

Eine häufig vorkommendes Muster ist der Kreis. Stellt man eine Tabelle mit einer Periode einer Sinusfunktion (gerundet auf ganze Zahlen) neben eine Tabelle mit einer Cosinusfunktion, dann kann das Koordinatensystem eines Kreises damit berechnet werden. Die erste Tabelle enthält nämlich die X-Koordinaten und die zweite die Y-Koordinaten. Die Amplituden ergeben die Linie in die X- und in die Y-Richtung. Sind beide gleich groß, dann ist das Ergebnis ein sauberer Kreis. Im anderen Fall erhält man eine Ellipse mit der Längsachse parallel zur X- oder Y-Achse. Werden die Phasenbeziehungen der beiden Sinusfunktionen geändert (wechselseitiges Verschieben der beiden Tabellen),

Tabelle 2a

Oktant	Enscheidungsprozedur	Schritt bei Ergebnis ≦ 0	Schritt bei Ergebnis > 0
0 . . . 45°C	2 * (y2−y1) − (x2−x1)	INC x	INC (x, y)
45° . . . 90°	2 * (x2−x1) − (y2−y1)	INC y	INC (x, y)
90° . . . 135°	2 * (x1−x2) − (y2−y1)	INC y	DEC x, INC y
135° . . . 180°	2 * (y2−y1) − (x1−x2)	DEC x	DEC x, INC y
180° . . . 225°	2 * (y1−y2) − (x1−x2)	DEC x	DEC (x, y)
225° . . . 270°	2 * (x1−x2) − (y1−y2)	DEC y	DEC (x, y)
270° . . . 315°	2 * (x2−x1) − (y1−y2)	DEC y	INC x, DEC y
315° . . . 360°	2 * (y1−y2) − (x2−x1)	INC x	INC x, DEC y

Tabelle 2b.

	Steuerwort
INC x	x x x x x x 0 0
DEC x	x x x x x x 1 0
INC y	x x x 0 0 x x x
DEC y	x x x 1 0 x x x

Tabelle 2. Die verwendete Entscheidungsprozedur hängt davon ab, in welchem Oktanten der Zielpunkt in Bezug auf den Startpunkt liegt.

13

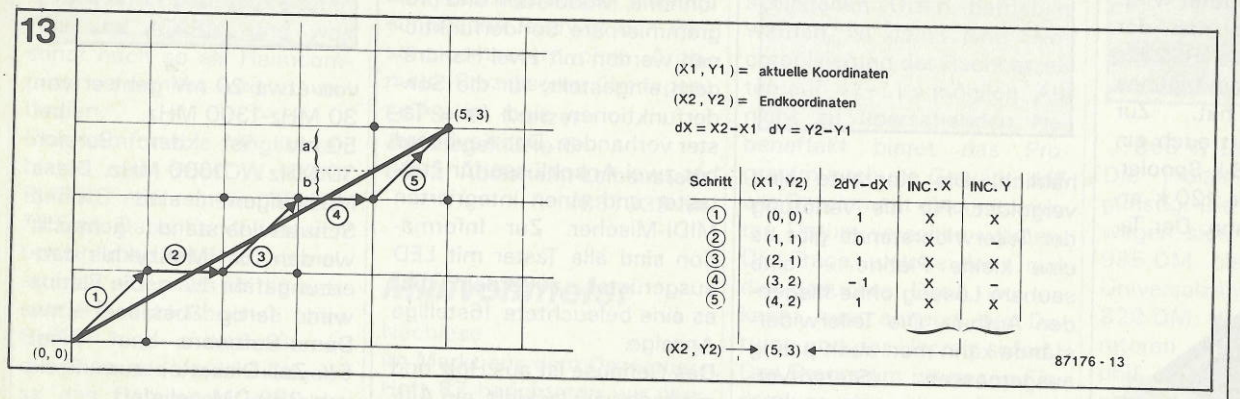


Bild 13. Der Bresenham-Linienalgorithmus. Die Ideallinie ist fett gezeichnet. Die Punkte im Raster stehen für die diskreten Positionen, die der Stift einnehmen kann. Der Unterschied zwischen a und b legt fest, ob der folgende Schritt in X-Y- oder in X- und Y-Richtung ausgeführt wird.

dann entstehen Ellipsen, die schräg im Achsensystem liegen.
Da die Berechnung der Koordinaten relativ lange dauert, wird sie im voraus erledigt. Das Ergebnis (die beiden Tabellen) wird gespeichert. Der Kreis kann nun dadurch gezeichnet werden, daß der Plotter von Punkt zu Punkt geht, wobei wieder der Bresenham-Algorithmus verwendet wird.

Erweiterung

Mit der zuerst beschriebenen Basisroutine und einem allgemeinen Algorithmus kann man ein universelles Steuerprogramm aufstellen. Man kann es in einer höheren Programmiersprache schreiben, da für die zeitkritischen Teile doch wieder die schon erwähnten (Maschinensprache-)Routinen aufgerufen werden. Mit einem solchen Programm können Linien zwischen beliebigen Punkten (absolut), Linien zwischen der aktuellen Stiftposition und einen in Bezug auf diesen Punkt definierten Punkt (relativ), Standardmuster wie Kreise und Rechtecke und so weiter gezeichnet werden. Es ist natürlich auch nützlich, einen Zeichensatz für Texte in Bildern zur Verfügung zu haben. Jedem Zeichen müssen bestimmte (relative) Koordinaten zugeordnet sein. Durch Multiplikation dieser Koordinaten mit einem Faktor können die Zeichen vergrößert oder verkleinert werden.

14

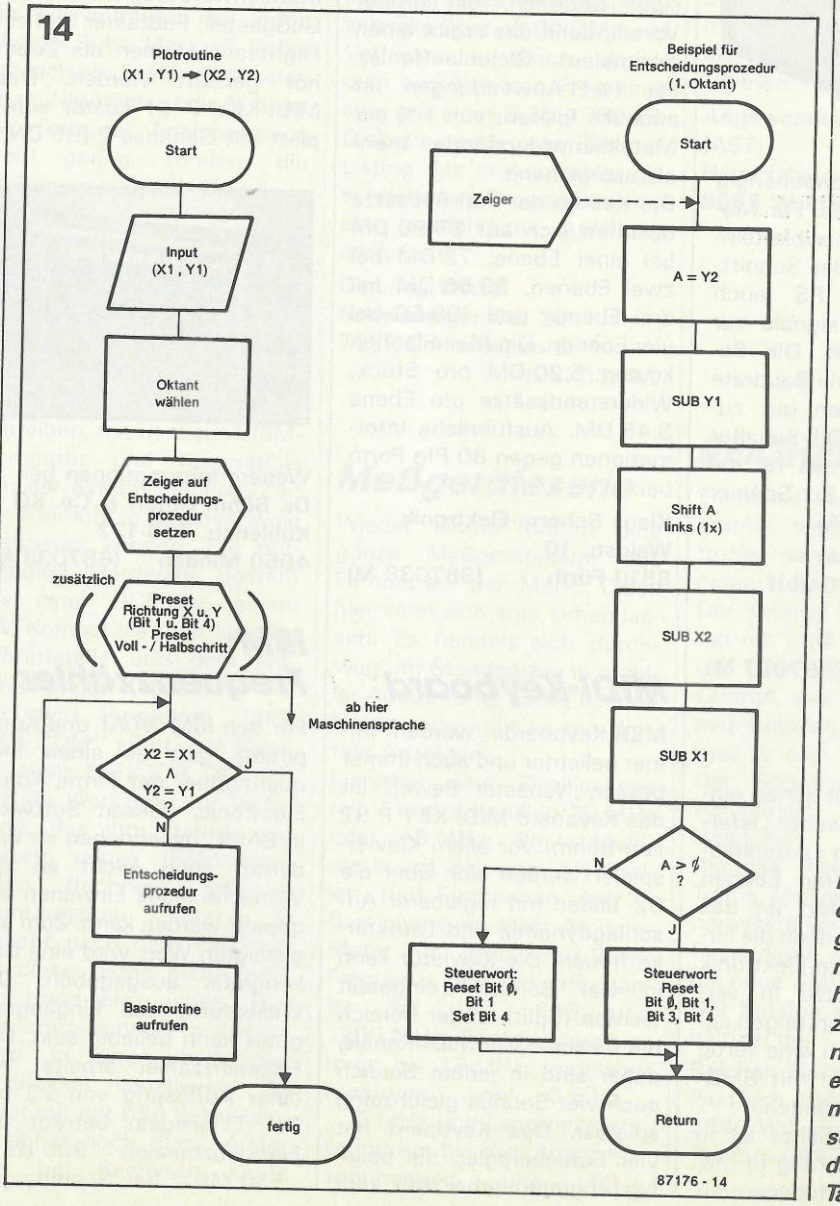


Bild 14. Ein mögliches Flußdiagramm, um Linien mit dem Bresenham-Algorithmus zeichnen zu können. Außerdem eine der acht möglichen Entscheidungsprozeduren (siehe auch Tabelle 2).