

Tiger und Maus

Gunther Zielosko

1. Der moderne Zeigestock - die Computermaus

In dieser Applikation werden wir feststellen, daß sich unter besonderen Bedingungen Katzen (Tiger) und Mäuse fabelhaft verstehen können.

Als bei Microsoft die Maus erfunden wurde, konnte keiner ahnen, daß ein paar Jahre später ein Computer kaum noch "ohne" denkbar ist. Oder kennen Sie alle "hot-keys", um ein Windows-Programm ordentlich zu bedienen? Wenn man sich aber nun so an die kleinen "Haustierchen" gewöhnt hat, liegt es nahe, sie auch für den BASIC-Tiger® einzusetzen. Natürlich wird es uns nicht gelingen, Windows mit dem BASIC-Tiger® zu kopieren, trotzdem kommen wir mit Tiger, Maus und ggf. Graphikdisplay in die Nähe. Einige Funktionen der Maus können dabei auch für typische BASIC-Tiger®-Anwendungen interessant sein - alles das werden wir versuchen.

Wichtiger Hinweis!

Dieser Applikationsbericht beinhaltet die Ansteuerung einer seriellen Microsoft-Maus Version 2.0a, andere (evtl. auch sogenannte "kompatible" Mäuse) wurden hier nicht erprobt. Es ist möglich, daß solche Mäuse aus Gründen des Datenformates, der Kommunikationsparameter, des Tiger-Basic Treibers usw. nicht funktionieren.

Was eigentlich tut eine Maus? Fangen wir mit der Maus am PC an, dort haben wir zunächst zwei oder mehr Tasten, die im allgemeinen zur Übernahme bestimmter Größen oder Funktionen benutzt werden. Welche Funktionen das sind, wird meist durch die Position des Mauszeigers auf der Bildschirmoberfläche und das Programm festgelegt. Die Position des Mauszeigers wird durch die Bewegung der Maus gesteuert. Diese hat eine gummierte Kugel (damit es besser rollt), die beide Bewegungsrichtungen (x und y) auf zwei Rollen weitergibt (Bilder 1 und 2).

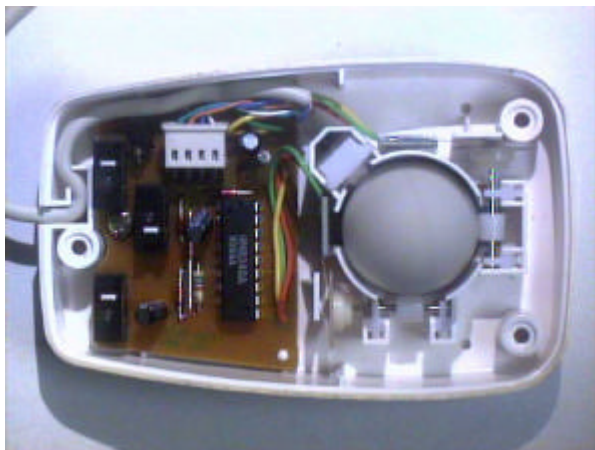


Bild 1 Innenleben einer Maus



Bild 2 Abtastung mit segmentierter Scheibe

An diesen Rollen befinden sich je nach Konstruktion Lochscheiben (für optische Abtastung) oder Scheiben mit metallischen Segmenten (für Schleifkontaktabtastung, siehe Bild 2). Bei Bewegung wird die Anzahl der vorbeilaufenden Löcher (oder Kontaktsegmente) gezählt, dabei kann die Mauselektronik mit einem genialen Trick selbständig feststellen, ob sie sich vorwärts oder rückwärts bewegt. Dadurch läuft das Ganze auf eine Streckenmessung hinaus. In bestimmten zeitlichen Abständen wird die Anzahl der "Schritte", ihre Richtung und der Zustand der Tasten seriell ausgegeben. Früher erfolgte dies fast ausschließlich über die serielle Schnittstelle des PC (da der BASIC-Tiger[®] auch eine solche hat, ist diese Sorte Maus für uns die richtige Wahl), mittlerweile gibt es auch Mäuse für PS-2- und USB-Anschluß, solche Varianten sind für unsere Experimente nicht geeignet. Wer sich eine solche neue Maus beschafft hat, kann die alte serielle Maus bestimmt entbehren und sie (ggf. nach einer gründlichen Reinigung) für BASIC-Tiger[®]-Anwendungen nutzen. Damit wir verstehen, was im Zusammenspiel von Maus und PC abläuft, müssen uns das Daten-Übertragungs-Protokoll etwas näher ansehen. Vorher sind noch zwei Hardwareprobleme zu klären um sicherzustellen, daß überhaupt ein Bit von der Maus in den BASIC-Tiger[®] gelangt.

2. Pegelfragen

Bisher sind wir wie selbstverständlich davon ausgegangen, daß wir am BASIC-Tiger[®] zwei serielle Schnittstellen haben. Das ist allerdings nicht in jedem Falle so, die Firma Wilke liefert ihre Tiger sowohl mit echtem RS232-Schnittstellen-Pegel als auch mit TTL-Pegel aus. Das Timing und die Logik ist bei beiden gleich, allerdings unterscheiden sich die Pegel grundsätzlich. Im Plug-and-Play-Lab spielt das keine Rolle, dort wurde hardwaremäßig dafür gesorgt, daß sich beide Varianten des BASIC-Tiger[®] exakt gleich verhalten. Deshalb sind für die, die mit dem Plug-and-Play-Lab weiterexperimentieren wollen, die folgenden Überlegungen nicht wichtig, wohl aber für die, die eigene Hardware aufbauen wollen. Die Bilder 3 und 4 zeigen zunächst die Pegelverhältnisse der beiden BASIC-Tiger[®]-Ausführungen:

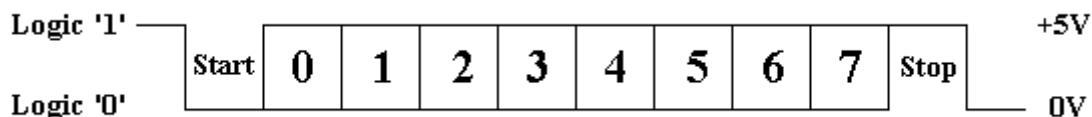


Bild 3 TTL-Pegel an Modulen ohne echte RS232-Schnittstellen

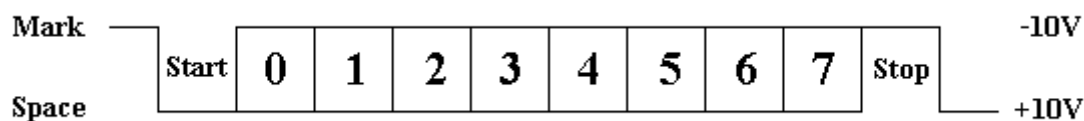


Bild 4 RS232-Pegel an Modulen mit echten RS232-Schnittstellen

Und was haben Sie? Falls Sie "nur" über ein TTL-Modul verfügen, brauchen Sie trotzdem nicht aufzugeben, es gibt eine Alternative. MAXIM baut ein IC, welches den einen Pegel auf einfachste Weise in den anderen umwandelt, ja sogar die höheren +10 V und vor allem die negativen -10 V werden vom IC selbst erzeugt. Nur ein paar Elkos sind nötig und die RS232-Schnittstelle ist fertig (übrigens macht es die Firma Wilke in ihren Modulen und im Plug-and-Play-Lab genauso!). Dieser IC, der MAX232, kann im übrigen für alle Fälle genutzt werden, in denen eine echte RS232-Schnittstelle erforderlich ist, so daß der Aufwand nicht nur für unsere Maus getrieben werden muß. Einen kleinen Wermutstropfen gibt es trotzdem, denn der MAX232 negiert den Pegel noch logisch. So ist es erforderlich, auf jeder TTL-Seite noch einen Inverter vorzusehen (ggf. können Sie das auch mit der Software erledigen...). Mit dieser Schaltung sollte es klappen, übrigens haben Sie die Schaltungsunterlagen mit dem Plug-and-Play-Lab evtl. schon mitgeliefert bekommen. Lassen Sie dort einfach alles weg, was zum Betrieb der Leuchtdioden dient. Die Beschaltung des MAX232 zeigt Bild 5, wie man sieht, kann der IC sogar zwei einfache serielle Schnittstellen bedienen.

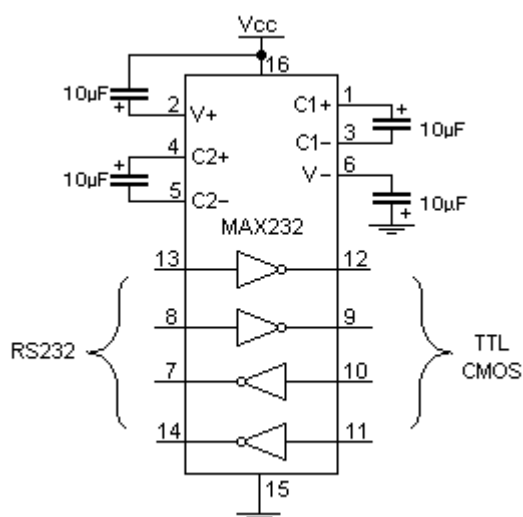


Bild 5 Beschaltung des MAX 232

3. Stromversorgung der Maus

Wenn man sich die Belegung der seriellen Schnittstelle 0 am Plug-and-Play-Lab anschaut, stellt man fest, daß dort folgende Leitungen angeschlossen sind: RxD0, TxD0, CTS0, RTS0 und GND. Diese Anschlüsse sind reine Signalleitungen, eine Batterie gibt es in der Maus auch nicht. Wo also nimmt die Maus ihre "Energie" her? Am PC, wo die Maus ja eigentlich hingehört, gibt es zwar ein paar weitere Leitungen an der seriellen Schnittstelle, aber Vcc oder irgendeine andere Powerleitung finden wir auch hier nicht. Das Geheimnis besteht darin, daß das Pegelspiel der einzelnen Signalleitungen über Gleichrichter zur Spannungsversorgung der Maus "mißbraucht" wird. Mit dem Plug-and-Play-Lab geht das eventuell nicht so elegant, da die vom MAX232 bereitgestellten Spannungen zusätzlich von anderen Komponenten "angezapft" werden (LED's usw.). Für die seriellen Schnittstelle am Plug-and-Play-Lab

benötigen wir deshalb u.U. einen kleinen Adapter, mit dem wir der Maus zusätzlich +5 V von einem Stiftkontakt des Plug-and-Play-Labs zuführen. Im übrigen ist dies kaum zusätzlicher Aufwand, ein Adapter ist immer erforderlich, weil Maus und Plug-and-Play-Lab jeweils 9-polige Sub-D-Buchsen haben. Wir verbinden also zwei 9-polige Sub-D-Stecker mit kurzen Drähten so, wie das Bild 6 zeigt. Beachten Sie, daß die Pins 2 und 3 hier "überkreuzt" werden müssen (jeweils Pin 2 an Pin 3!).

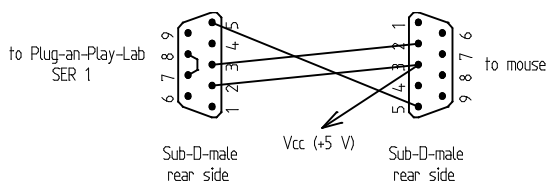


Bild 6 Einfacher Maus-Adapter für die SER 0 - Schnittstelle

4. Das Microsoft-Serien-Maus-Format

Die serielle Datenübertragung von der Maus zum PC erfolgt mit 1200 Baud, ohne Parität, mit 2 Stopbits. Wird mit der Maus nichts gemacht, erfolgt keine serielle Übertragung, jede Veränderung (Taste gedrückt oder losgelassen, Bewegung in X- oder Y-Richtung) hat die Übertragung von 3 Bytes zur Folge. Die Reihenfolge der Übertragung ist Byte 1 Bit 0 nach Byte 3 Bit 7, die folgende Tabelle zeigt die Bedeutung der einzelnen Bits dieser 3 Bytes:

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
Byte 1	0 (1)*	1	L	R	Y7	Y6	X7	X6
Byte 2	0 (1)*	0	X5	X4	X3	X2	X1	X0
Byte 3	0 (1)*	0	Y5	Y4	Y3	Y2	Y1	Y0

Tabelle 1 Datenformat der Microsoft-Serien-Maus

Dabei ist

L	Status der linken Taste	1 für gedrückt, 0 für nicht gedrückt
R	Status der rechten Taste	1 für gedrückt, 0 für nicht gedrückt
X0-X7	X-Achsen-Bewegungsdaten	vorzeichenbehaftete Binärwerte
Y0-Y7	Y-Achsen-Bewegungsdaten	vorzeichenbehaftete Binärwerte

* manche Mäuse erzeugen dort eine "1"

Kryptischer hätte die Aufteilung der einzelnen Bits nicht sein können, dennoch wird es der BASIC-Tiger® schaffen, die verschiedenen Einzelinformationen in nutzbare Daten umzusetzen.

5. Etwas Datenverarbeitung

Wie wir gesehen haben, überträgt die Maus, wenn überhaupt, immer 3 Bytes. Das kann sehr schnell hintereinander erfolgen. Deshalb ist es zunächst erforderlich, das erste Byte eines solchen "Paketes" sicher zu erkennen. Dieses unterscheidet sich von den anderen durch eine logische "1" in der zweithöchsten Positionen, was mit Tiger-Basic relativ einfach getestet werden kann.

Aus Tabelle 1 können sofort zwei Informationen gewonnen werden, nämlich der Status der beiden Tasten. Byte 1 enthält sie auf den Bitpositionen 4 (rechte Taste) und 5 (linke Taste). Das Programm kann dann das jeweilige Bit testen und daraus ableiten, ob und welche Taste gedrückt wird. Hier muß man etwas aufpassen, denn auch beim Loslassen werden wieder 3 Bytes übertragen, diesmal mit einer 0 auf der betreffenden Bitposition.

Bei den Daten der Mausposition (X- und Y-Richtung) wird es ein wenig komplizierter, da diese Werte auf verschiedene Bytes verteilt sind. Fangen wir mit der Y-Information an. Die höherwertigen Bits Y7 und Y6 befinden sich im ersten Byte. Mit einer BIT_AND 00001100 Anweisung werden sie zunächst separiert und dann mit einem Bitverschiebefehl 4x nach links an die Position verschoben, wo sie hingehören. Dann wird Byte 3 wieder mit einem BIT_AND 00111111 Befehl bearbeitet, daß nur die Bits 5 bis 0 von Y übrigbleiben. Diese stehen bereits an der richtigen Position. Schließlich werden die jetzt separierten Bits von Byte 1 und 3 durch Addition zu einer Zahl zusammengefaßt, die dann die Y-Position in einem Byte enthält. Analog wird mit der X-Information verfahren. Im ersten Byte wird mit Separation und einer 6-maligen Bitverschiebung nach links der höherwertige Teil und im zweiten Byte mit einer Separation der niederwertige Teil gewonnen. Auch hier wird mit einer Addition der beiden Teile die komplette X-Position ermittelt.

Was ist nun aber mit diesen Daten? Ist das eigentlich die Mausposition? Nein!

Wie die Daten zu interpretieren sind, wollen wir uns im folgenden ansehen.

Die X- und Y-Werte sind Bytes, also Werte von 00000000 bis 11111111. Im Falle der Maus werden Werte mit einer 0 auf der Position von Bit 7 als positive Werte und solche mit einer 1 auf Position von Bit 7 als negative Werte behandelt, die Darstellung am Zahlenstrahl macht das deutlich:

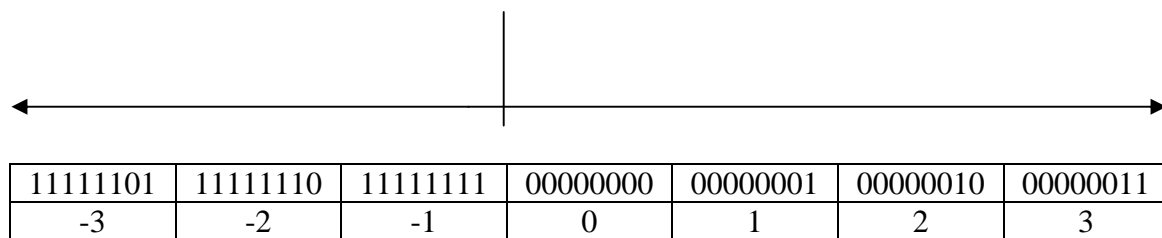


Bild 7 Interpretation binärer vorzeichenbehafteter Zahlen

Man sieht, daß das Bit 7 über das Vorzeichen der Zahl entscheidet. Die Werte mit positivem Vorzeichen werden wie bei binären Zahlen üblich umgesetzt. Wir erkennen aber, daß der positive Zahlenbereich nur bis 127 reicht. Werte größer oder gleich 128 werden schon als negative Werte erkannt. Bei diesen negativen Zahlen erfolgt die Umrechnung etwas anders. Ein Schritt von 0 aus rückwärts (also 11111111, in üblicher Umrechnung 255) wird nun als -1 interpretiert. Solch eine vorzeichenbehaftete Zahlendarstellung wird im übrigen häufig

benutzt. Für uns scheint es allerdings auf eine komplizierte Berechnung hinzusteuern. Aber der Aufwand hält sich in Grenzen, wie wir sehen werden. Genau das gewünschte Verhalten wird nämlich mit dem sogenannten 2-er Komplement erreicht. Im Falle Bit 7 = 1 kann der BASIC-Tiger[®] mit drei einfachen Befehlen (bitweise negieren, 1 addieren, mit -1 mutiplizieren) die rückwärts zu zählenden Schritte als negative Zahlen interpretieren. Im Falle Bit 7 = 0 wird nichts verändert und die Zahl einfach übernommen.

Tabelle 2 zeigt einige Beispiele für die Umrechnung vorzeichenbehafteter Binärzahlen:

Ausgangszahl	komplett negiert	1 addiert	Vorzeichen	Dezimalwert
00000000	→	→	+	0
00000001	→	→	+	1
01111111	→	→	+	127
11111111	00000000	00000001	-	1
11111110	00000001	00000010	-	2
10000000	01111111	10000000	-	128
10000011	01111100	01111101	-	125

Tabelle 2 Beispiele für vorzeichenbehaftete Binärzahlen

Aber auch das ist noch nicht die ganze Wahrheit. Nehmen wir an, die Maus wäre in ihrer Ausgangslage (z.B. nach einem RESET oder nach Drücken einer Taste hätten beide Positionen den Wert 0, abgelegt z.B. in je einer Variablen XPOS und YPOS). Dann erfolgt die erste Bewegung, was sofort als 3 Bytes über die serielle Schnittstelle eingelesen werden kann. Die X- und die Y-Position dieses ersten Datenpakets wird jeweils zu XPOS und YPOS addiert und gespeichert. Beim zweiten Datenpaket kommen wieder (relativ zur ersten!) Schritte in positiver bzw. negativer Richtung hinzu, die ebenfalls einfach addiert werden können und so fort. Zum Schluß erhält man die Position der Maus über eine Reihe von einzelnen Bewegungen in den beiden Variablen XPOS und YPOS. Fährt man zum Ausgangspunkt zurück, müßten X und Y wieder den Wert 0 erreichen (theoretisch!). Bei schnellen Bewegungen reagiert die Maus übrigens so, daß die letzten Datenpakete infolge der relativ langsamen Übertragung erst ankommen, wenn die Maus längst steht. Dafür gibt es in dem System Puffer, die auch solche schwierigen Bewegungsabläufe korrekt erfassen können. Im Gegensatz zur Maus am PC-Bildschirm bekommen wir im BASIC-Tiger[®] eine Absolutposition, die beim PC vom Bildschirmrand begrenzt ist, eine Weiterbewegung über den Bildschirmrand hinaus wird nicht mehr registriert. Wenn wir die Maus immer auf dem Boden lassen, können wir im BASIC-Tiger[®] mit praktisch unbegrenzten Zahlenwerten auch große Positionsänderungen erfassen.

Damit wir mit soviel Mathematik nicht ganz die Lust am Experimentieren mit der Maus verlieren, hat die Firma Wilke einen Treiber entwickelt, der fast die gesamte Steuerung und Umrechnung der Mauswerte übernimmt. In Kapitel 7 wird darauf noch eingegangen.

6. Wozu das Ganze

In diesem Abschnitt ein paar Gedanken zu Anwendungsmöglichkeiten der Tiger-Maus. Besonders beim Aufbau eigenständiger Geräte ohne Plug-and-Play-Lab werden wir dieses einfache Hilfsmittel schätzen lernen.

- Zunächst kann man die Maus, wie das Programm "MAUS02.TIG" zeigt, ganz normal wie am PC nutzen. Voraussetzung dafür ist natürlich ein Grafikdisplay.
- Dann haben wir ohne großen Aufwand 2 Tasten, die sich ohne die übliche Einbindung des Tastaturtreibers verwenden lassen.
- Oft ist der serielle Anschluß SER 0 nicht belegt, damit kann man auch bei knappen Portleitungen noch komfortable Bedienfunktionen unterbringen.
- Die eigentliche Stärke der Maus liegt in der Möglichkeit, Strecken und Standorte zu ermitteln. Das kann z.B. für Fahrmodelle nützlich sein, aber auch zum Messen benutzt werden.
- Interessant kann auch eine Menü-Steuerung über ein LC-Display sein, egal, ob nun alphanumerisch oder graphisch. Je nach Bewegungsrichtung scrollen Sie durch das Menü, eine Taste wird zur Übernahme des aktuellen Menüpunktes ("Enter") benutzt, die andere zum Verlassen des Menüs ("ESC").
- Nicht zuletzt können Werte feinfühlig eingestellt werden. Nehmen wir an, Sie benötigen in einem Programm einen Vorgabewert. Eine Bewegung der Maus kann Sie sehr schnell und genau zur gewünschten Zahl führen.
- Abschließend noch ein Tip. Man kann die Maus auch auseinandernehmen und die Abtastrollen ganz artfremd nutzen. So läßt sich z.B. ein "Meßschieber" oder ein Streckenmeßgerät für "krumme" Wege aufbauen. Denken Sie auch an die Positionsbestimmung für einen Mikroskop-Kreuztisch und ähnliche Anwendungen.

7. Software

Kernstück der Maussoftware ist der Treiber "SER1C_K1.TDD", der alle wichtigen Aufgaben für die Abfrage der Maus erledigt. Insbesondere die Einstellung der Parameter für die serielle Kommunikation und die Aufbereitung der "verschachtelten" Informationen der Maus (siehe Abschnitt 5) werden automatisiert.

Noch ein Hinweis! In der Software werden Sie eine BASIC-Instruktion finden, die (noch) nicht im Handbuch steht, die LIMIT-Anweisung. Diese ist wie folgt definiert:

```
var_a = LIMIT ( var_b, min_value, max_value )
```

Die Variable a, die aus der Variablen b errechnet wird, kann sich nur zwischen den beiden Werten Minimum und Maximum bewegen, wird sie größer oder kleiner, wird automatisch der jeweilige Grenzwert eingestellt.

Voraussetzung für die Funktion der folgenden Software ist, daß alle Dateien, die unter "MAUS.ZIP" diesem Applikationsbericht beigelegt sind, in Ihre Tigersoftware eingebunden werden:

- Fügen Sie die neuen .INC-Dateien in das INC-Verzeichnis von TIGERBAS und ersetzen Sie alle alten .INC-Dateien durch die mitgelieferten neuen gleichen Namens (Tip: die alten sollten Sie sich irgendwo sicher aufheben!).
- Fügen Sie die Datei "SER1C_K1.TDD" sowie alle .TAC-Dateien in das BIN-Verzeichnis von TIGERBAS ein (die alten .TAC-Dateien wie die alten .INC-Dateien irgendwo sichern und durch die neuen ersetzen).
- Alle .BMP-Dateien sind nur für das Grafik-Display notwendig und sind ggf in das Verzeichnis zu kopieren, in dem Sie auch das Programm "MAUS02.TIG" abspeichern. Ebenso können Sie sie in das Verzeichnis FLASH kopieren, denken Sie aber daran, im Tigerbasic unter der Funktion Optionen -> Verzeichnisse den Pfad für Flash-Dateien entsprechend einstellen.

Mit diesem Artikel werden zwei Programme angeboten:

Das Programm "MAUS01.TIG" benutzt das Plug-and-Play.Lab mit seinem alphanumerischen Standard-Display und die Schnittstelle "ser0". Diese ist mit einem Adapter wie oben beschrieben beschaltet, so daß direkt eine serielle Maus angeschlossen werden kann. Auf dem Display sind dann auf der ersten Zeile der Status der beiden Tasten, auf der zweiten Zeile die gerade zuletzt übertragenen X- und Y-Koordinaten sowie die kumulierte numerischen Werte der X- und Y-Position zu sehen.

"MAUS02.TIG" ist die Abwandlung eines Demoprogrammes (MOUSE_01.TIG) der Firma Wilke, das die Möglichkeiten der Maussteuerung zusammen mit dem Grafikdisplay zeigt. Dies setzt ein Grafik-Toolkit oder ein entsprechendes System voraus. Auch hier wird ein Mausadapter für die serielle Schnittstelle 0 wie oben beschrieben benötigt. Im Bildspeicher wird eine Deutschlandkarte abgelegt, die viel größer ist als eine Bildschirmgröße (240x128 Pixel). Der Cursor bewegt sich auf dem sichtbaren Teil des Bildschirms, fast wie unter Windows. Bei gedrückter linker Maustaste können Sie die komplette Karte "unter" dem Bildschirm bewegen. Die gedrückte rechte Maustaste negiert das Bild.

Viel Spaß beim Experimentieren!