

Digi-Controller



Lebenslauf

Das Programmieren mit der Sprache Basic habe ich in zwei Basic-Kursen gelernt, die ich während meiner Berufsausbildung besuchte. Zu diesem Zeitpunkt hatte ich meinen ersten intensiven Kontakt zu Computern. Es blieb nicht bei diesen beiden Kursen: Bis vorletztes Jahr nahm ich in meiner Freizeit – abends oder samstags – an einem Lehrgang über freiprogrammierbare Steuerungen teil.

Ich bin verheiratet und habe einen Sohn von drei Jahren.

Mein Lebenslauf in Kürze:

1957 geboren in Oberreichenbach bei Weißenhorn,

1972 Hauptschulabschluß, 1972-1975 Berufsausbildung zum Starkstromelektriker bei einer Mühlenbau-Firma mit anschließender Gesellenprüfung.

Erwerb des Elektronikpasses 1-2 und 4a-4d, Grundlehrgänge in Basic, 1978-1984 Lehrgang für frei programmierbare Steuerungen.

Seit 1979 Technischer Angestellter in einem Betrieb für schmiegsame Wärmegeräte.

Hobbies: Schützen- und Veteranenverein, Programmieren.

(Ernst Merk)

Fehlersuche in Gatterschaltungen und Simulation von Digitalschaltungen. Mittels eines Testlaufes, der die Wahrheitstabelle durchprüft, kann dieses Programm funktionale Schwächen frühzeitig entdecken. Außerdem läßt sich dieses Programm zu Lernzwecken bei angehenden Digitaltechnikern einsetzen.

Durch fortschreitende Automatisierungstechnik werden immer größere Anforderungen an elektronische Steuerungen gestellt. Ein immer häufiger auftretendes Schlagwort ist in der Elektronikbranche zu vernehmen: »Digitale Steuerungstechnik«. Dieser Beitrag wird Sie schrittweise in die modernen Automatisierungstechniken der Gegenwart einführen.

Zur Simulation und praxisgerechten Anwendung wurde das Programm »Digi-Controller« geschrieben, das der Grundversion einer speicherprogrammierbaren Steuerung (abgekürzt: SPS) entspricht. Dieser Begriff sagt schon sehr viel über die Funktionsweise moderner Steuerungsanlagen aus. In der Regel sind diese Steuerungen computerunterstützt.

Zunächst haben Sie sich sicher schon gefragt, wie Sie diese Technik im Hobbybereich einsetzen können. Zuerst sind einmal alle Modell-eisenbahner angesprochen, die sich einen automatischen Ablauf ihrer Bahnsteuerung zwar vorstellen können, jedoch die Hardware-Kosten bei einem fehlerhaften Aufbau scheuen. Als nächstes können alle Einsteiger in die Digitaltechnik aufhorchen, die die mühsamen Hardware-Aufbauten von Digital-labors verschmähen. Ja sogar für den Profi-Anwender sei auf das Austesten von Anweisungslisten nach DIN 19239 SPS verwiesen, da das Programm alle Eingaben (Inputs), Merker (Marker) und Ausgaben (Outputs) zu Papier bringt.

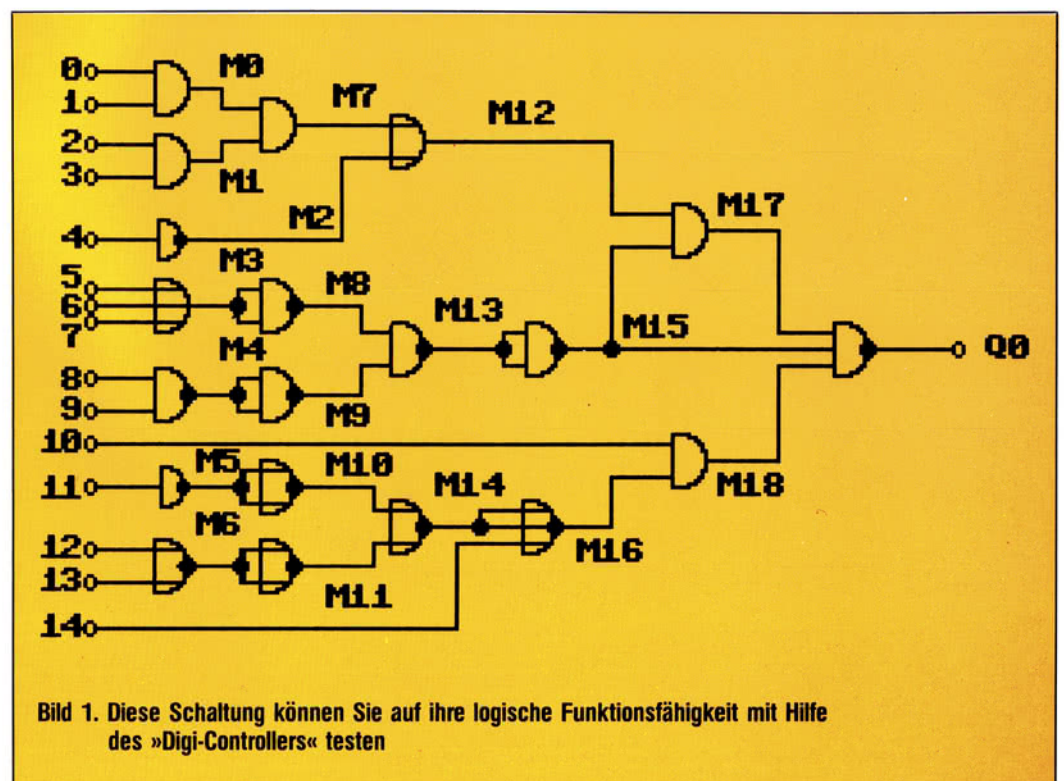
Alle 64 Inputs sind voreinstellbar oder Input 0 bis 7

werden im Dualzahlensystem automatisch hochgezählt, das Programm nach der Anweisungsliste (AWL) abgearbeitet und die digitalen Zustände ausgedruckt. Es kann auch ein TRACE-Lauf der eingegebenen Schaltung simuliert werden.

Maximal angesprochen werden können 64 Inputs, 64 Outputs und 64 Marker bei 512 Programmschritten.

Bild 1 zeigt Ihnen eine komplexe Schaltung, deren Funktion Sie anhand des Digi-Controllers durchspielen sollen. Die Schaltung soll als Ausgang eine 0 liefern. Versuchen Sie, die mögliche Kombination der Eingänge herauszufinden und auszutesten (die richtige Lösung finden Sie im Listing-Teil). Probieren Sie es doch einmal aus!

(Ernst Merk/dm)



Digi-Controller

Simulieren und Austesten digitaler Schaltungen und speicherprogrammierbarer Steuerungen ohne unnötigen und teuren Hardwareaufwand — Dieses Programm erledigt das für Sie.

Jeder Bastler, Digitaltechniker oder Informatikstudent kennt das Problem: Man hat nach mühevoller Überlegung eine digitale Schaltung, etwa für eine Ampel-Tippastensteuerung, entwickelt und muß nun diese Schaltung auf ihre Funktion testen. Hierzu überprüft man den Aufbau mit einer Wahrheitstabelle. Dies bedeutet, daß an die Eingänge verschiedene Bitkombinationen angelegt und die Zustände aller Gatter bis zum Ausgang auf ihre Pegel getestet werden. Haben Sie nun eine Schaltung mit vielen Eingängen und Gattern, so können Sie sich bestimmt vorstellen, wieviele Bitwerte da auftauchen. In dem Beispiel auf Seite 50 existieren 15 Eingänge, 20 Gatter und ein Ausgang. Das bedeutet, daß an den Eingängen 2^{15} (32768) verschiedene Bitkombinationen anliegen können. In einer Wahrheitstabelle würde dies 15 Spalten bedeuten. Per Hand dürften Sie sicher einige Zeit dafür benötigen, doch wozu gibt es schließlich Computer?

Das Programm »Digi-Controller« (Listing 1) eignet sich zur Simulation und Fehlersuche in Digitalschaltungen. Die Wahrheitstabelle kann sowohl von komplexen Digitalschaltungsentwürfen als auch von Steueranweisungen speicherprogrammierbarer Steuerungen (SPS) erstellt werden. Der Anwendungsbereich reicht vom Hobby-Anwender, der eine Eisenbahnsteuerung entwirft, bis zum Profi, der Programmteile von SPS austestet (der Autor benutzt das Programm zur Kontrolle selbst entwickelter Prüfgeräte für betriebliche Zwecke).

Nach DIN-Norm 19239 sind mit Ausnahme der Timerfunktion alle Anweisungen zur Erstellung einer Anweisungsliste (AWL) enthalten. Jedoch besteht die Möglichkeit, durch Programmierung anstelle eines Timers einen Input-Befehl zu wählen. Der jeweilige Zeitablauf kann somit simuliert werden (es ist bestimmt nicht sinnvoll, während einer Simulation auf einen Timerablauf von zum Beispiel 30 Minuten zu warten). Das Programm ist mit einem Standard-Compiler übersetzbar, was die Verarbeitungsgeschwindigkeit des Programms merklich erhöht (auf der Programmservice-Diskette ist selbstverständlich auch eine compilierte Version enthalten).

Doch vor der eigentlichen Bedienungsanleitung noch einige Begriffserklärungen:

Speicherprogrammierbare Steuerung — Elektronik-Steuer-
teil (Hardware)

Anweisungsliste (AWL) — Programm einer SPS (Software)

Eingänge einer SPS (Inputs) — Signalgeber (zum Beispiel Schalter)

Merker einer SPS (Marker) — Zwischenspeicher digitaler Ergebnisse

Ausgänge einer SPS (Outputs) — Signalempfänger (zum Beispiel Lampen)

Binärstelle — ein Bit (Inhalt: 0 oder 1)

Für den Einsteiger in die Digitaltechnik soll hier noch die Definition binärer Signale aufgezeigt werden:

Spannung vorhanden — keine Spannung vorhanden

Stromkreis geschlossen — Stromkreis geöffnet

Schalter eingeschaltet — Schalter ausgeschaltet

High-(oder 1-)Pegel — Low-(oder 0-)Pegel

Ordnet man diesen Zuständen Signale zu, so spricht man von binären (zweiwertigen) Signalen.

Die binäre Größe High-/Low-Potential (abgekürzt H-/L-Pegel) bedeutet, daß das elektrische Potential hoch (H, zum Bei-

spiel +5V) oder niedrig (L, zum Beispiel 0V) sein kann. Zu deren Darstellung können die binären Ziffern 0 und 1 herangezogen und als Signale bezeichnet werden. Eine binäre Stelle wird auch als »Bit« bezeichnet. Aus den binären Signalen läßt sich ein Dezimal-Dual-Zahlensystem aufbauen.

Wer sich intensiver mit dem Dualzahlensystem und dessen Anwendung beschäftigen möchte, der sei auf die überaus zahlreiche Literatur zu diesem Thema verwiesen.

Bevor es mit den Befehlen zur Erstellung einer Anweisungsliste losgeht, soll noch auf den Hardware-Aufwand für den Digi-Controller hingewiesen werden. Das Programm ist lauffähig auf einem C 64 mit Floppy 1541 und einem Epson RX/FX-80 mit Data-Becker-Interface. Andere Drucker sind über Änderung der Sekundäradresse im OPEN- und PRINT #-Befehl anpassungsfähig.

Programmbeschreibung

Für die Anweisungslisten (AWL), die in den modernen SPS verschlüsselt abgelegt sind und mit denen auch der Digi-Controller arbeitet, gilt die DIN-Norm 19239. Hier sind den Operationen und Operanden anwenderspezifische Kurzzeichen in englisch und deutsch zugeordnet. Es werden folgende Abkürzungen verwendet:

Für die Operationen:

O	OR	ODER-Verknüpfung
A	AND	UND-Verknüpfung
X	XOR	EXKLUSIV-ODER-Verknüpfung
=	ASSIGNMENT	ZUWEISUNG
J	JUMP	absolute Sprunganweisung
J!	JUMP IF	indirekte Sprunganweisung
;		Kommentarzeile

und für die Operanden:

I	INPUT	Eingang
M	MARKER	Merker
Q	OUTPUT	Ausgang
N	NOT	Negation

Aufbau einer Anweisungsliste (AWL)

Die Anweisungsliste besteht im allgemeinen aus Anweisungs- und Kommentarzeilen.

Zeilennummer / Operation / Operand / Parameter / Kommentar

3stellig / 1stellig / 2stellig / 3stellig / 21stellig

Die Zeilennummer kann mittels Eingabe von Ziffern oder durch die Cursor-Tasten eingestellt werden. Der Zeilennummernbereich umfaßt 512 Zeilen. Die Zeilen werden vom Compiler in aufsteigender Reihenfolge abgearbeitet. Ausnahme: Absolute oder indirekte Sprungbefehle.

Die Operation »;« kennzeichnet eine Kommentarzeile. Die Operatoren »O«, »A« und »X« stellen eine logische Verknüpfung zwischen Inputs, Markern und Outputs dar. Das »=« Zeichen weist das Ergebnis einer logischen Operation einem Marker oder Output zu. »J« kennzeichnet einen Sprungbefehl.

Der Operand definiert Input, Marker oder Output. Durch hinzufügen eines »N«-Zeichens kann der Operand negiert (invertiert) werden. Wurde als Operation ein »J«-Sprungbefehl definiert, so kann durch Angabe eines »I«-Zeichens im Operanden ein indirekter Sprungbefehl festgelegt werden.

Der Parameter legt die Nummer eines Inputs, Markers oder Outputs fest. Bei Sprungbefehlen wird hier die Zieladresse der nächsten Zeilennummer festgelegt.

Zulässige Eingabewerte:

0 bis 64 — Input, Marker, Output

0 bis 512 — absolute Sprungadresse (Jump)
0 bis 512 — indirekte Sprungadresse (Jump if)

Der Kommentar dient der Programmdokumentation.

Bevor wir nun in die Programmierung eigener AWL einsteigen, hier noch eine detaillierte Beschreibung des Controllers. Vom Hauptmenü aus sind aufzurufen:

— Anweisungsprogramme löschen: Das zuletzt gelistete Programm (AWL) kann mittels der <F1>-Taste gelöscht werden. Anzeige des nächsten Programms mit <F5>, Rückkehr ins Hauptmenü mittels <F7>.

— Anweisungsliste von Disk lesen: Die Datendiskette mit den gespeicherten AWL kann auch die Programmdiskette des Controllers sein. Diskettenfehler werden ausgegeben. In der Directory-Übersicht kann durch Drücken von:

<F1> das zuletzt angezeigte AWL geladen;
<F5> die nächste AWL angezeigt und
<F7> in das Hauptmenü gesprungen werden.

— Anweisungsliste erstellen: Das ist nun die Eingabeebene, von der aus die Anweisungs- und Kommentarzeilen erstellt und geändert werden. Die Eingaben sind jeweils mit <RETURN> zu quittieren. Unterhalb der Kopfzeile werden die Auswahlmöglichkeiten angezeigt. Um eine größere Programmübersicht zu erhalten, drücken Sie bitte <F1>. Mit <F7> gelangen Sie in ein Untermenü, wobei Sie nun zwischen dem Rücksprung ins Hauptmenü oder Speichern der AWL wählen können. Es ist sinnvoll, die AWL gelegentlich zu speichern.

— Anweisungsliste ausdrucken: Der Ausdruck kann auf Bildschirm oder Drucker gelenkt werden. Es ist auch möglich, die AWL nur teilweise auszugeben.

— Auswertung von AWL ausdrucken (Wahrheitstabelle): Mittels der TRACE-Funktion werden die abgearbeiteten Zeilennummern der AWL angezeigt (Compilerlauf). Hierbei erkennt das Programm Endlosschleifen der AWL (bei »J«-Befehlen).

Bei der Einzelauswertung können anfangs alle Bit-Kombinationen der einzelnen Inputs mit »0«, »1« oder einem Leerzeichen belegt werden. Spricht das Programm einen Input an, der mit einem Leerzeichen belegt ist, so wird der Inhalt des Inputs wie eine logische Null (0) behandelt. Im Ausdruck wird der Inhalt des Inputs ebenfalls mit einem Leerzeichen ausgegeben.

Wählen Sie die Gesamtauswertung, so wird zunächst der höchste Inputparameter (maximal Input 7) der AWL ermittelt. Das bedeutet: wurde zum Beispiel Input 0 bis 3 in der AWL angesprochen, so generiert das Programm $2^4 = 16$ Programmdurchläufe. Dabei setzt der Digi-Controller alle möglichen Bit-Zustände von Input 0 bis 3 automatisch. Diese arbeitet der Compiler ab und druckt sie aus (bei diesem Beispiel reichen die Bit-Zustände von 0000 bis 1111). Falls Zuweisungen an Marker oder Outputs übergeben wurden, so druckt das Programm auch diese Werte aus.

Umsetzen digitaler Schaltungen in eine AWL

Die Anzahl der Inputs aller logischen Schaltungen muß nicht drei sein, sondern kann je nach Bedarf bis 64 betragen (in unserem Beispiel auf Seite 50 existieren 15 Eingänge). Wichtig bei der Erstellung einer AWL ist:

Immer bei einer neuen AWL oder nach jeder Zuweisung (»=«) mit einer ODER-Anweisung als Operation beginnen. Zuerst muß ein internes Hilfsregister mit dem Inhalt eines Inputs (Bit = 1 oder 0) geladen werden. Dies ist nur mit der ODER-Anweisung möglich. Da das Hilfsregister nach jeder Zuweisung mit »=« gelöscht wird, ist es ebenfalls notwendig, das Hilfsregister mittels einer ODER-Anweisung zu laden.

Das Ergebnis einer logischen Verknüpfung ergibt sich aus dem Vergleich eines Inhalts von einem Input, Marker oder Output mit dem Inhalt des Hilfsregisters und wird im Hilfsregister abgelegt.

Die ODER-Schaltung (Bild 1)

Merksatz: Der Output Q0 führt H-Pegel, wenn mindestens an einem Input (I0 bis I2) ein H-Pegel anliegt.

AWL:

0 ; *OR-Verknüpfung mit 3 Inputs

1 0 I0 *Hilfsregister mit Inhalt von I0 laden
2 0 I1 *Inhalt Input I1 mit Inhalt Hilfsregister verknüpfen
3 0 I2 *Inhalt Input I2 mit Inhalt Hilfsregister verknüpfen
4 ; *Ergebnis der Verknüpfung steht im Hilfsregister
5 = Q0 *Inhalt Hilfsregister Output Q0 zuweisen

Bitte geben Sie diese Anweisungszeilen im Digi-Controller ein. Sie können nun die AWL ausdrucken und auf Richtigkeit überprüfen. Das Ergebnis Ihrer Arbeit können Sie als Auswertung (Gesamtausdruck) ausgeben lassen.

Die UND-Schaltung (Bild 2)

Merksatz: Der Output Q0 führt nur dann einen H-Pegel, wenn alle Inputs (I0 bis I2) einen H-Pegel führen.

AWL:

0 ; *AND-Verknüpfung mit 3 Inputs
1 0 I0 *Hilfsregister mit Inhalt von Input I0 laden
2 A I1 *Inhalt Input I1 mit Inhalt Hilfsregister verknüpfen
3 A I2 *Inhalt Input I2 mit Inhalt Hilfsregister verknüpfen
4 ; *Ergebnis der Verknüpfung steht im Hilfsregister
5 = Q0 *Inhalt Hilfsregister Output Q0 zuweisen

Die EXCLUSIV-ODER-Schaltung (Bild 3)

Merksatz: Der Output Q0 führt nur dann einen H-Pegel, wenn an genau einem Input (I0/I1) ein H-Pegel anliegt. Führt mehr als ein Input einen H-Pegel oder liegen alle Inputs auf L-Pegel, so steht am Ausgang Q0 ein L-Pegel.

AWL:

0 ; *XOR-Schaltung mit zwei Inputs
1 0 I0 *Hilfsregister mit Inhalt von Input I0 laden
2 X I1 *Inhalt Hilfsregister mit Inhalt Input I1 verknüpfen
3 ; *Ergebnis steht im Hilfsregister
4 = Q0 *Inhalt Hilfsregister Output Q0 zuweisen

Die NICHT-Schaltung (Bild 4)

Merksatz: Der Output Q0 führt immer den umgekehrten Pegel des Inputs I0.

AWL:

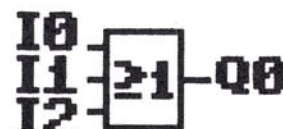


Bild 1. ODER-Schaltung

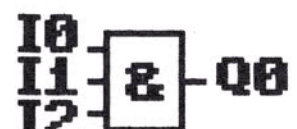


Bild 2. UND-Schaltung

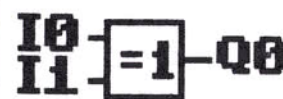
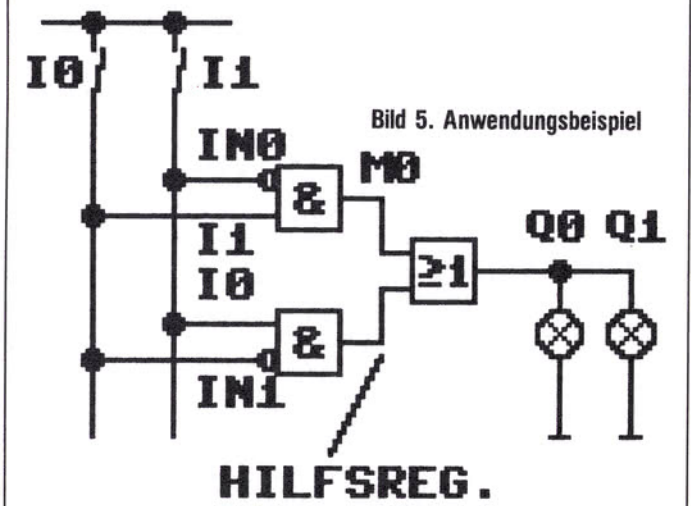


Bild 3. EXOR-Schaltung



Bild 4. NICHT-Schaltung



```

0 ;      *NICHT-Verknüpfung am Output
1 0 IN1  *Inhalt Input I1 invertiert in Hilfsregister la-
den
2 = Q1   *Inhalt Hilfsregister Output Q1 zuweisen

```

Sprung-Befehle

Zum Einfügen und Austesten von Programmteilen können Sprungbefehle in der AWL programmiert werden.

— Der JUMP-Befehl: Gibt im Parameter die nächste Zeilennummer an, die vom Compiler abgearbeitet werden soll. Damit ist es zum Beispiel möglich, aus dem Hauptprogramm an

das Programmende zu springen, eingefügte Programmzeilen abzuarbeiten und wieder ins Hauptprogramm zurückzuspringen.

— Der JUMP IF-Befehl: Gibt im Parameter ebenfalls die nächste Zeilennummer an, die vom Compiler abgearbeitet werden soll. Jedoch wird der indirekte Sprungbefehl nur dann ausgeführt, wenn das Ergebnis der letzten Verknüpfung einen L-Pegel ergeben hat (Inhalt von Hilfsregister = L-Pegel).

Achtung: Als Zieladresse nie auf eine UND-Anweisung mit indirektem Sprungbefehl springen.

AWL:

```

0 ;      *Sprungbefehle
1 0 IO   *Hilfsregister mit Inhalt von Input IO laden
2 J 5    *Springe nach Zeilennummer 5
3 0 I1   *ODER-Verknüpfung bei indirektem Sprung
4 ;      *Kommentarzeile
5 0 I2   *ODER-Verknüpfung mit Inhalt von Input I2
6 J I3   *Springe nach Zeilennummer 3, wenn Hilfsregi-
ster = L-Pegel
7 = Q0   *Ergebnis Output Q0 zuweisen

```

Die Sprünge werden solange wiederholt, bis ein Input (I1/I2) einen H-Pegel annimmt.

Praktisches Anwendungsbeispiel:

Steuerung für eine Treppenhausbeleuchtung:

In einem Treppenhaus soll die Beleuchtung von zwei Stellen unabhängig voneinander ein- und ausgeschaltet werden können (Wechselschaltung). Die Schalter I0 und I1 wirken über die logische Verknüpfungsschaltung auf zwei Lampen (Q0/Q1), die parallel geschaltet sind.

Wahrheitstabelle:

I0	I1	Q0	Q1
0	0	0	0
0	1	1	1
1	0	1	1
1	1	0	0

AWL: (Bild 5)

```

0 ;      *Treppenhausbeleuchtung
1 0 IN 0  *Hilfsregister laden
2 A I 1   *UND-Verknüpfung mit Hilfsregister
3 = M 0   *Zwischenergebnis im Marker ablegen
4 0 IN 1  *Hilfsregister neu laden
5 A I 0   *UND-Verknüpfung mit Hilfsregister
6 ;      *Ergebnis steht im Hilfsregister
7 0 M 0   *ODER-Verknüpfung Hilfsregister und Marker 0
8 = Q 0   *Ergebnis nach Output Q0 schreiben

```

Bitte AWL eingeben und das Ergebnis mittels Gesamtauswertung überprüfen.

Zum Testen Ihrer erworbenen Fähigkeiten noch eine kleine Aufgabe, bevor die AWL und die Lösung der auf Seite 50 vorgestellten Schaltung, die Sie mit dem Controller testen sollten, präsentiert wird.

Aufgabe: Weichenstellung für Modelleisenbahn

Mittels dreier Schalter (I0 bis I2) sollen sieben Weichen einer Modelleisenbahn umgestellt werden. Die Weichenstellung wird durch eine Umwandlung von Dual- in Dezimalcode erreicht.

Schalterstellung			Weiche Nr. (1 = Änderung)						
Dualcode	1	2	3	4	5	6	7		
0 0 0	0	0	0	0	0	0	0		
0 0 1	1	0	0	0	0	0	0		
0 1 0	0	1	0	0	0	0	0		
0 1 1	0	0	1	0	0	0	0		
1 0 0	0	0	0	1	0	0	0		
1 0 1	0	0	0	0	1	0	0		
1 1 0	0	0	0	0	0	1	0		
1 1 1	0	0	0	0	0	0	1		

Und hier die Auflösung der auf Seite 50 abgebildeten Schaltung. Tabelle 1 zeigt die AWL, wie sie für einen Q0 = L-Pegel aussehen sollte. Tabelle 2 veranschaulicht den Testlauf, um zu sehen, ob die richtigen Pegel anliegen.

(Ernst Merk/dm)

```

Datei- Name : testschaltung
Datum      : Tg.Mo.Ja

> 0 0 I 0 Eingang 0
> 1 A I 1 AND Eingang 1
> 2 = M 0 = Marke 0
> 3 0 I 2 Eingang 2
> 4 A I 3 AND Eingang 3
> 5 = M 1 = Marke 1
> 6 0 IN 4 NOT Eingang 4
> 7 = M 2 = Marke 2
> 8 0 I 5 Eingang 5
> 9 0 I 6 OR Eingang 6
> 10 0 I 7 OR Eingang 7
> 11 = M 3 = Marke 3
> 12 0 I 8 Eingang 8
> 13 A I 9 AND Eingang 9
> 14 = M 4 = Marke 4
> 15 0 MN 4 invertieren
> 16 = M 4 von Marke 4
> 17 0 IN 11 NOT Eingang 11
> 18 = M 5 = Marke 5
> 19 0 I 12 Eingang 12
> 20 0 I 13 OR Eingang 13
> 21 = M 6 = Marke 6
> 22 0 MN 6 invertieren
> 23 = M 6 von Marke 6
> 24 0 M 0 Marke 0
> 25 A M 1 AND Marke 1
> 26 = M 7 = Marke 7
> 27 0 MN 3 NOT Marke 3
> 28 = M 8 = Marke 8
> 29 0 MN 4 NOT Marke 4
> 30 = M 9 = Marke 9
> 31 0 MN 5 NOT Marke 5
> 32 = M 10 = Marke 10
> 33 0 MN 6 NOT Marke 6
> 34 = M 11 = Marke 11
> 35 0 M 7 Marke 7
> 36 0 M 2 OR Marke 2
> 37 = M 12 = Marke 12
> 38 0 M 8 Marke 8
> 39 A M 9 AND Marke 9
> 40 = M 13 = Marke 13
> 41 0 MN 13 invertieren
> 42 = M 13 von Marke 13
> 43 0 M 10 Marke 10
> 44 0 M 11 OR Marke 11
> 45 = M 14 = Marke 14
> 46 0 MN 14 invertieren
> 47 = M 14 von Marke 14
> 48 0 MN 13 NOT Marke 13
> 49 = M 15 = Marke 15
> 50 0 M 14 Marke 14
> 51 0 I 14 OR Eingang 14
> 52 = M 16 = Marke 16
> 53 0 MN 16 invertieren
> 54 = M 16 von Marke 16
> 55 0 M 12 Marke 12
> 56 A M 15 AND Marke 15
> 57 = M 17 = Marke 17
> 58 0 I 17 Eingang 17
> 59 A M 16 AND Marke 16
> 60 = M 18 = Marke 18
> 61 0 M 17 Marke 17
> 62 A M 15 AND Marke 15
> 63 A M 18 AND Marke 18
> 64 = Q 0 = Ausgang 0
> 65 0 QN 0 invertieren
> 66 = Q 0 des Ausganges

```

Tabelle 1. So müßte die AWL für die Schaltung auf Seite 52 aussehen, wenn am Ausgang ein L-Pegel anliegen soll

```

Datei- Name      : testschaltung
Einzel - Auswertung : Tg.Mo.Ja

Opr/Bit : 6686855555555555444444443333333322222222111111111000000000000
          4321098765432109876543210987654321098765432109876543210
Input I  : 000111100001111
Marker M : 1111001011110001111
Output Q : 0

```

Tabelle 2. Das Testprotokoll von Tabelle 1 in der Einzelauswertung

```

15 PRINT CHR$(147);:DIM I$(65),M$(65),M$(65) <158>
5) ,0$(513) <010>
80 O1$="" :O2$="(2SPACE)":O3$="(3SPACE)":O <053>
4$="(1SPACE)" <173>
90 ZL$="" :PO=3 <038>
100 PRINT CHR$(147)"(GREY 1)":POKE 53280,5 <050>
:Poke 53281,5 <137>
120 KO$="(RVSON)DIGIT-CONTROLLER V. E. MER <163>
K (07309/6501)(RVOFF)" <192>
130 D$="DIGIT1" <161>
140 BR$="(RVSON,40SPACE,RVOFF)" <185>
150 BL$="(40SPACE)" <125>
160 KZ$=".Z.N..OPR.OP..NR...KOMMENTAR..... <125>
....." <125>
180 LI$="(RVSON)OKSR(RVOFF)=ZN+/(2SPACE,R <125>
VSON,SPACE)E(RVOFF)INGABE(2SPACE,RVSON <125>
)E1(RVOFF)=LIST(2SPACE,RVSON)E7(RVOFF) <125>
=MENUE(RVSON,SPACE,RVOFF)" <125>
200 O$="(RVSON)OPERATION(RVOFF,2SPACE)O&A <125>
N&XOR&JMP&JI = ;(2SPACE,RVSON)RETURN(R <125>
VROFF)" <125>
210 OP$="(RVSON)OPERAND(RVOFF,SPACE)I(N) & <125>
(N) M(N)(2SPACE)(N=NICHT)(SPACE,RVSON) <125>
RETURN(RVOFF)" <125>
220 OE$="(RVSON)OPERANDEN- NR.(RVOFF,SPACE <125>
)J(512)(6SPACE)*(64)(2SPACE,RVSON)BETU <125>
RN(RVOFF)" <125>
230 KM$="(RVSON)KOMMENTAR(RVOFF,SPACE)0-21 <125>
ZEICHEN(12SPACE,RVSON)RETURN(RVOFF)" <125>
240 DA$="(RVSON,SPACE,RVOFF,3RIGHT,RVSON,2 <125>
SPACE,RVOFF,RIGHT,RVSON,2SPACE,RVOFF,2 <125>
RIGHT,RVSON,2SPACE)" <125>
242 DA$=DA$+"(RVOFF,3RIGHT,RVSON,2SPACE,RV <125>
OFF,21RIGHT,RVSON,SPACE,RVOFF)" <125>
250 LA$="(RVSON,6SPACE)ANWEISUNGSLISTE AUS <125>
DRUCKEN!(6SPACE,RVOFF)" <125>
260 Z1$=".OPR/BIT : 6666655555555554444444 <125>
4433333" <125>
270 Z2$="33333222222222211111111110000000 <125>
00" <125>
280 Z3$="(10SPACE)432109876543210987654321 <125>
098765" <125>
290 Z4$="432109876543210987654321098765432 <125>
10" <125>
1000 REM ***** <125>
1010 PRINT CHR$(147);KO$:PRINT:PRINT:PRINT <125>
1020 PRINT" A E N U E " <125>
1025 PRINT"UUUUUUUUUU":PRINT:PRINT <125>
1040 PRINT" A = ANWEISUNGS- PROGRAMME LOES <125>
CHEN":PRINT <125>
1050 PRINT" B = ANWEISUNGSLISTE (AHL) V. B <125>
ISK LESEN" <125>
1060 PRINT" C = ANWEISUNGSLISTE (AHL) ERST <125>
ELLEN":PRINT <125>
1070 PRINT" D = ANWEISUNGSLISTE (AHL) AUSD <125>
RUCKEN":PRINT <125>
1080 PRINT" E = AUSWERTUNG V. AHL AUSDRUCK <125>
EN":PRINT <125>
1090 PRINT" F = ENDE" <125>
1100 GET E$:IF E$=""THEN 1100 <125>
1105 IF E$="A"THEN 21000 <125>
1110 IF E$="B"THEN F2=0:GOTO 1200 <125>
1120 IF E$="C"THEN 1500 <125>
1130 IF E$="D"THEN 4000 <125>
1140 IF E$="E"THEN 5000 <125>
1150 IF E$="F"THEN PRINT CHR$(142)CHR$(147 <125>
)"(LG.BLUE)";:POKE 53280,14:POKE 532 <125>
81,6:END <125>
1160 GOTO 1100 <125>
1200 REM ***** <125>
1210 PRINT CHR$(147);KO$:BL$ <125>
1225 PRINT:PRINT:PRINT <125>
1230 PRINT" BITTE DATENDISKETTE EINLEGEN U <125>
ND(8SPACE,RVSON)RETURN(RVOFF,SPACE)DR <125>
UECKEN !":PRINT:PRINT <125>
1235 PRINT"(SPACE,RVSON)E7(4SPACE,RVOFF,SP <125>
ACE)=ZURUECK ZUM MENUE":PRINT <125>
1240 GET EE$:IF EE$=""THEN 1240 <125>
1250 IF EE$=CHR$(13)THEN GOSUB 53000:GOTO <125>

```

```

1945 IF Z>4 THEN 1600 <031>
1950 PRINT O$(J);Z=Z+1 <236>
1960 NEXT <192>
1970 GOTO 1600 <228>
2000 REM *** ZEILE+1 ***** <183>
2010 ZN=VAL(ZL$):ZN=ZN+1:IF ZN>500 THEN ZN=500 <169>
2020 ZL$=LEFT$(BL$,4)+STR$(ZN):ZL$=RIGHT$(ZL$,3) <023>
2030 GOTO 1700 <048>
2050 REM *** ZEILE-1 ***** <251>
2060 ZN=VAL(ZL$):ZN=ZN-1:IF ZN<0 THEN ZN=0 <149>
2070 ZL$=LEFT$(BL$,4)+STR$(ZN):ZL$=RIGHT$(ZL$,3) <075>
2080 GOTO 1700 <161>
2500 REM *** EINGABE OPERATION ***** <011>
2510 PRINT "HOME,3DOWN";O$;GOTO 2580 <070>
2520 E$="":GET E$:IF E$="" THEN 2520 <188>
2530 IF E$=CHR$(13)AND O1$="" THEN PRINT "HOME,3DOWN";LI$;O$(VAL(ZL$))="":GOTO 2000 <177>
2531 IF E$=CHR$(13)AND O1$="" THEN PRINT "HOME,3DOWN";LI$;O2$="{2SPACE}";O3$="{3SPACE}":GOTO 3100 <066>
2535 IF E$=CHR$(13)AND O1$<>"" THEN 2600 <105>
2540 IF E$=CHR$(79)OR E$=CHR$(65)OR E$=CHR$(74)OR E$=CHR$(61)OR E$=CHR$(20) THEN 2560 <077>
2545 IF E$=CHR$(88)OR E$=CHR$(59) THEN 2560 <071>
2550 GOTO 2520 <100>
2560 IF E$=CHR$(20) THEN O1$="":GOTO 2580 <129>
2565 IF E$=CHR$(61) THEN O1$=E$:GOTO 2580 <179>
2568 IF E$=CHR$(59) THEN O1$=E$:GOTO 2580 <197>
2570 FA=ASC(E$)+128:E$=CHR$(FA):O1$=E$: <011>
2580 PRINT "HOME,8DOWN,6RIGHT,SPACE,LEFT";O1$; <139>
2590 GOTO 2520 <142>
2600 REM *** OPERAND ***** <075>
2610 PRINT "HOME,3DOWN";OP$;GOTO 2680 <045>
2620 E$="":GET E$:IF E$="" THEN 2620 <050>
2625 IF E$=CHR$(73)AND O1$=CHR$(61) THEN 2620 <187>
2627 IF E$=CHR$(13)AND O1$=CHR$(74+128) THEN 2740 <206>
2630 IF E$=CHR$(13)AND O2$="" THEN 2500 <157>
2635 IF E$=CHR$(13)AND O2$<>"" THEN 2800 <048>
2640 IF O1$<>CHR$(74+128) THEN IF E$=CHR$(73)OR E$=CHR$(81)OR E$=CHR$(77) THEN 2700 <176>
2642 IF E$=CHR$(20) THEN 2660 <092>
2645 IF E$=CHR$(78)AND LEN(O2$)=1 THEN 2720 <242>
2647 IF O1$=CHR$(74+128)AND E$=CHR$(73) THEN 2700 <054>
2650 GOTO 2620 <218>
2660 IF LEN(O2$)=2 THEN O2$=LEFT$(O2$,1):GOTO 2680 <216>
2670 IF LEN(O2$)=1 THEN O2$="" <104>
2680 PRINT "HOME,8DOWN,9RIGHT,2SPACE,2LEFT";O2$; <021>
2690 GOTO 2620 <002>
2700 FA=ASC(E$)+128:E$=CHR$(FA):O2$=E$: <143>
2710 GOTO 2680 <215>
2720 FA=ASC(E$)+128:E$=CHR$(FA):O2$=O2$+E$ <098>
2730 GOTO 2680 <235>
2740 PRINT "HOME,8DOWN,9RIGHT,2SPACE,2LEFT";O2$; <081>
2750 GOTO 2800 <030>
2800 REM *** OPERANDEN- NUMMER ***** <028>
2805 IF LEN(O2$)=1 THEN O2$=O2$+LEFT$(BL$,1) <241>
2807 IF O2$="" THEN O2$="{2SPACE}" <135>
2810 PRINT "HOME,3DOWN";OE$;P1=3:GOTO 2970 <044>
2850 E$="":GET E$:IF E$="" THEN 2850 <154>
2860 IF E$=CHR$(13)AND RIGHT$(O3$,1)=CHR$(32) THEN 2600 <115>
2870 IF E$=CHR$(13)AND RIGHT$(O3$,1)<>CHR$(32) THEN 3100 <250>
2880 IF E$>CHR$(47)AND E$<CHR$(58) THEN 2900 <166>
2885 IF E$=CHR$(20) THEN 2920 <177>
2890 GOTO 2850 <076>
2900 O3$=RIGHT$(O3$,2)+E$:O3$=RIGHT$(O3$,3):P1=P1+1: <150>
2910 GOTO 2954 <113>
2920 IF P1>3 THEN P1=3:GOTO 2954 <196>
2930 O3$=CHR$(32)+LEFT$(O3$,2):O3$=LEFT$(O3$,3):P1=P1-1 <037>
2940 IF P1<1 THEN P1=1: <146>
2954 IF VAL(O3$)>512 AND LEFT$(O1$,1)=CHR$(74+128) THEN O3$="{3SPACE}":P1=3:GOTO 2970 <153>
2960 IF VAL(O3$)>64 AND LEFT$(O1$,1)<>CHR$(74+128) THEN O3$="{3SPACE}":P1=3:GOTO 2970 <193>
2970 PRINT CHR$(19)"(8DOWN,13RIGHT,3SPACE,3LEFT)";O3$; <005>
2980 GOTO 2850 <166>
3100 REM *** KOMMENTAR ***** <140>
3110 PRINT "HOME,3DOWN";KM$ <226>
3120 P2=0:GOTO 3220 <125>
3150 E$="":GET E$:IF E$="" THEN 3150 <096>
3160 IF E$=CHR$(13) THEN PRINT "HOME,3DOWN";LI$;GOTO 3300 <101>
3165 IF E$=CHR$(58)OR E$=CHR$(44) THEN 3150 <062>
3170 IF E$=CHR$(20) THEN P2=P2-1:GOTO 3250 <115>
3175 IF E$>CHR$(31)AND E$<CHR$(91) THEN P2=P2+1:GOTO 3200 <241>
3180 IF E$>CHR$(31+128)AND E$<CHR$(91+128) THEN P2=P2+1:GOTO 3200 <166>
3190 GOTO 3150 <018>
3200 IF P2>21 THEN P2=21 <178>
3210 O4$=LEFT$(O4$,P2-1)+E$+LEFT$(BL$,21-P2) <040>
3220 PRINT "HOME,8DOWN,18RIGHT";O4$; <101>
3230 GOTO 3150 <058>
3250 IF P2<0 THEN P2=0 <241>
3260 O4$=LEFT$(O4$,P2)+LEFT$(BL$,21-P2): <143>
3270 GOTO 3220 <018>
3300 REM *** EINGABESTRING GESAMT ***** <206>
3302 IF RIGHT$(ZL$,1)<>CHR$(32) THEN 3320 <049>
3310 GOTO 1700 <058>
3320 ZN=VAL(ZL$) <061>
3330 O$(ZN)=CHR$(62)+ZL$+"{2SPACE}"+O1$+"{2SPACE}"+O2$+"{2SPACE}"+O3$+"{2SPACE}"+O4$+" " <068>
3340 GOTO 2000 <242>
4000 REM *** LISTE AUSDRUCKEN ***** <086>
4010 PRINT CHR$(147);KO$;BL$;LA$ <120>
4020 PRINT "BRUECKE{SPACE,RVSON}B{RVOFF}ILDSCHIRM ODER{SPACE,RVSON}B{RVOFF}RUCKER{4SPACE}":PRINT:PRINT <227>
4030 E$="":GET E$:IF E$="" THEN 4030 <142>
4040 IF E$=CHR$(66) THEN F3=0:GOTO 4100 <155>
4050 IF E$=CHR$(68) THEN F3=1:GOTO 4100 <041>
4055 IF E$=CHR$(136) THEN 1000 <150>
4060 GOTO 4030 <054>
4100 IF F3=1 THEN PRINT "RVSON,7SPACE}BITTE BRUCKER EINSCHALTEN !!!{4SPACE,RVOFF}"; <209>
4110 IF F3=1 THEN PRINT BL$; <088>
4130 INPUT "VON ZEILE :{5SPACE}0{3LEFT}";O1$ <173>
4140 INPUT "BIS ZEILE :{5SPACE}512{5LEFT}";O2$ <039>
4142 PRINT <180>
4150 P1=VAL(O1$):P2=VAL(O2$):IF P1>P2 THEN 4000 <199>
4160 IF P1<0 OR P1>512 OR P2<0 OR P2>512 THEN 4000 <227>
4164 INPUT "DATUM :{4SPACE}JG.MD.JA{10LEFT}";EE$ <218>
4170 PRINT:PRINT:PRINT "RVSON,6SPACE}E1= START{6SPACE}E7= MENUE{10SPACE,RVOFF}":PRINT:PRINT <244>
4200 E$="":GET E$:IF E$="" THEN 4200 <250>
4210 IF E$=CHR$(136) THEN 1000 <051>
4220 IF E$<>CHR$(133) THEN 4200 <038>
4225 PRINT CHR$(147); <052>
4230 IF F3=0 THEN CLOSE 1:OPEN 1,3,8 <006>
4240 IF F3=1 THEN CLOSE 1:OPEN 1,4,7 <144>
4250 Z=0:PRINT#1,KO$;BR$ <129>
4260 IF F3=1 THEN PRINT#1,"DATEI- NAME : ";F$;PRINT#1,"DATUM{7SPACE}":EE$;PRINT#1 <008>
4300 FOR J=P1 TO P2 <203>
4310 IF F3=0 AND Z=20 THEN GOSUB 51000 <160>
4320 GET E$ <242>
4330 IF E$=CHR$(136) THEN CLOSE 1:GOTO 1000 <221>
4340 IF O$(J)="" THEN 4360 <050>
4350 Z=Z+1:PRINT#1,O$(J);IF F3=1 THEN PRI

```

Listing 1. »Digi-Controller«

NT#1	<193>	6660 E\$="":GET E\$:IF E\$=""THEN 6660	<045>
4360 NEXT	<052>	6670 IF E\$=CHR\$(13)THEN 1000	<148>
4370 IF F3=0 THEN GOSUB 51000	<065>	6690 IF E\$=CHR\$(133)THEN 6590	<074>
4380 PRINT#1:PRINT#1:PRINT#1	<066>	6700 GOTO 6660	<108>
4390 CLOSE 1:GOTO 1000	<027>	7000 REM *** EINZELAUSWERTUNG *****	<230>
5000 REM *** AUSWERTUNG *****	<208>	7002 INPUT " DATUM : (4SPACE)TG. MO. JA(10LEFT	
5010 PRINT CHR\$(147)K0\$;BL\$:PRINT:PRINT	<010>	)";EE\$:PRINT:PRINT	<074>
5020 PRINT" BITTE WAECHELEN SIE :":PRINT"UUU		7005 PRINT" (RVSON)#####(2SPACE)BITTE DRUCK	
UUUUUUUUUUUUUUUUUUUUUU":PRINT	<144>	ER EINSCHALTEN(2SPACE)#####(RVOFF)";	<045>
5025 PRINT" (SPACE,RVSON)I(RVOFF)RACE(17SPA		7010 PRINT" (RVSON)#####(2SPACE)E1= START(7	
CE)OHNE DRUCKER(3SPACE)":PRINT	<091>	SPACE)E7= MENUE(2SPACE)#####(RVOFF)"	
5030 PRINT" (SPACE,RVSON)E(RVOFF)INZELAUWE		:PRINT	<203>
RTUNG = I(0-64) MANUELL":PRINT	<022>	7020 E\$="":GET E\$:IF E\$=""THEN 7020	<076>
5040 PRINT" (SPACE,RVSON)E(RVOFF)ESAMTAUSWE		7030 IF E\$=CHR\$(136)THEN 1000	<077>
RTUNG = I(0-7) (2SPACE)AUTOMATISCH":PR		7040 IF E\$<>CHR\$(133)THEN 7020	<001>
INT:PRINT:PRINT:PRINT	<241>	7090 MA=-1	<019>
5050 E\$="":GET E\$:IF E\$=""THEN 5050	<218>	7100 PRINT CHR\$(147)K0\$;BL\$	<035>
5055 IF E\$=CHR\$(84)THEN 6500	<136>	7110 PRINT BR\$:" (RVSON,SPACE)E5E(RVOFF)=E	
5060 IF E\$=CHR\$(71)THEN 6000	<230>	ING. (SPACE,RVSON)E1-E3(RVOFF)=MODUS(S	
5070 IF E\$=CHR\$(69)THEN 7000	<015>	PACE,RVSON)RETURN(RVOFF,SPACE,RVSON)E	
5080 GOTO 5050	<130>	7 (RVOFF)=MENUE(RVSON,SPACE,RVOFF)";BR	
6000 REM *** GESAMTAUSWERTUNG *****	<167>		<243>
6002 INPUT" DATUM : (4SPACE)TG. MO. JA(10LEFT		7140 K=0:FB\$="(9SPACE)":	<157>
)";EE\$:PRINT:PRINT	<090>	7145 GOTO 7700	<131>
6005 PRINT" (RVSON)#####(2SPACE)BITTE DRUCK		7150 B\$="":GET B\$:IF B\$=""THEN 7150	<204>
ER EINSCHALTEN(2SPACE)#####(RVOFF)";	<061>	7160 IF B\$=CHR\$(17)THEN 7200	<104>
6010 PRINT" (RVSON)#####(2SPACE)E1= START(7		7170 IF B\$=CHR\$(145)THEN 7250	<131>
SPACE)E7= MENUE(2SPACE)#####(RVOFF)"		7180 IF B\$=CHR\$(13)THEN 7800	<055>
:PRINT	<219>	7185 IF B\$=CHR\$(133)OR B\$=CHR\$(134)OR B\$=C	
6020 E\$="":GET E\$:IF E\$=""THEN 6020	<084>	HR\$(135)THEN 7300	<088>
6030 IF E\$=CHR\$(136)THEN 1000	<093>	7190 IF B\$=CHR\$(136)THEN 1000	<215>
6040 IF E\$<>CHR\$(133)THEN 6020	<209>	7195 GOTO 7150	<247>
6090 M=0:MA=-1	<204>	7200 REM *** INPUT +1 *****	<136>
6100 OPEN 1,4,7	<247>	7210 K=K+1:IF K>64 THEN K=0	<192>
6110 PRINT#1,K0\$;BR\$;"DATEI- NAME(9SPACE):		7220 I\$(K)=C\$:GOTO 7700	<124>
" ;F\$	<075>	7250 REM *** INPUT -1 *****	<127>
6115 PRINT#1,"GESAMT - AUSWERTUNG : ";EE\$:		7260 K=K-1:IF K<0 THEN K=64	<054>
PRINT#1	<063>	7270 I\$(K)=C\$:GOTO 7700	<174>
6120 FOR E=0 TO 512	<207>	7300 REM *** MODUS *****	<035>
6125 P1=VAL(MID\$(0\$(E),14,3)):IF P1>7 THEN		7310 IF B\$=CHR\$(133)THEN C\$="1":FB\$=" GESE	
6128 6128	<076>	TZT":GOTO 7270	<005>
6127 IF MID\$(0\$(E),10,1)="I"AND P1>M THEN		7320 IF B\$=CHR\$(134)THEN C\$="0":FB\$=" GESE	
M=P1	<072>	TZT":GOTO 7270	<017>
6128 IF 0\$(E)<>" "THEN MA=E	<178>	7330 IF B\$=CHR\$(135)THEN C\$=" ":FB\$="(8SPA	
6130 NEXT	<042>	CE)":GOTO 7270	<030>
6131 IF MA<0 THEN PRINT#1,"EIN PROGRAMM V		7690 REM *** ANZEIGE INPUT *****	<136>
ORHANDEN !":PRINT#1:CLOSE 1:GOTO 1000	<217>	7700 PRINT CHR\$(19)" (BDOWN,RVSON)INPUT(2SP	
6132 C=0:REM *****		ACE) (I) (2SPACE): (SPACE,RVOFF,4SPACE,4	
*	<022>	LEFT)"LEFT\$(BL\$,3-LEN(STR\$(K)));K;	<066>
6135 IF C>(2*(M+1))-1 THEN 6340	<033>	7710 IF I\$(K)=" "THEN PRINT" (RVSON,2SPACE,R	
6140 FOR L=0 TO 64:I\$(L)=" ":M\$(L)=" ":Q\$(		VOFF,SPACE,LEFT,3SPACE,RVSON,SPACE,RV	
L)=" ":NEXT	<123>	OFF)";FB\$:GOTO 7150	<040>
6150 Z=C:FOR J=M TO 0 STEP-1	<238>	7720 PRINT" (RVSON,2SPACE,RVOFF,SPACE,LEFT,	
6160 IF Z<2J THEN I\$(J)="0":GOTO 6180	<120>	SPACE)":I\$(K)"; (SPACE,RVSON,SPACE,RVO	
6170 Z=Z-2J:I\$(J)="1"	<195>	FF)";FB\$	<105>
6180 NEXT	<094>	7730 GOTO 7150	<018>
6200 GOSUB 9000	<124>	7800 PRINT CHR\$(147)K0\$;BL\$;"INPUT - UEBER	
6240 K\$=RIGHT\$(STR\$(C),3)	<185>	SICHT (DUALWERT) ":PRINT	<032>
6250 PRINT#1,Z1\$;Z2\$; " ;DEZ":PRINT#1,Z3\$;Z		7810 FOR K=0 TO 64	<199>
4\$	<040>	7815 FB\$=STR\$(K):FB\$=MID\$(FB\$,2,2):FB\$=LEF	
6280 Q\$="":M\$="":I\$="":FOR D=FB TO 0 STEP-		T\$(BL\$,2-LEN(FB\$))+FB\$	<178>
1	<143>	7820 PRINT" I(;FB\$;)= "I\$(K),	<129>
6290 Q\$=Q\$+Q\$(D):M\$=M\$+M\$(D):I\$=I\$+I\$(D)	<163>	7830 NEXT	<220>
63			

```

8330 IF Q$(L)="" THEN Q$(L)=" " <175>
8340 IF M$(L)="" THEN M$(L)=" " <137>
8350 IF I$(L)="" THEN I$(L)=" " <099>
8370 Q$=Q$+Q$(L):M$=M$+M$(L):I$=I$+I$(L) <220>
8380 NEXT L <242>
8390 PRINT#1,"INPUT (3SPACE) I ";I$ <076>
8400 PRINT#1,"MARKER (2SPACE) M ";M$ <244>
8410 PRINT#1,"OUTPUT (2SPACE) Q ";Q$:PRINT#1 <193>
8420 CLOSE 1:GOTO 1000 <204>
9000 REM *** COMPILER ***** <241>
9010 H=0:L=0:F3=0 <107>
9060 REM <232>
9062 GET B$:IF B$=CHR$(136) THEN F3=2:CLOSE <104>
1:RETURN <185>
9070 IF L>MA THEN RETURN <159>
9080 IF O$(L)="" OR MID$(O$(L),7,1)=CHR$(59) <128>
) THEN 20000 <174>
9090 IF E$="" THEN PRINT LEFT$(O$(L),4), <203>
9100 O1$=MID$(O$(L),7,1):O2$=MID$(O$(L),10, <196>
1):O3$=MID$(O$(L),14,3) <120>
9105 O4$=MID$(O$(L),11,1):REM NEGATION <208>
9110 : <212>
9120 IF O1$="J" AND O2$=" " THEN L=VAL(O3$): <216>
GOTO 9060 <091>
9122 : <102>
9125 IF O1$="J" AND O2$="J" AND H=0 THEN L=V <188>
AL(O3$):GOTO 9060 <198>
9126 : <172>
9130 : <202>
9140 IF O1$<>" " THEN 9200 <125>
9150 IF O4$="J" AND H=0 THEN H=1:GOTO 9170 <095>
9160 IF O4$="J" AND H=1 THEN H=0 <115>
9170 IF O2$="J" THEN Q$(VAL(O3$))=RIGHT$(ST <133>
R$(H),1):H=0:GOTO 20000 <149>
9180 IF O2$="J" THEN M$(VAL(O3$))=RIGHT$(ST <155>
R$(H),1):H=0:GOTO 20000 <092>
9190 : <009>
9200 IF O1$<>"J" THEN 9270 <167>
9210 IF O2$="J" THEN HH=VAL(I$(VAL(O3$))) <185>
9220 IF O2$="J" THEN HH=VAL(M$(VAL(O3$))) <203>
9230 IF O2$="J" THEN HH=VAL(Q$(VAL(O3$))) <219>
9240 IF O4$="J" THEN GOSUB 52000 <192>
9250 H=H OR HH:GOTO 20000 <162>
9260 : <220>
9270 IF O1$<>"J" THEN 9340 <237>
9280 IF O2$="J" THEN HH=VAL(I$(VAL(O3$))) <255>
9290 IF O2$="J" THEN HH=VAL(M$(VAL(O3$))) <017>
9300 IF O2$="J" THEN HH=VAL(Q$(VAL(O3$))) <033>
9310 IF O4$="J" THEN GOSUB 52000 <079>
9320 H=H AND HH:GOTO 20000 <161>
9330 : <242>
9340 IF O1$<>"J" THEN 9410 <038>
9350 IF O2$="J" THEN HH=VAL(I$(VAL(O3$))) <176>
9360 IF O2$="J" THEN HH=VAL(M$(VAL(O3$))) <042>
9370 IF O2$="J" THEN HH=VAL(Q$(VAL(O3$))) <114>
9380 IF O4$="J" THEN GOSUB 52000 <200>
9390 IF H=HH THEN H=0:GOTO 20000 <123>
9400 IF H<>HH THEN H=1 <102>
9410 : <150>
20000 L=L+1:GOTO 9060 <220>
21000 REM *** PROGRAMME LOESCHEN ***** <195>
21010 F2=2:P1=0:F3=-1 <047>
21020 PRINT CHR$(147)" {RVSON,3SPACE} ANWEIS <050>
UNGS- PROGRAMME LOESCHEN !!! {3SPACE, <091>
RVOFF}";BL$ <244>
21030 PRINT BR$ {RVSON} E1 {RVOFF}=LOESCHEN <194>
{3SPACE,RVSON} E5 {RVOFF}=WEITER {3SPACE, <148>
RVSON} E7 {RVOFF}=MENUE {5SPACE,RVSON, <148>
SPACE,RVOFF}";BR$ <148>
21040 CLOSE 2:CLOSE 15:GOSUB 53010 <148>
21045 GET E$:IF E$=CHR$(136) THEN 1000 <248>
21050 IF F3=-1 THEN 1000 <017>
21060 GOTO 21000 <167>
22000 REM *** PROGRAMM LOESCHEN *****
22010 CLOSE 15:OPEN 15,8,15,"S:"+F$:CLOSE <047>
15:OPEN 15,8,15:F3=0: <050>
22020 IF P1=0 THEN P1=1:PRINT {RVSON}"F$ { <091>
RVSON}";:GOTO 22040 <244>
22030 P1=0:PRINT:PRINT {UP,20RIGHT}"; {RVSON} <194>
"F$ {RVOFF}" <148>
22040 FOR L=1 TO 1500:NEXT:F$="" <248>
22050 GOTO 21000 <017>
50000 REM *** FLOPPY- TEST ***** <167>
50010 OPEN 15,8,15:INPUT#15,FA,FB$,FC,FD <047>
50020 IF FA=0 THEN F1=0:GOTO 50100 <050>
50030 PRINT CHR$(147)KO$;BR$ <248>
50040 PRINT "FLOPPY - FEHLER ":PRINT:PRIN <017>
T FA;" "FB$ "FC" "FD:PRINT:PRINT <089>
PRINT "BITTE FEHLER BEHEBEN UND (16SP <16SP
ACE,RVSON) RETURN {RVOFF,SPACE} DRUECKE <230>
N !":PRINT:PRINT <180>
50060 GET EE$:IF EE$="" THEN 50060 <204>
50070 IF EE$<>CHR$(13) THEN 50060 <026>
50080 F1=1 <131>
50100 CLOSE 15:RETURN <138>
51000 REM *** ABFRAGE WEITER / STOP *** <210>
51010 Z=0 <076>
51020 PRINT#1,CHR$(19)" {23DOWN}"; <176>
51030 PRINT#1,{RVSON,SPACE}SPACE = WEITER <188>
{8SPACE}E7 = MENUE {7SPACE,RVSON}"; <214>
51050 E$="" :GET E$:IF E$="" THEN 51050 <036>
51060 IF E$=CHR$(136) THEN CLOSE 1:GOTO 100 <115>
0 <057>
51070 IF E$<>CHR$(32) THEN 51050 <077>
51080 PRINT#1,CHR$(19):FOR K=1 TO 24:PRINT <005>
#1,BL$;:NEXT <028>
51090 PRINT#1,CHR$(19);:RETURN <237>
51990 REM *** NEGATION ***** <088>
52000 IF HH=0 THEN HH=1:GOTO 52020 <062>
52010 IF HH=1 THEN HH=0 <080>
52020 RETURN <157>
52990 REM *** DIRECTORY LESEN ***** <146>
53000 PRINT CHR$(147)" {GREY 1}";:CLOSE 15: <243>
CLOSE 2 <229>
53002 IF F2=0 THEN PRINT {RVSON,9SPACE}DIR <173>
ECTORY - UEBERSICHT : {7SPACE,RVOFF}" <255>
;BL$; <095>
53004 IF F2=0 THEN PRINT BR$; {RVSON}E1 {RV <139>
OFF}=LADEN {6SPACE,RVSON}E5 {RVOFF}=WE <115>
ITER {4SPACE,RVSON}E7 {RVOFF}=MENUE {4S <155>
PACE,RVSON,SPACE,RVOFF}";BR$ <190>
53005 IF F2=1 THEN PRINT BR$; {RVSON,SPACE <050>
,RVOFF,7SPACE}ANWEISUNGEN SPEICHERN <070>
! {8SPACE,RVSON,SPACE,RVOFF}";BR$ <175>
53007 IF F2=1 THEN PRINT {RVSON,9SPACE}DIR <072>
ECTORY - UEBERSICHT : {7SPACE,RVOFF}" <099>
;BL$; <004>
53008 OPEN 15,8,15,"I":CLOSE 15:GOSUB 5000 <081>
0:IF F1=1 THEN 1200 <011>
53010 OPEN 15,8,15,"I":OPEN 2,8,2,"#" <123>
53020 T=18:S=1:P1=0 <052>
53030 PRINT#15,"B-R";2;0;T;S <116>
53040 PRINT#15,"B-P";2;0 <086>
53050 GET#2,X$:IF X$="" THEN X$=CHR$(0) <029>
53060 T=ASC(X$) <110>
53070 GET#2,X$:IF X$="" THEN X$=CHR$(0) <157>
53080 S=ASC(X$) <039>
53090 FOR X=0 TO 7 <177>
53100 PRINT#15,"B-P";2;X*32 <111>
53110 F$="" <123>
53120 FOR Y=0 TO 20 <052>
53130 GET#2,X$:IF X$="" THEN X$=CHR$(0) <116>
53140 IF ASC(X$)=160 THEN 53160 <086>
53150 IF Y>4 THEN F$=F$+X$ <029>
53155 IF Y<5 THEN V(Y)=ASC(X$) <004>
53160 NEXT <081>
53161 IF V(2)<>129 OR V(2)=0 THEN 53190 <011>
53162 IF P1=0 THEN P1=1:PRINT F$;:GOTO 531 <123>
70 <052>
53164 P1=0:PRINT:PRINT {UP,20RIGHT}";F$ <116>
53170 IF F2=1 THEN 53190 <086>
53180 GET E$:IF E$="" THEN 53180 <029>
53181 IF F2=2 AND E$=CHR$(133) THEN CLOSE 2 <110>
:CLOSE 15:GOTO 22000 <157>
53182 IF E$=CHR$(133) THEN CLOSE 15:CLOSE 2 <039>
:RETURN <177>
53184 IF E$=CHR$(136) THEN CLOSE 15:CLOSE 2 <111>
:F2=0:GOTO 1000 <134>
53186 IF E$=CHR$(135) THEN 53190 <067>
53188 GOTO 53180 <021>
53190 NEXT <083>
53200 IF T<>0 THEN 53030 <140>
53210 CLOSE 15:CLOSE 2:PRINT:IF F2=2 THEN <020>
RETURN <117>
53215 IF F2=0 THEN 53000 <221>
53219 IF P1=1 THEN PRINT <140>
53220 PRINT {GREY 1,RVSON}** ENDE VON DIRE <020>
CTORY ** {2SPACE}RETURN= MENUE {SPACE, <117>
RVOFF}";PRINT:PRINT <221>
53230 F$="" :INPUT DATENAME : ";F$ <140>
53240 IF F$="" THEN 1000 <117>
53250 RETURN <221>

```

Listing 1. »Digi-Controller« (Schluß)