



©FH Kiel, ZKW, Pressebilder (2014)

SIEMENS 2002

Vorbemerkung, Inhalt

SIEMENS 2002

Einführung ➤

Kernspeicher ➤

Bedientisch ➤

Handhabung ➤

Programmierung ➤

Programmbeispiele ➤

Beispiel 1 Manuell ➤

Beispiel 2 Zügig ➤

Beispiel 3 Zählschleife ➤

Vorbemerkung der Autorin Gabriele Sowada:

Dies ist eine Dokumentation zur **Rechenanlage Siemens 2002**.

Die Dokumentation habe ich 2011 unentgeltlich angefertigt für das Exponat im **Computermuseum der Fachhochschule Kiel**. Die Fakten habe ich nach bestem Wissen und Gewissen zusammengestellt. Einige Kleinigkeiten habe ich nachträglich korrigiert bzw. ergänzt.

Ich habe ab April 1969 für etwa ein Jahr an genau dieser Rechenanlage im Hahn-Meitner-Institut gearbeitet. Jetzt steht sie im Computermuseum der Fachhochschule Kiel.

Im Museum stehen die 9 Kapitel dieser Dokumentation als einzelne, selbstablaufende PowerPoint-Präsentationen auf der Medienstation am Exponat zur Verfügung. Hier sind sie in einer PDF-Datei zusammengefasst, dabei lassen sich die Kapitel einzeln ansteuern.

Als Hintergrundbild habe ich ein Foto von einem Stück **Magnetkernspeicher** dieser Anlage verwendet, eine CD hinter dem Kernspeicher liefert die Farbe.

Diese Dokumentation darf gern verwendet und verteilt werden, privat, in Lehre und Museen, aber darf nicht verändert werden und darf nicht für kommerzielle Zwecke genutzt werden.



SIEMENS 2002

Einführung



Datenverarbeitungsanlage 2002

S2002

Datenverarbeitungsanlagen vom Typ **Siemens 2002** wurden zwischen 1959 und 1966 in Serie hergestellt.



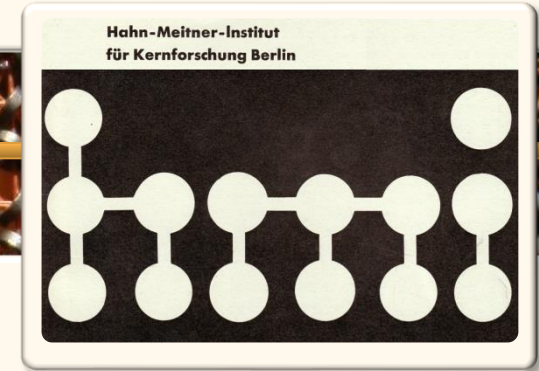
©Siemens AG, Pressebilder: Hannover 1959

Hier ausgestellt ist ein Teil der beiden Anlagen vom Typ Siemens 2002 aus dem **Hahn-Meitner-Institut** für Kernforschung in **Berlin-Wannsee**. In den 1960er-Jahren wurde sie dort im **Bereich Mathematik** betrieben.

Geschichte

Sie wurde institutsintern für die **Forschung** auf den Gebieten der Mathematik, Physik und Chemie genutzt. Externe, wie das Deutsche Institut für Wirtschaftsforschung (DIW), das Meteorologische Institut der Freien Universität, konnten **Rechenzeit mieten**.

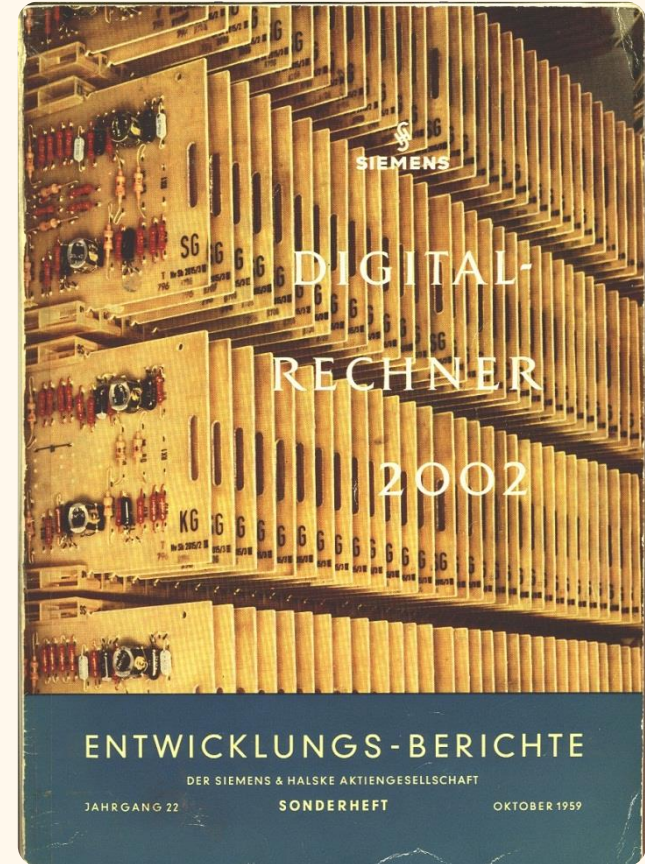
Die gesamte Anlage wurde 1970 nach dem Abbau an die **Dreilindenschule** in Berlin-Wannsee verschenkt, wo sie in einer **Informatik-AG** für den Unterricht modifiziert und mehr als 10 Jahre genutzt wurde.



Die Siemens 2002 war der **erste** kommerziell in Serie gebaute deutsche **Digital-Rechner in Transistor-Technik.**

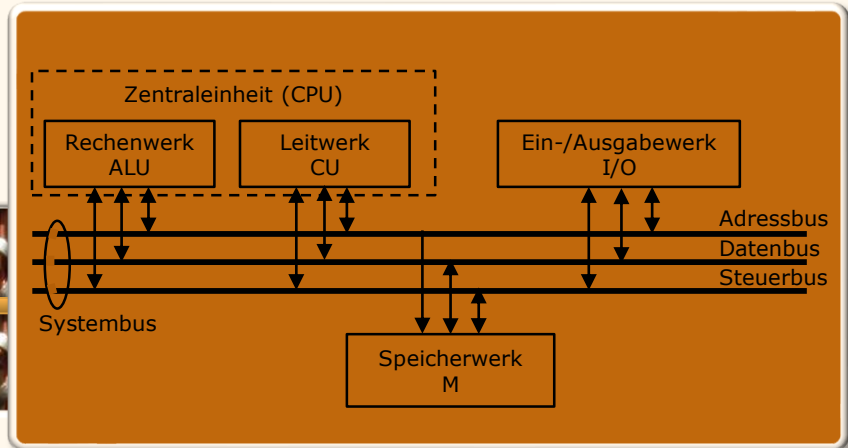
Technik

Sämtliche Schaltkreise waren im Bausteinprinzip aus steckbaren und mit Transistoren bestückten Baugruppen realisiert.



Die Siemens 2002 entsprach dem Konzept der **Von-Neumann-Architektur**.

Aufbau



Diese Architektur teilt die Rechanlage logisch und räumlich in verschiedene Komponenten auf und ermöglicht damit das Konzept des **Universalrechners**, der in seiner Struktur unabhängig ist von den zu lösenden Problemen.

Die Siemens 2002 konnte, je nach verwendetem **Programm**, in kommerziell-kaufmännischen Bereichen der **Industrie** eingesetzt werden, wie auch in technisch-wissenschaftlichen Bereichen der **Forschung**.

Die **Systemeinheit** bestand aus großen grauen Schränken, in denen sich alle **Werke** befanden. Sie waren im HMI in einem klimatisierten Raum untergebracht

Werke



Im **Speicherwerk** werden Daten **und** Programme gespeichert.

Im **Rechenwerk** werden die Rechenoperationen ausgeführt. Dem Rechenwerk vorgeschaltet ist der **Akkumulator**, das zentrale Register für alle Rechenoperationen. Das **Leitwerk** steuert den Programmablauf. Rechenwerk und Leitwerk bilden die **Zentraleinheit** (central processing unit, CPU).

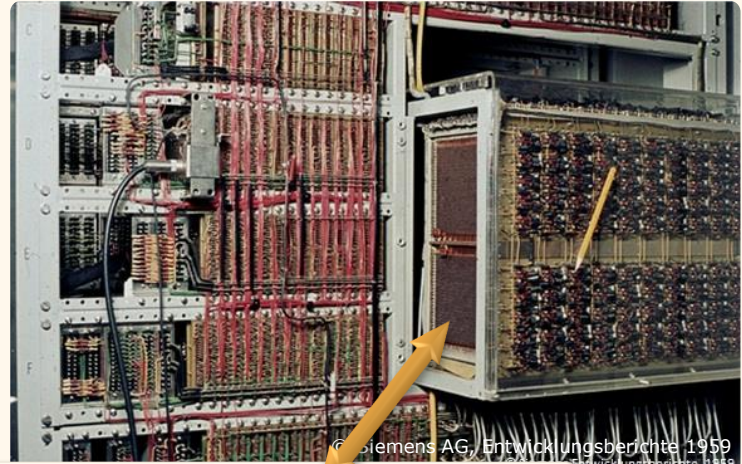
Über das **Ein-Ausgabewerk** werden Daten und Programme ein- und ausgegeben.

Der Hauptspeicher der Siemens 2002 war ein **Magnetkernspeicher**, in der 2. Ausbaustufe mit 5000 Maschinenworten à 52 Bits.

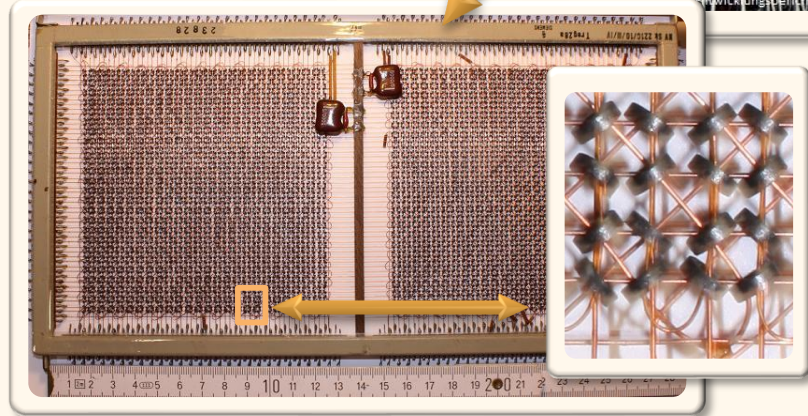
Speicher

2 mal 50 mal 50 = 5000 **Ferritkerne**, Ringe mit ca. 3 mm Außendurchmesser, waren als **Doppelmatrix** angeordnet und speicherten jeweils **1 Bit pro Kern**.

Als optionale Erweiterung gab es einen Trommelspeicher, der als Zubringerspeicher genutzt wurde.



© Siemens AG, Entwicklungsberichte 1959



Hauptschnittstelle Rechanlage – Anwender war der **Bedientisch** (Bedienerpult, Steuerpult).



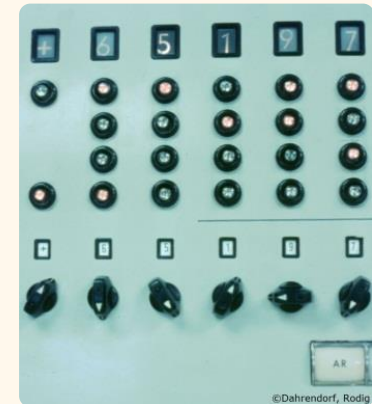
Bedientisch



Der Anwender (Benutzer) war i. A. gleichzeitig der Programm-Entwickler, Programmierer und Bediener.

Am Bedientisch wurde jede Aktion gestartet.

Der Programm-Ablauf konnte geprüft und überwacht werden (Tasten, Schalter, Glühlampen). Im Notfall war auch die manuelle Eingabe kleiner Programme möglich.



©Dahrendorf, Rodig

Wichtigster Datenträger bei der Arbeit mit der Siemens 2002 war der **gelbe 5-Kanal-Lochstreifen**.



©Dahrendorf, Rodig

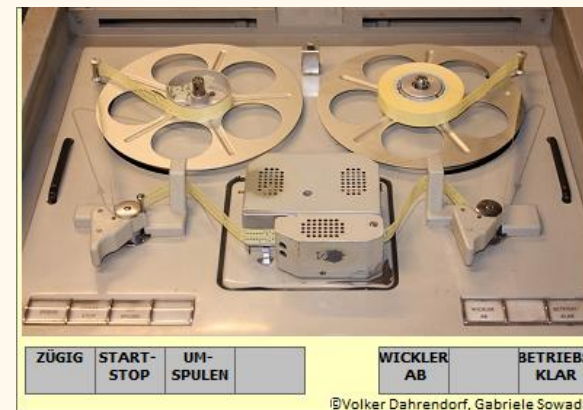


©Dahrendorf, Rodig

LS-Eingabe

Die Lochstreifen wurden hauptsächlich am **Fernschreiber** erstellt.

Über das Eingabegerät **Lochstreifen-Leser** wurden Programme im **Maschinencode** mit dem **Züigig-Modus direkt in den Kernspeicher** geladen. Daten wurden im Start-Stop-Modus eingelesen.



©Volker Dahrendorf, Gabriele Sowada



Als **mehrfach beschreibbares** und **auswechselbares** Speichermedium konnten **Magnetbänder** für Ein- und Ausgabe eingesetzt werden.

Magnetband

An die Siemens 2002 konnten **verschiedene Typen** von Magnetbandgeräten angeschlossen werden! Im HMI wurden 4 Geräte von IBM verwendet. Alle Magnetbandgeräte mussten damals unbedingt in einem **klimatisierten Raum** untergebracht werden!

Magnetbänder wurden zum Austausch und Sortieren großer Datenmengen, sowie für Programm-Bibliotheken genutzt.



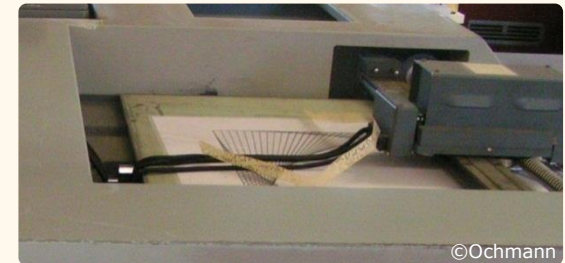


Übliches Ausgabemedium war der gelbe
5-Kanal-Lochstreifen.

Ein Lochstreifen konnte dabei höchst
Unterschiedliches zum Inhalt haben.

Ausgabe

Waren es Daten und Rechenergebnisse, konnten sie anschließend am
Fernschreiber ausgedruckt werden. Als Ergebnis eines Assemblier- oder
Compilierungslaufs war der Lochstreifen
beispielsweise wiederum **Eingabe für die S2002**.
Als Ergebnis eines Zeichenprogramms konnte der
Lochstreifen aber auch zur Steuerung des
Zeichengerätes **Zuse Graphomat Z64** dienen.





Zum direkten Ausdruck auf Papier war eine **Tabelliermaschine** angeschlossen. Sie druckte auf Endlospapier, Format Din-A3-quer, 120 Zeichen gleicher Breite pro Zeile.

Ausdrucke

Es gab nur Großbuchstaben, Ziffern und einige Sonderzeichen. Für jede Zeichenposition hatte die Maschine eine senkrecht stehende Stange mit allen verfügbaren Zeichen. Zum Drucken wurden immer erst für eine Zeile alle Stangen auf das richtige Zeichen eingestellt. Danach wurde die ganze Zeile mit einem Schlag gedruckt – ein wahrer Höllenlärm!

Später konnten auch **Schnelldrucker** angeschlossen werden. Bei diesen Geräten waren die Zeichen auf einer Druckkette angeordnet.



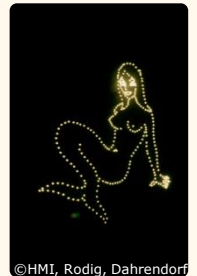


Für die punktweise grafische Anzeige von berechneten oder Messdaten gab es ein monochromes **Analog-Sichtgerät**.

Anzeige

Dieses Gerät war ein **reines Anzeigegerät** und hatte keinesfalls die Funktion späterer Monitore als Bedienungsschnittstelle!

So wie der Zuse Graphomat auch genutzt wurde, um Dekoratives für die Bürowände zu erzeugen, gab es ein Programm, das für spezielle Anlässe die Funktion des Anlogsichtgeräts mit attraktiveren als trockenen Messdaten demonstrierte.



©HMI, Rodig, Dahrendorf



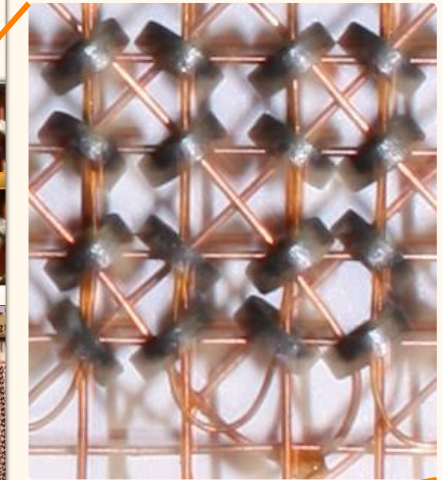
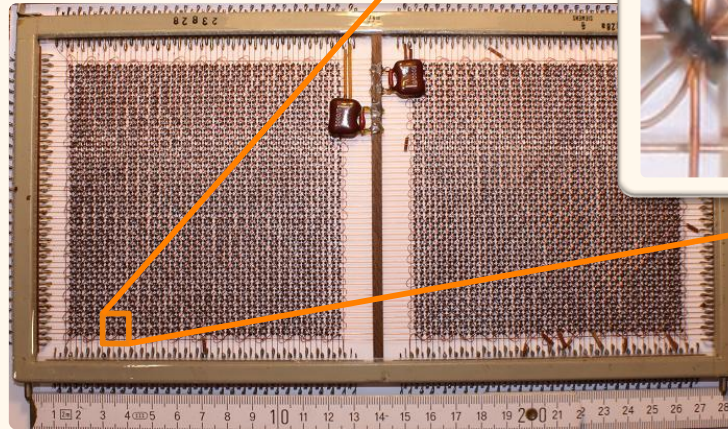
SIEMENS 2002

Kernspeicher

Das Hintergrundbild zeigt **Kernspeicher** (Magnetkernspeicher) aus der Datenverarbeitungsanlage **Siemens 2002**. Er wurde als Hauptspeicher verwendet und in Matrizen angeordnet.

Ferritkerne

2 mal 50 mal 50 = 5000
Ferritkerne, Ringe mit
ca. 3 mm Außen-
durchmesser, waren als
Doppelmatrix angeordnet
und speicherten jeweils
1 Bit pro Kern.



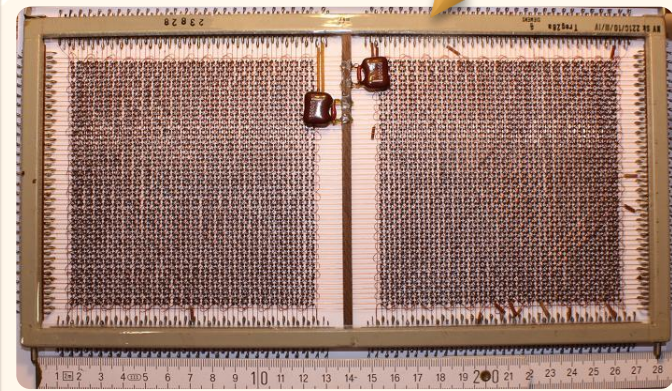
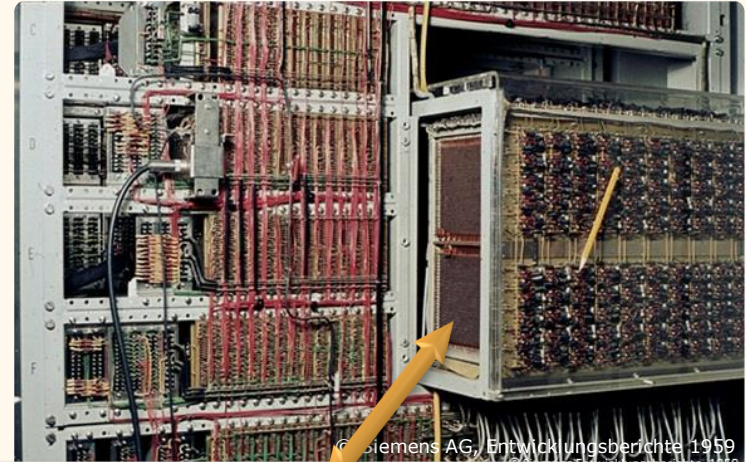
Ein **Maschinenwort** der Siemens 2002 bestand aus **52 Bits**, 52 dieser Matrizen stehen für 5000 Maschinenworte.

Maschinenwort

Ein Ferritkern kann, abhängig von der Magnetisierungsrichtung, den binären Wert 0 oder 1 darstellen.

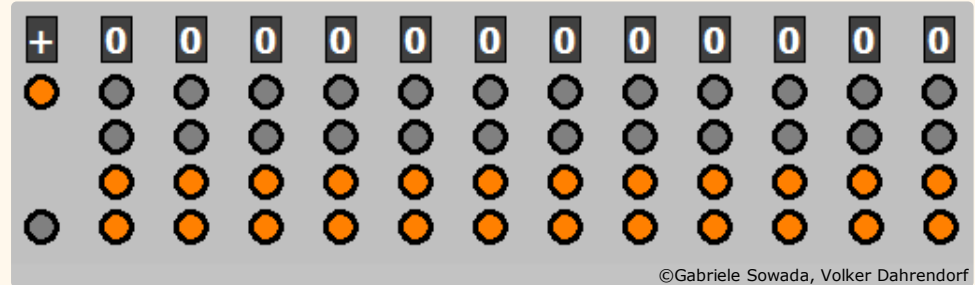
Zur Magnetisierung des Kerns ist ein Strom von 600 mA erforderlich, der durch die Koordinatenströme X und Y realisiert wird*.

**(Gerhard Ploch, 2009)*



Die **52 Bits** eines Maschinenwortes der Siemens 2002 waren in **13 Tetraden** gegliedert.

Tetraden



Jede Tetrade stellt eine **binär codierte Dezimalziffer** dar.
(BCD-Code, »binary coded decimal«)

12 Tetraden standen bei der S2002 für je eine Dezimalziffer, von der 13. Tetrade wurde je ein Bit für Vorzeichen und Parity-Kontrolle verwendet.

Die 4 **Bits** einer Tetrade wurden **parallel** übertragen, die **Tetraden** eines Maschinenwortes **seriell**.

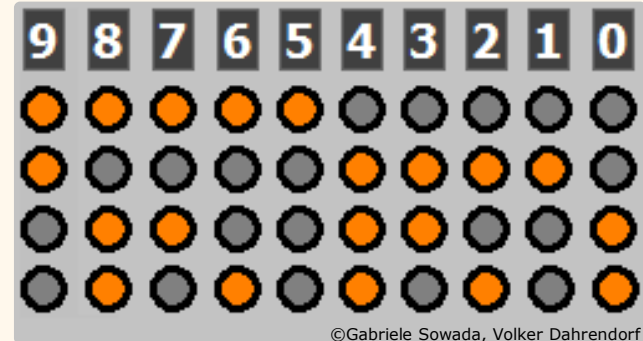
Verwendet wurde der **3-Exzess-Code** (auch Stibitz-Code genannt, nach George Robert **Stibitz**, 20. April 1904 – 31. Januar 1995).

3-Exzess-Code

Im 3-Exzess-Code wird zu jeder Tetrade des BCD-Codes eine 3 addiert (0011_2).

Es ergibt sich ein **symmetrisches** Bild.

Die Tetraden 0000_2 , 0001_2 , 0010_2 und 1101_2 , 1110_2 , 1111_2 sind unzulässig und werden **Pseudo-Tetraden** genannt.



Ein **Wort** im Speicher konnte auf unterschiedliche Weisen interpretiert werden.

- Als zwölfstellige **Festkommazahl** mit Vorzeichenstelle

Festkommazahl



Festkommazahl

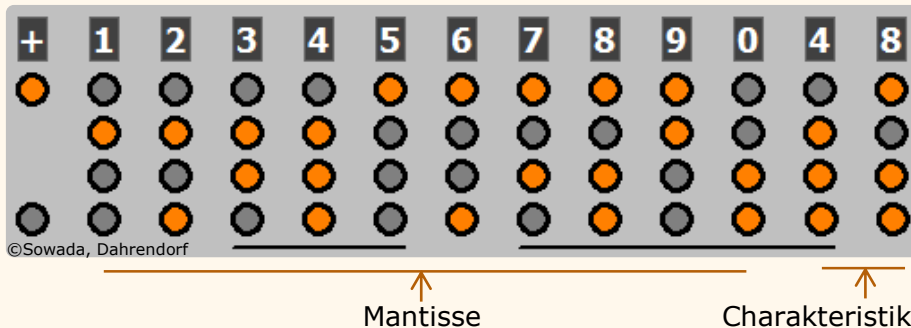
+0,123.456.789.048

Das Dezimalkomma wird vor der ersten Ziffer angesetzt.

Ein **Wort** im Speicher konnte auf unterschiedliche Weisen interpretiert werden.

- Als **Gleitkommazahl** mit Vorzeichenstelle, 10-stelliger Mantisse und 2-stelliger Charakteristik (Exponenten-Darstellung ohne Vorzeichenstelle)

Gleitkommazahl



Gleitkommazahl

+0,1234567890*10-2

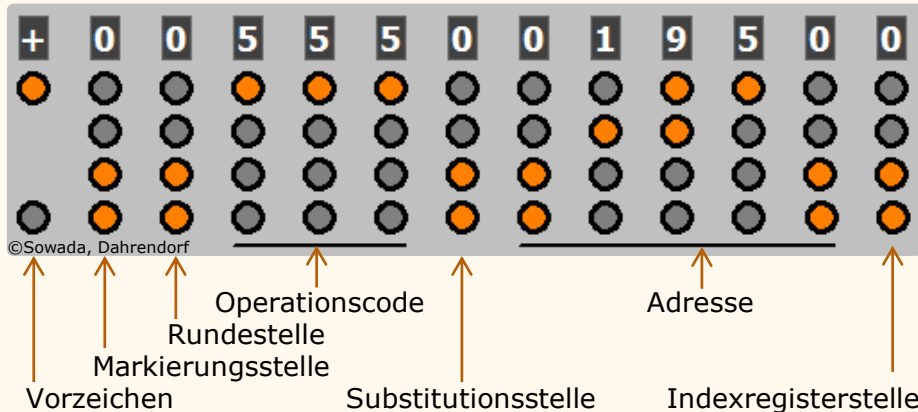
Mantisse: die Stellen nach dem Dezimalkomma

Charakteristik: Exponenten-Darstellung ohne Vorzeichenstelle; Charakteristik=Exponent+50

Ein **Wort** im Speicher konnte auf unterschiedliche Weisen interpretiert werden:

- als **Maschinenbefehl** mit 3-stelligem Operationscode, 5 Stellen für die Adresse (Ein-Adress-Befehle), dazu Vorzeichen-, Markierungs-, Runde-, Substitutions- und Indexregisterstelle

Befehl



Maschinenbefehl **TEP 1950**

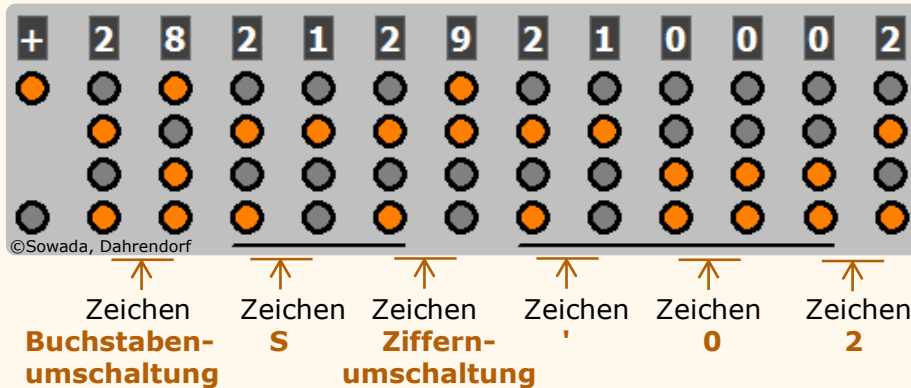
Transfer des **Inhalts** des Wortes an der Speicheradresse 1950 in das **akkumulative Register**.

Waren Tetraden mit unzulässigen Werten für einen Befehl belegt, konnte ein Fehler ausgelöst werden.

Ein **Wort** im Speicher konnte auf unterschiedliche Weisen interpretiert werden:

- als alphanumerische **Zeichenfolge**
(2 Dezimalziffern repräsentierten 1 Zeichen)

Zeichenfolge



Zeichenfolge **S'02**

Die Zeichencodierung folgte der sequentiellen Ausgabe auf Lochstreifen, war nicht eindeutig, sondern abhängig von einer vorangegangenen Buchstaben- oder Ziffern-Umschaltung.



SIEMENS 2002

Bedientisch

Hauptschnittstelle Rechenanlage – Anwender war der **Bedientisch** (Bedienerpult, Steuerpult).

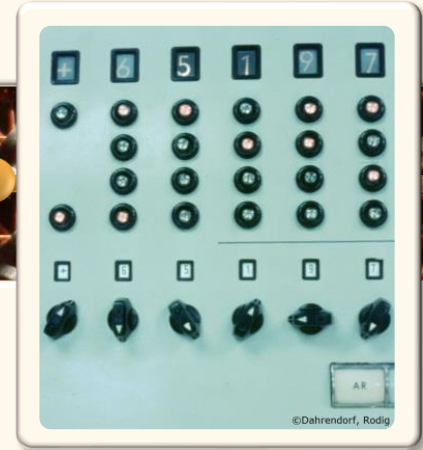
Bedientisch



Am Bedientisch wurde jede Aktion gestartet.

Am **Bedientisch** konnte mit Hilfe von Tasten, Schaltern und Glühlampen der Programm-Ablauf geprüft und überwacht werden. Im Notfall war auch die manuelle Eingabe kleiner Programme möglich.

Bedientisch



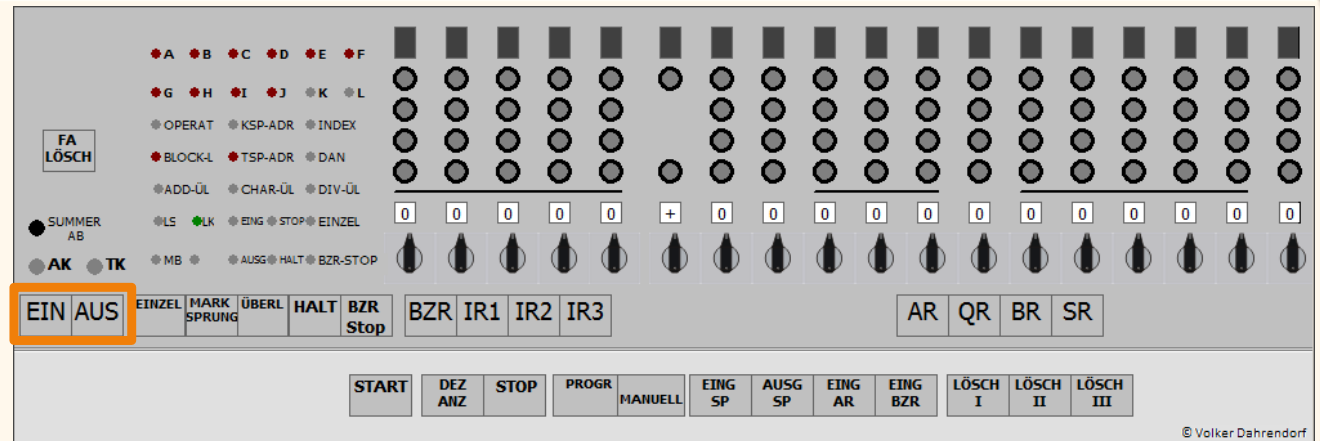
Der Anwender (Benutzer) war i. A. gleichzeitig der Programm-Entwickler, Programmierer und Bediener.

Der Bedientisch bot damals das zum Anfassen, was heute jeder Software-Entwickler virtuell als **IDE** (integrated development environment) kennt.

In den folgenden Erläuterungen wird der Bedientisch schematisch dargestellt. Verwendet wird dabei die Benutzerschnittstelle der **S2002-Emulation** von Volker Dahrendorf.

EIN/AUS

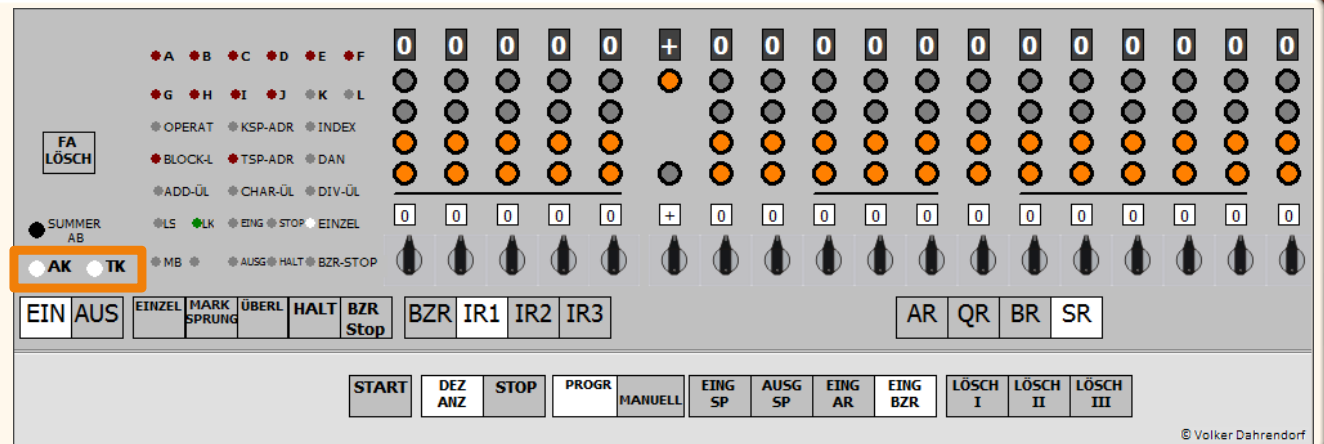
Mit den Tasten **EIN** bzw. **AUS** wurde die S2002 ein- bzw. ausgeschaltet.



Der Einschaltvorgang (die mehrstufige Spannungseinschaltung) dauerte einige Sekunden. Der Anlagen-Status war an zwei Anzeigelämpchen abzulesen.

EIN/AUS

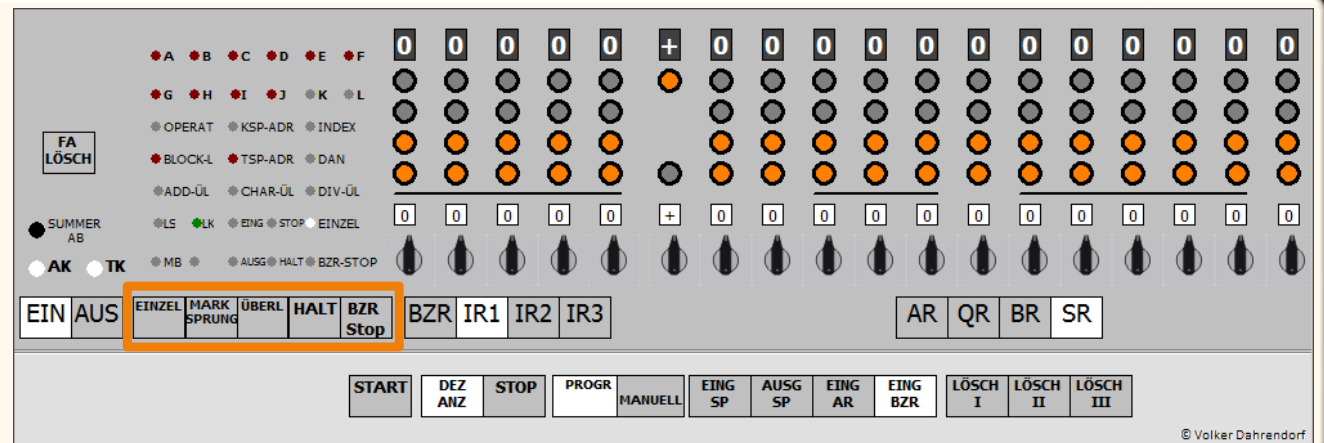
TK (Technik klar)
und
AK (Anlage klar)



© Volker Dahrendorf

Die nächste Tastengruppe dient der Steuerung von **Haltepunkten**, die Tasten konnten einzeln ein- und ausgeschaltet werden.

HALT

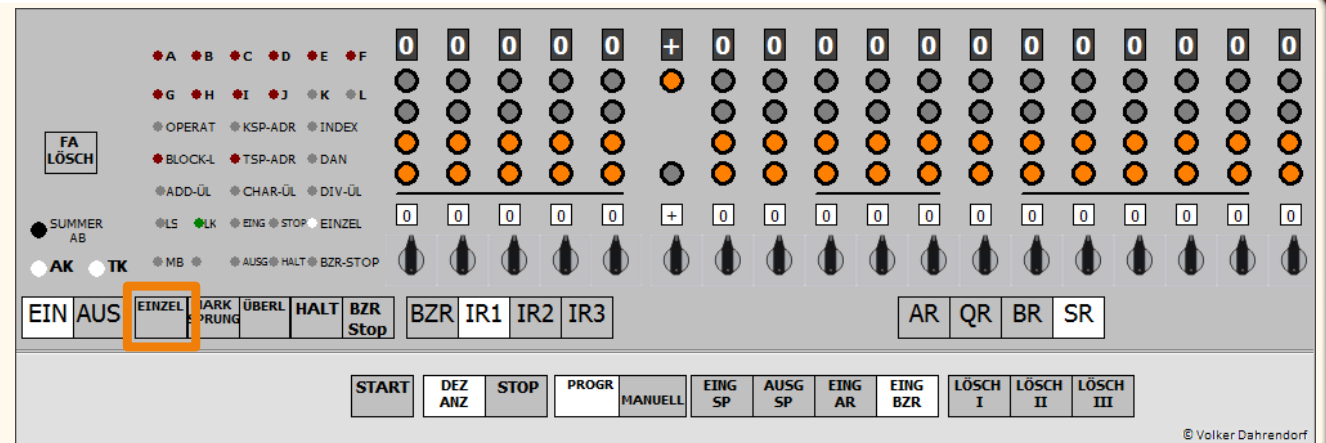


© Volker Dahrendorf

Die nächste Tastengruppe diente der Steuerung von **Haltepunkten**, die Tasten konnten einzeln ein- und ausgeschaltet werden.

HALT

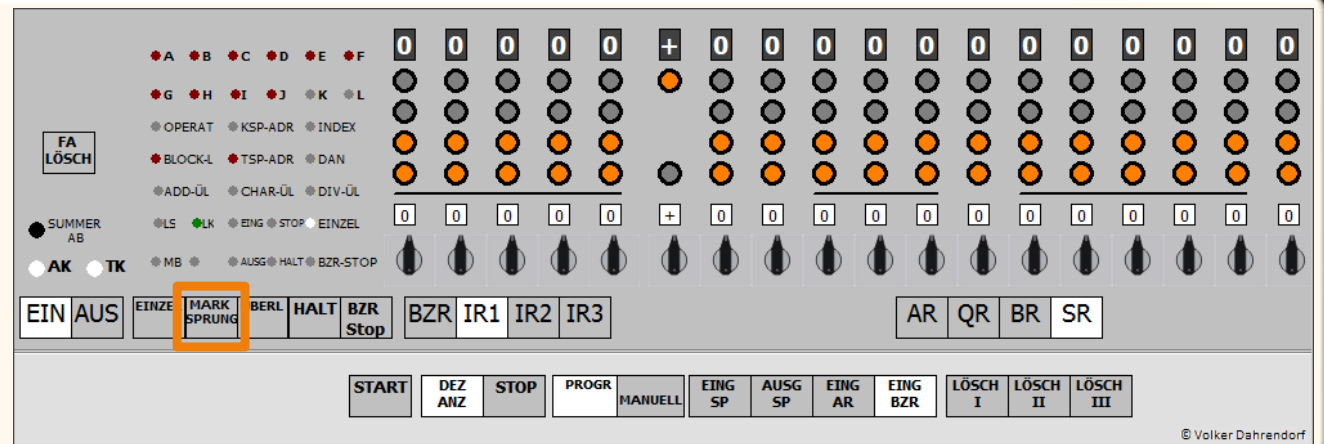
Die Taste **EINZEL** bewirkte eine Einzelschrittausführung.



Die nächste Tastengruppe diente der Steuerung von **Haltepunkten**, die Tasten konnten einzeln ein- und ausgeschaltet werden.

HALT

Die Taste **MARKSPRUNG** bewirkte einen programm-spezifischen Ablauf.

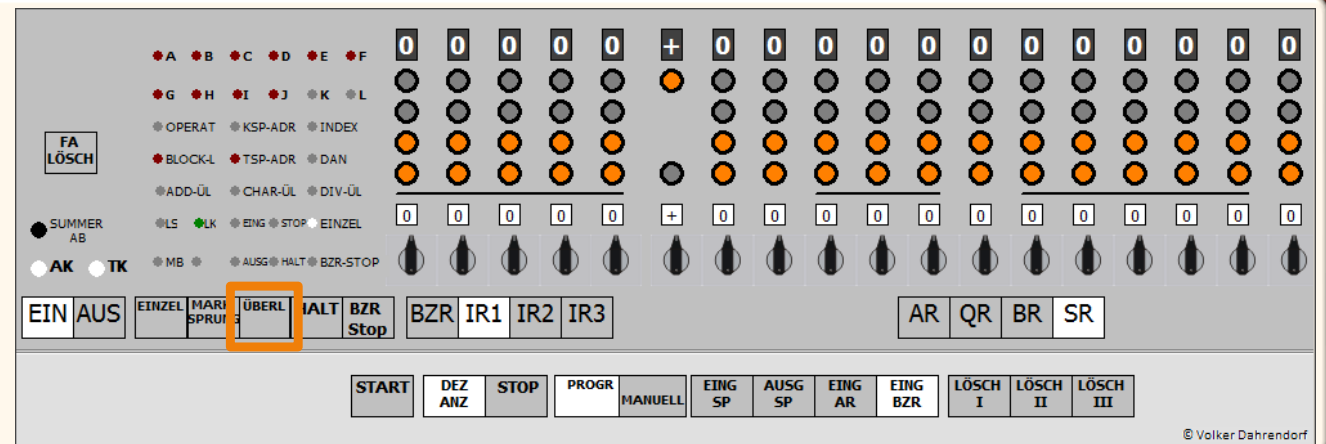


© Volker Dahrendorf

Die nächste Tastengruppe diente der Steuerung von **Haltepunkten**, die Tasten konnten einzeln ein- und ausgeschaltet werden.

HALT

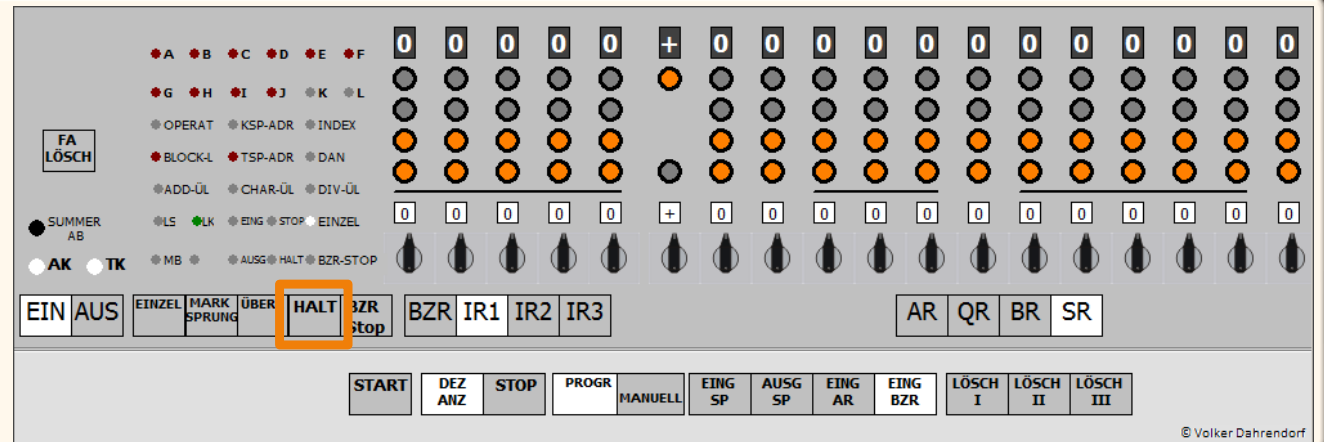
Die Taste **ÜBERL** unterdrückte den Programmstopp bei arithmetischem Überlauf.



Die nächste Tastengruppe diente der Steuerung von **Haltepunkten**, die Tasten konnten einzeln ein- und ausgeschaltet werden.

HALT

Die Taste **HALT** bewirkte, dass das Programm bei einem Halt-Befehl im Programmcode stoppte.

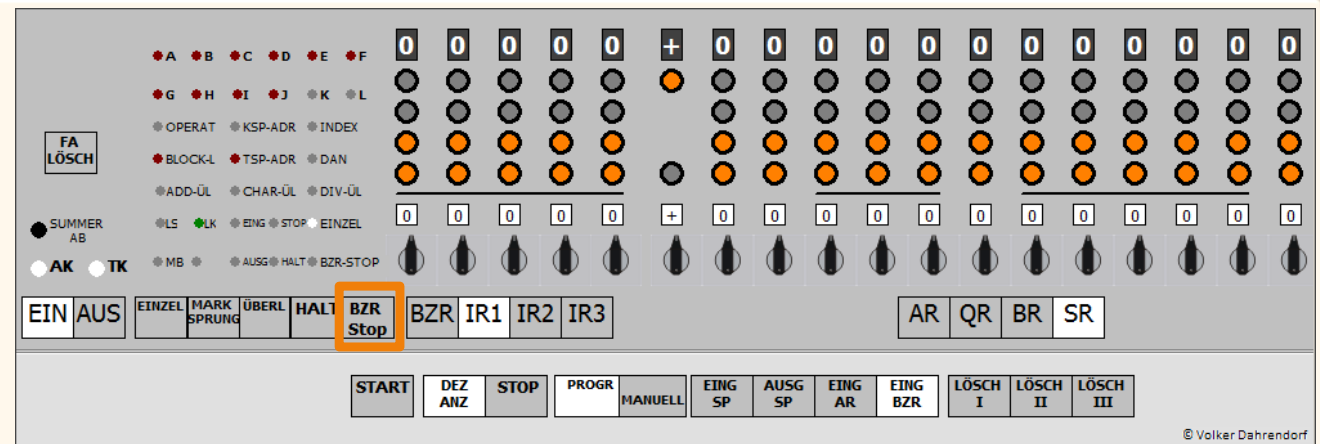


© Volker Dahrendorf

Die nächste Tastengruppe diente der Steuerung von **Haltepunkten**, die Tasten konnten einzeln ein- und ausgeschaltet werden.

HALT

Die Taste **BZRSTOP** bewirkte, dass das Programm an einer eingestellten Adresse stoppte.

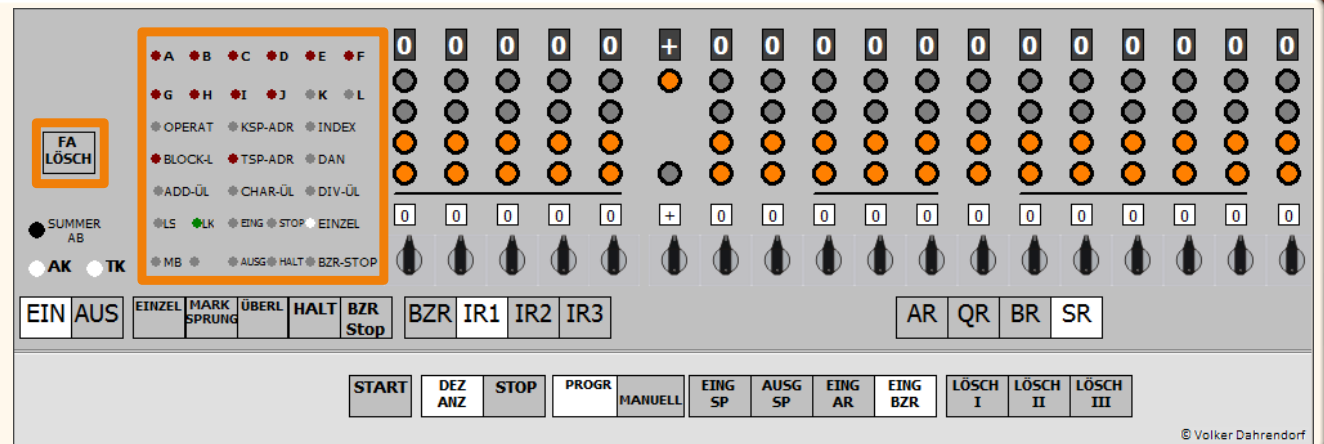


© Volker Dahrendorf

Im oberen linken Bereich wurden verschiedene **Hard- und Softwarefehler** sowie **Anlagenzustände** mittels kleiner Lampen angezeigt.

Zustand

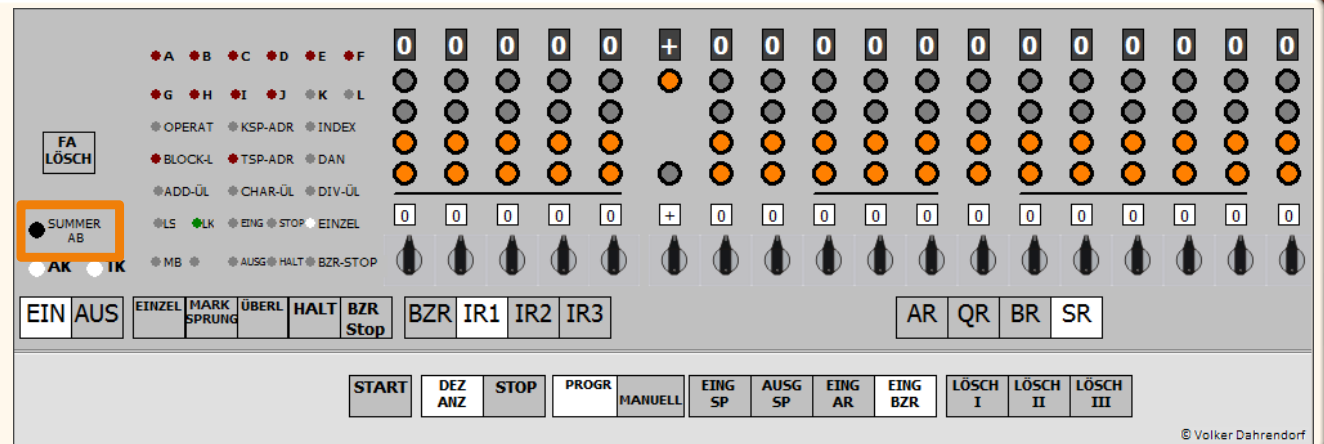
Die Taste **FALÖSCH** bewirkte das Löschen aller Fehleranzeigen.



Wurde ein **Fehleralarm** an der S2002 ausgelöst, ertönte zusätzlich ein Summer. Der Summer-Ton konnte einmalig durch Drücken auf SUMMER AB, permanent durch Drehen von SUMMER AB ausgeschaltet werden.

Zustand

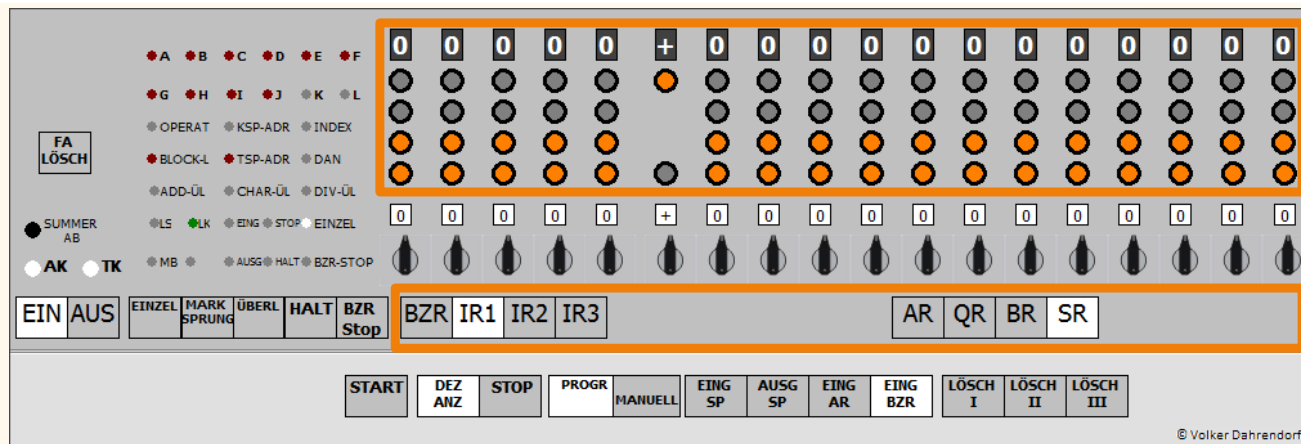
SUMMER AB
war eine
Kombination aus:
-Anzeigelampe
-Taster
-Drehschalter



Kontroll-Lämpchen und, im manuellen Modus darüber auch die **Dezimal-Anzeige**, zeigten den Speicherinhalt an. Darunter angeordnet sind die zugehörigen **Auswahl-Tasten**.

Anzeige

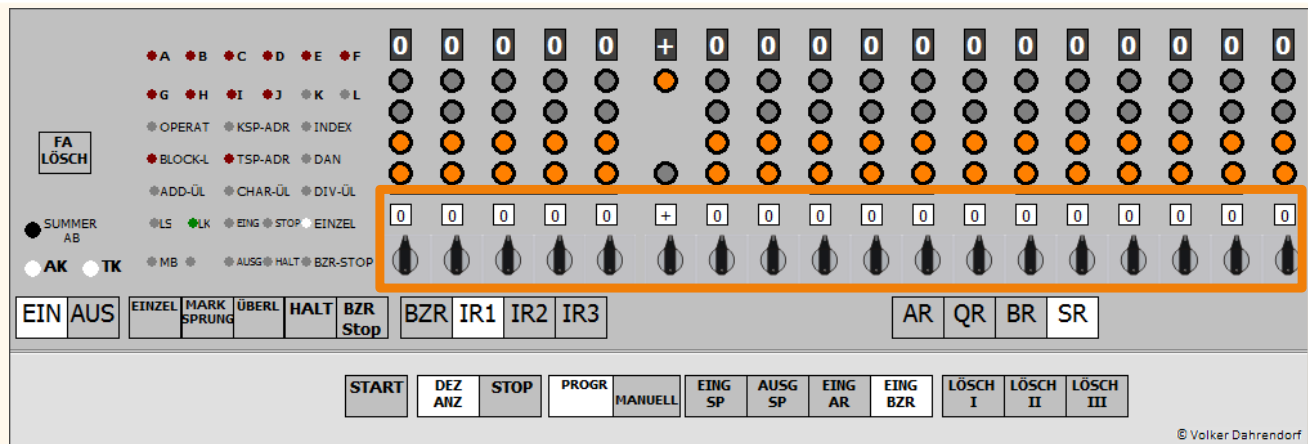
Die Tasten dienten der Auswahl, zeigten durch ihre Beleuchtung aber auch an, welcher Speicherbereich ausgewählt war.



Zwischen Anzeigelämpchen und Auswahltasten liegt das **Drehschalterfeld**.

Eingabe

Das Drehschalterfeld ermöglichte die manuelle Eingabe von Werten in die ausgewählten Speicherbereiche.

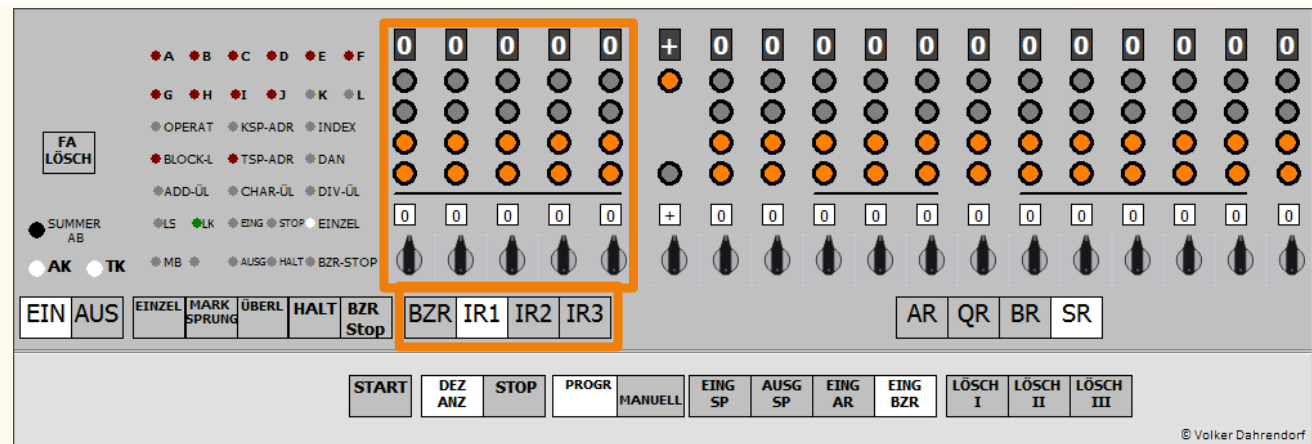


© Volker Dahrendorf

Links sind Tastengruppe, Drehschalterfeld und Anzeige für die **5-stelligen Register** angebracht. Es konnte immer nur eine Taste dieser Gruppe eingeschaltet sein, d. h. eines der Register ausgewählt werden.

5-stellig

BZR steht für das **Befehlszählregister**, die Tasten IR1, IR2, IR3 für die drei **Indexregister**.

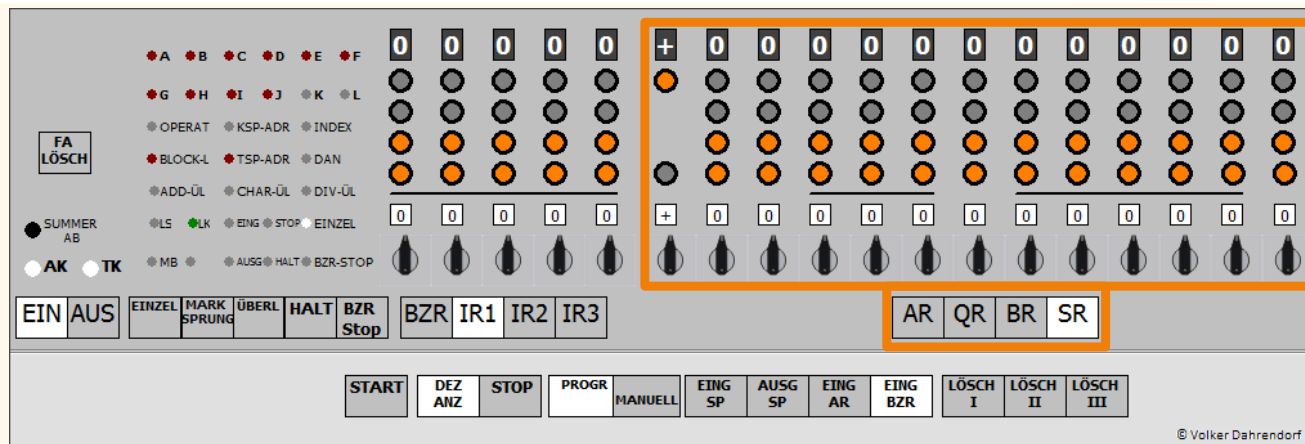


© Volker Dahrendorf

Rechts sind Tastengruppe, Drehschalterfeld und Anzeige für die **12-stelligen Register** angebracht. Es konnte immer nur eine Taste dieser Gruppe eingeschaltet sein, d. h. eines der Register ausgewählt werden.

12-stellig

AR steht für das
akkumulative
Register
(**Akkumulator**),
QR für das
Quotienten-
register.

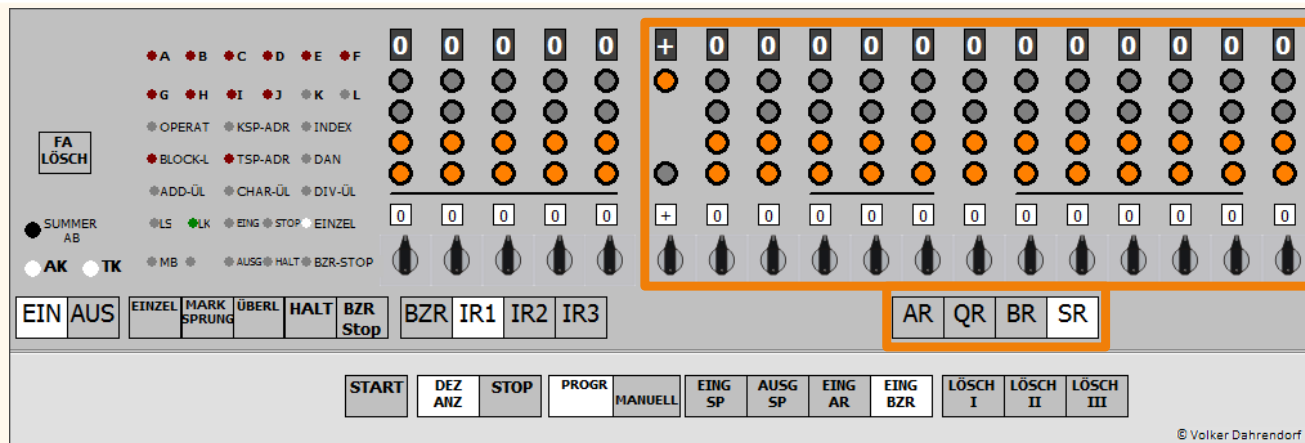


© Volker Dahrendorf

Rechts sind Tastengruppe, Drehschalterfeld und Anzeige für die **12-stelligen Register** angebracht. Es konnte immer nur eine Taste dieser Gruppe eingeschaltet sein, d. h. eines der Register ausgewählt werden.

12-stellig

BR steht für das **Befehlsregister**,
SR für das **Speicherregister**.

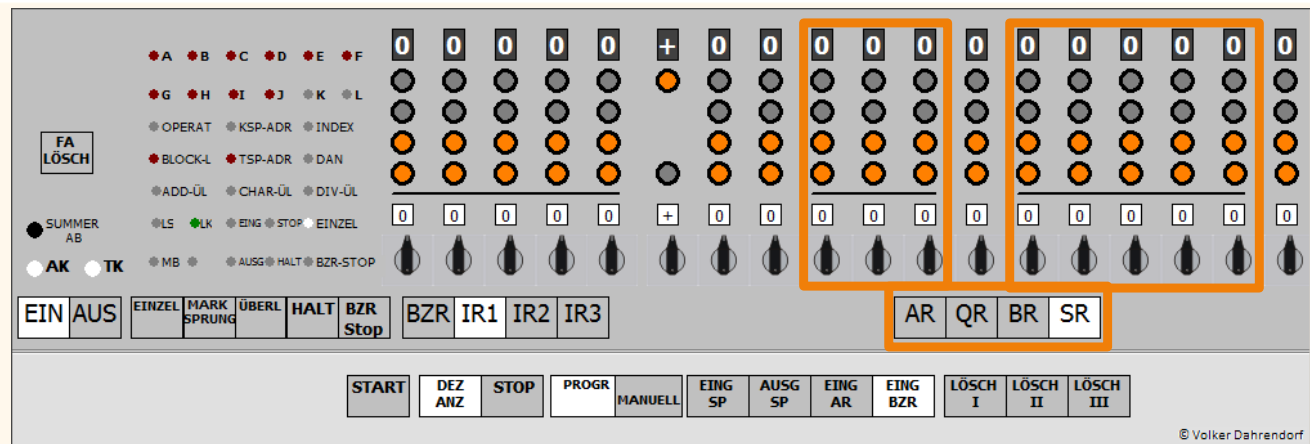


© Volker Dahrendorf

Rechts sind Tastengruppe, Drehschalterfeld und Anzeige für die **12-stelligen Register** angebracht. Es konnte immer nur eine Taste dieser Gruppe eingeschaltet sein, d. h. eines der Register ausgewählt werden.

12-stellig

Die schwarzen Linien gruppieren die 12 Stellen eines Wortes und ermöglichen so das schnelle Erfassen von Befehlen.

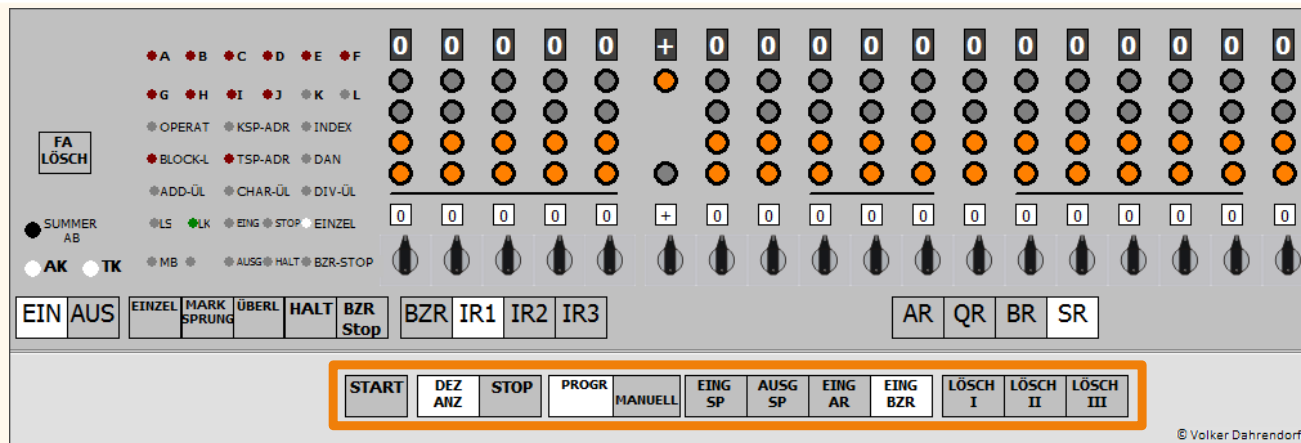


© Volker Dahrendorf

Die unterste Tastengruppe ist in die Tischplatte integriert und diente der **allgemeinen Programm-Steuerung**, sowie der **Definition des relevanten Speicherbereichs** für einen manuellen Datentransfer.

Steuerung

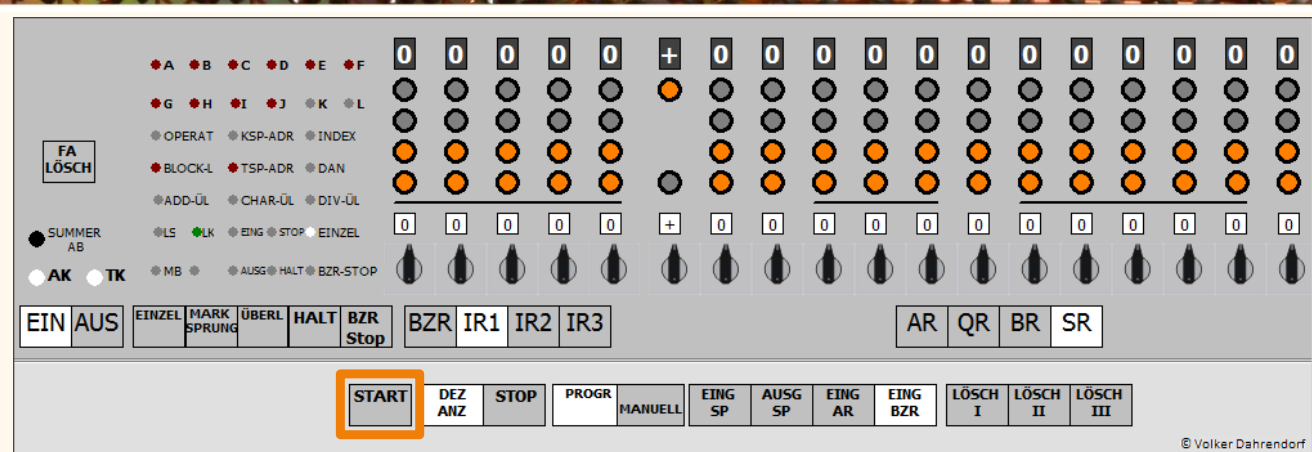
Die Tasten sind nach Funktion gruppiert.



Die unterste Tastengruppe ist in die Tischplatte integriert und diente der **allgemeinen Programm-Steuerung**, sowie der Definition des relevanten Speicherbereichs für einen manuellen Datentransfer.

Steuerung

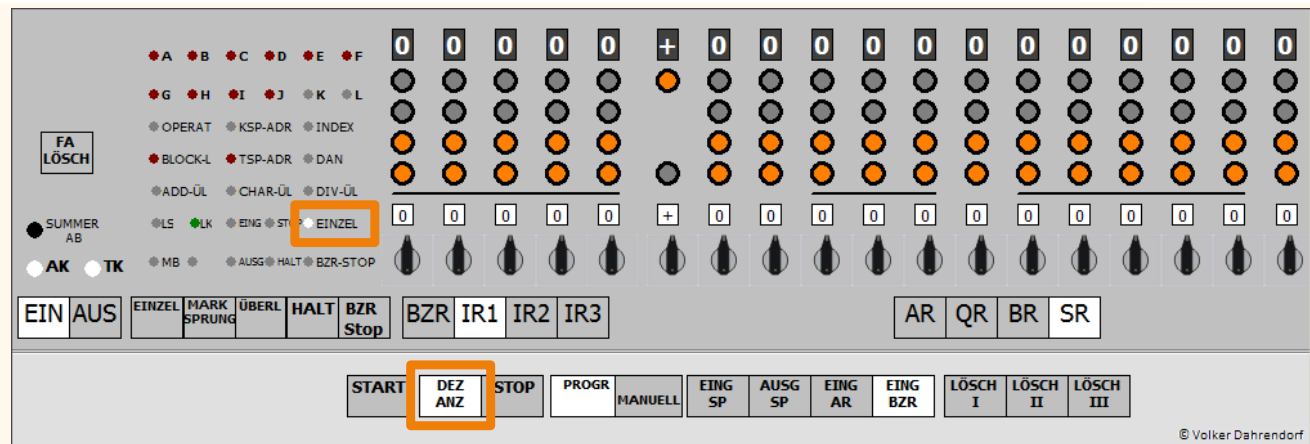
Die Taste **START** startete – entweder das eingegebene Programm oder einen manuell angewählten Vorgang.



Die unterste Tastengruppe ist in die Tischplatte integriert und diente der **allgemeinen Programm-Steuerung**, sowie der Definition des relevanten Speicherbereichs für einen manuellen Datentransfer.

Steuerung

DEZANZ schaltete die Dezimalanzeige ein, gleichzeitig wurde der Einzelschrittmodus aktiviert.

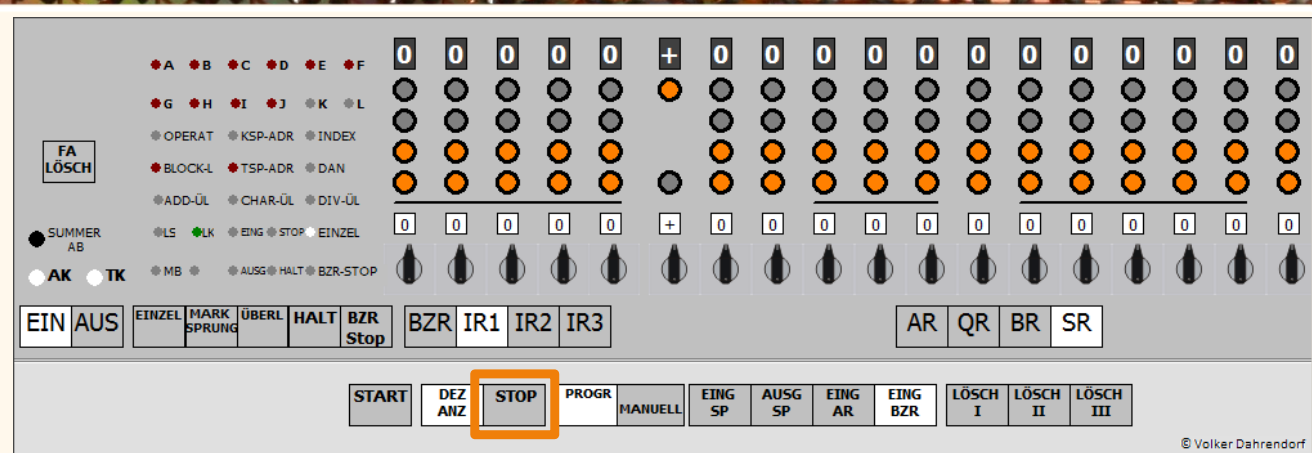


© Volker Dahrendorf

Die unterste Tastengruppe ist in die Tischplatte integriert und diente der **allgemeinen Programm-Steuerung**, sowie der Definition des relevanten Speicherbereichs für einen manuellen Datentransfer.

Steuerung

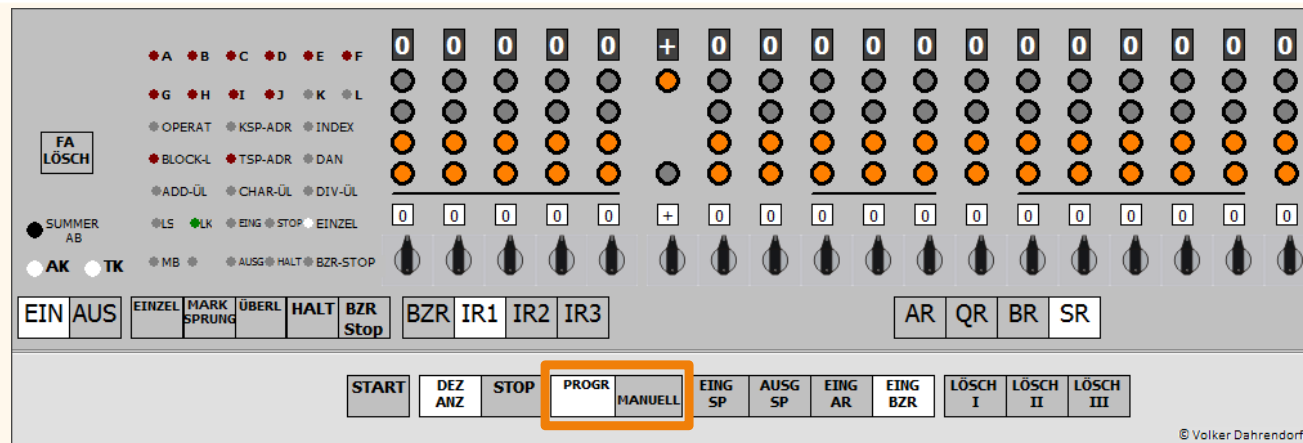
STOP unterbrach
ein laufendes
Programm.



Die unterste Tastengruppe ist in die Tischplatte integriert und diente der **allgemeinen Programm-Steuerung**, sowie der Definition des relevanten Speicherbereichs für einen manuellen Datentransfer.

Steuerung

PROGR und **MANUELL** schalteten die Betriebsart um. Es konnte immer nur eine der beiden Tasten eingeschaltet sein.



© Volker Dahrendorf

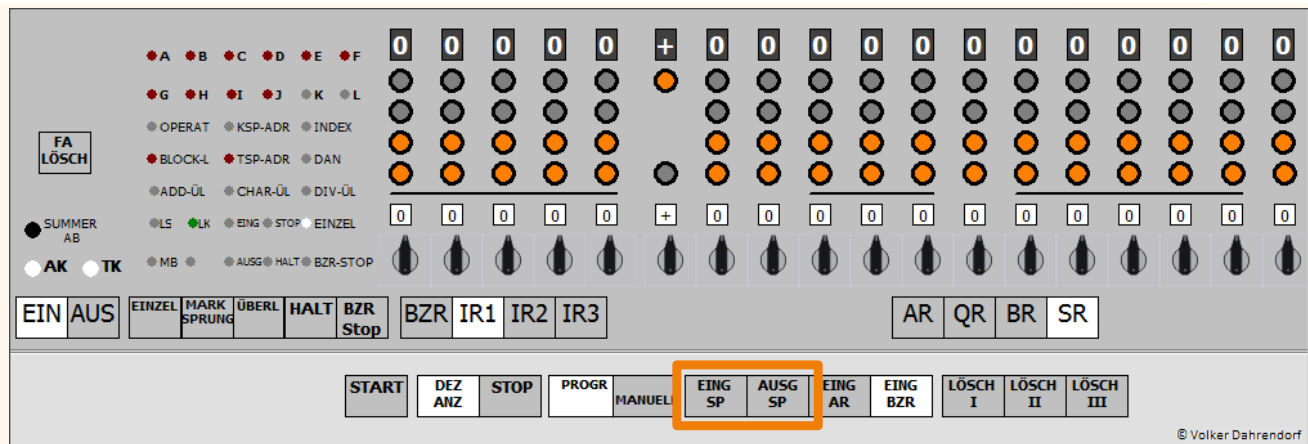
Die unterste Tastengruppe ist in die Tischplatte integriert und diente der allgemeinen Programm-Steuerung, sowie der **Definition des relevanten Speicherbereichs für einen manuellen Datentransfer.**

Transfer

EINGSP

definierte Eingabe
in den
Kernspeicher.

AUSGSP definierte
Auslesen aus dem
Kernspeicher.



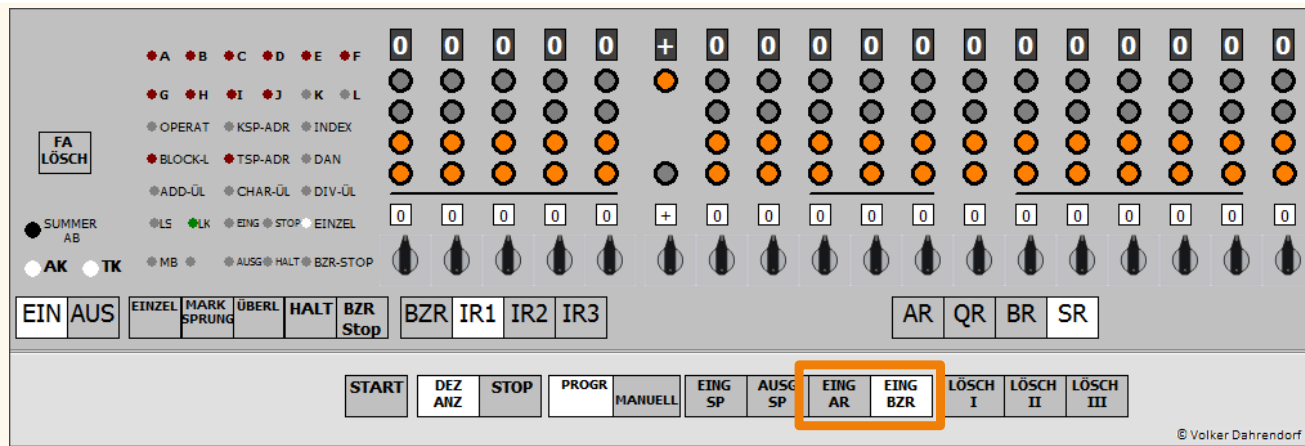
© Volker Dahrendorf

Die unterste Tastengruppe ist in die Tischplatte integriert und diente der allgemeinen Programm-Steuerung, sowie der **Definition des relevanten Speicherbereichs für einen manuellen Datentransfer.**

Transfer

EINGAR definierte
Eingabe in das
akkumulative
Register.

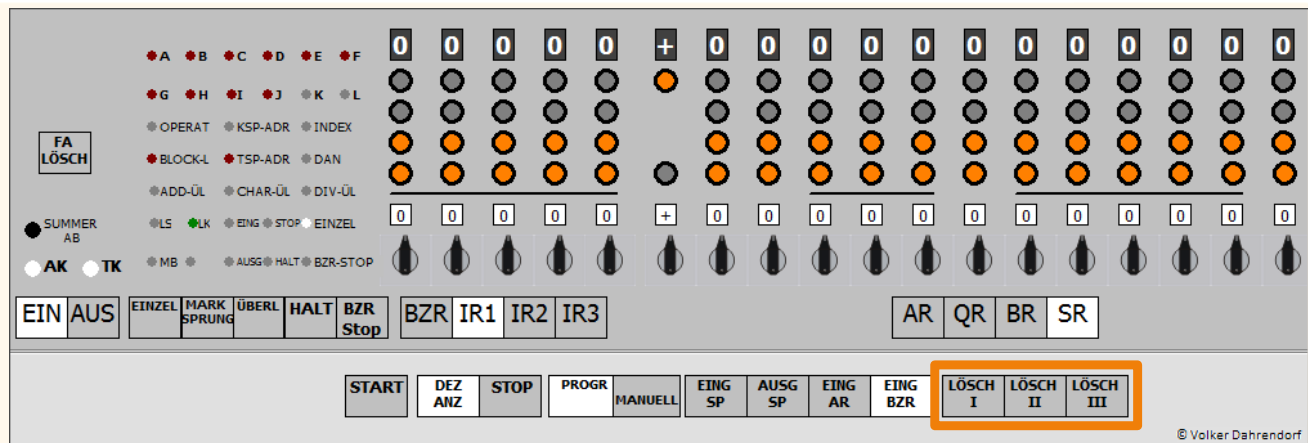
EINGBZR
definierte Eingabe
in das
Befehlszählregister.



Die unterste Tastengruppe ist in die Tischplatte integriert und diente der allgemeinen Programm-Steuerung, sowie der Definition des relevanten Speicherbereichs für einen manuellen Datentransfer.

Löschen

**LÖSCH I,
LÖSCH II und
LÖSCH III**
bewirkten das
Rücksetzen
verschiedener
Anlagenzustände.



© Volker Dahrendorf



SIEMENS 2002

Handhabung

Anmerkungen zur damaligen Handhabung
der hier ausgestellten Rechananlage.



Datenverarbeitungsanlage 2002

Standort

Hier ausgestellt ist ein Teil einer der beiden Anlagen vom Typ Siemens 2002
aus dem **Hahn-Meitner-Institut*** für Kernforschung in **Berlin-Wannsee**.
In den 1960er-Jahren wurde sie dort im **Sektor Mathematik** betrieben.

*2008 ging das Hahn-Meitner-Institut ein in das Helmholtz-Zentrum Berlin.

Anders als bei der abgebildeten Werbeaufstellung auf der Hannover-Messe war die Rechenanlage im HMI **in mehreren Räumen** untergebracht.

Aufstellung



Im Betrieb mussten Systemeinheit, Trommelspeicher und auch die Magnetbandgeräte in **klimatisierten Räumen** untergebracht werden, waren deshalb getrennt von Bedientisch mit Lochstreifen-Leser und -Stanzer, Analogsichtgerät und Tabelliermaschine.

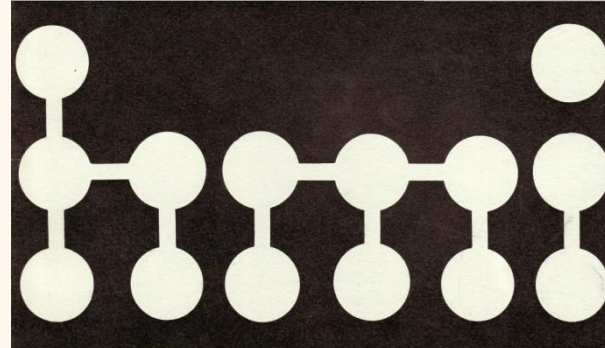
Fernschreiber, die zwar für die Herstellung der Lochstreifen, aber nicht für den direkten Betrieb erforderlich waren, standen in separaten Räumen.

Betrieb der Rechananlage im Sektor Mathematik des HMI

Betrieb

Ein Technik-Team, bestehend aus 3 Ingenieuren, sorgte für einen reibungslosen Betrieb der Rechananlage.

Hahn-Meitner-Institut
für Kernforschung Berlin



Bericht des Sektors Mathematik
im dreizehnten Jahresbericht 1969

Auszug aus
dem
Jahresbericht
1969

5.1 Wartung der S 2002

Die Wartung war wie in den vergangenen Jahren aufgrund der guten Zusammenarbeit zwischen unseren Technikern und dem Wartungspersonal der Firma Siemens sehr zufriedenstellend. Die Reparatur an den veralteten Magnetbandgeräten (IBM 727, Alter zum Teil 12 Jahre !) stößt allerdings auf zunehmende Schwierigkeiten.

Die S2002 lief **tagsüber** an Werktagen und unregelmäßig auch an Wochenenden.

Nutzung

Die Rechenanlage wurde vom Sektor Mathematik für **eigene** Forschung, Software-Entwicklung und Ausbildung genutzt, aber auch in hohem Maße von **anderen** Bereichen des HMI und **Externen**.

5.2 Gerätebelegung S 2002

Die Anlage war im Berichtsjahr 4.874 Std. in Betrieb. 4.422,6 Std. (90,7 o/o) waren echte Rechenzeiten. Für Wartung und Reparatur wurden 169,8 Std. (3,5 o/o) aufgewendet. Die restliche, nicht belegte Zeit (Leerzeit) betrug 281,6 Std. (5,8 o/o).

Die Rechenzeit der Anlage setzt sich wie folgt zusammen:

	<u>Std.</u>	<u>o/o</u>
Technische Universität	1.076,1	24,3
Freie Universität	692,1	15,7
Organisation und Programmbibliothek	101,8	2,3
Eigene Arbeiten des Sektors Mathematik (u. a. Forschungen über Compiler, Vorarbeiten für Bibliotheksprogramme und Dissertationen)	425,7	9,6
Andere Sektoren des Hahn-Meitner-Instituts	1.816,4	41,1
Deutsches Institut für Wirtschaftsforschung	310,5	7,0
	<hr/> 4.422,6	<hr/> 100,0

Auszug aus dem Jahresbericht 1969

Diplom-Mathematiker (ca. 14) und -innen (1) des Bereichs, zum Teil promoviert und habilitiert, nutzten die Anlage für eigene Forschung, entwickelten aber auch wichtige Software für den Betrieb der S2002.

Gender

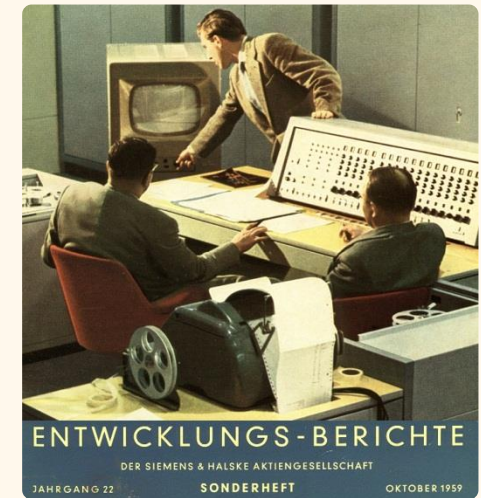
Unterstützt wurden Sie von **mathematisch-technischen Assistentinnen** (ca. 5, dazu ca. 7 im ersten und ca. 7 im zweiten Ausbildungsjahr). Im Jahr 1969 begann erstmalig ein junger Mann diese Ausbildung! Die Ausbildung umfasste die praktische Programmier-Ausbildung in den verschiedenen Arbeitsfeldern der unterschiedlichen Arbeitsgruppen, incl. Bedienung der Rechner, sowie Vorlesungen aus dem Bereich der Mathematik an der TU. Der Studiengang Informatik wurde dort erst 1971 eingerichtet.

Für die **Benutzung** der Rechenanlage für alle größeren Projekte musste **Rechenzeit** reserviert werden. Für kurzfristige Kurzbelegung gab es täglich 2 Zeiträume.

Rechenzeit

Die Anlage wurde meist morgens von mathematisch-technischen Assistentinnen in Betrieb genommen. Danach war jede(r), die/der Rechenzeit reserviert hatte, für die Bedienung selbst verantwortlich.

In dieser Hinsicht war die Anlage ein **PC**, ein **persönlicher Computer**!



Bei der hier ausgestellten Rechenanlage gab **keinen Bootvorgang** im heutigen Sinne und **kein Betriebssystem**. (Lt. Wikipedia konnte allerdings als Betriebssystem ORG 2002 eingesetzt werden.)

Lochstreifen

Nach dem Einschalten musste **sämtliche benötigte Software** im **Maschinencode** in den **Hauptspeicher** (Kernspeicher) **geladen** werden.

Wichtigster Datenträger dafür, zumindest für den Beginn-Teil, war der **gelbe 5-Kanal-Lochstreifen**.



©Dahrendorf, Rodig

Die Lochstreifen wurden hauptsächlich am **Fernschreiber** erstellt, konnten aber auch als Ausgabe von Assemblier- oder Compiler-Läufen an der S2002 erzeugt werden.

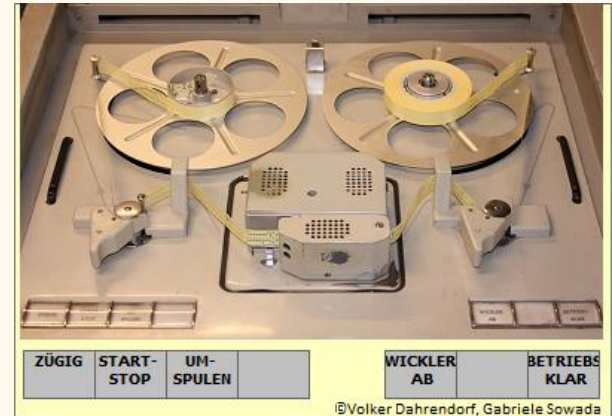


©Dahrendorf, Rodig

Lochstreifen

Über das Eingabegerät **Lochstreifen-Leser** wurde der Maschinencode im **Zügig-Modus direkt in den Kernspeicher** geladen. Insofern war der Zügig-Modus eine Analogie zum Bootvorgang.

Ein so eingelesenes Programm konnte weitere Programmteile vom **Magnetband** in den Kernspeicher nachladen.



©Volker Dahrendorf, Gabriele Sowada


```
00050+005550000250.  
00051+000550000260.  
00052+004440000270.
```

```
TEP I  
ADD J  
TAS K
```

```
k:=i+j;
```

Assembler und **Compiler** halfen schon damals, einfacher lesbaren Programmcode als den **Maschinencode** zu benutzen.

Sprachen

Hauptsächlich wurde in **Assemblersprache** programmiert, damals **mnemotechnischer Operationscode** genannt, an dieser Anlage in **POESIA**, einer eigenen Weiterentwicklung der für diese Rechanlage üblichen Assembler-Sprache **PROSA** (**P**rogrammieren mit **s**ymbolischen **A**dressen).

Es gab auch einen **Compiler** für eine höhere Programmiersprache: **Algol 60**, genutzt für Haupt-Anwendungen im wissenschaftlichen Bereich.

Am **Bedientisch** wurde **jedes Programm gestartet**.

Hier konnte auch der Programm-Ablauf geprüft und überwacht werden.

Im Notfall war auch die manuelle Eingabe kleiner Programme möglich.

Start



Der Bedientisch bot damals das zum Anfassen, was heute jeder Software-Entwickler virtuell als **IDE** (integrated development environment) kennt.

Hauptsächlich genutztes **Speichermedium** für Daten bei Ein- und Ausgabe war der **Lochstreifen**.



Magnetband

Als wichtigstes **transportables** und **überschreibbares Speichermedium** für große Datenmengen wurden **Magnetbänder** genutzt.

Eine besondere Herausforderung an die Technik stellten statistische Auswertungen mit Sortierläufen dar, die auf alle 4 Magnetbandgeräte gleichzeitig zugriffen.

Direkte Ausdrücke auf Papier als Ausgabe waren eher unüblich. Rechenzeit war kostbar, Lochstreifenausgabe war schneller als Druck. Die Lochstreifen konnten später am Fernschreiber ausgedruckt werden.

Papier

Die Tabelliermaschine wurde häufig zum Ausdrucken von **Dumps** genutzt – Kernspeicherauszüge, die für die Fehlersuche dringend gebraucht wurden.

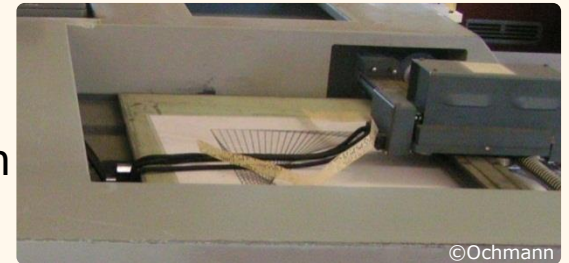
Auf Papier konnte man den Programmablauf am Schreibtisch nachvollziehen, Speicherinhalte überprüfen und Fehler finden, ohne kostbare Rechenzeit mit der Fehlersuche am Bedienpult zu verschwenden.



Für die punktweise grafische Anzeige von berechneten oder Messdaten gab es ein monochromes **Analog-Sichtgerät**.

Grafik

Üblicherweise wurden grafische Auswertungen jedoch auf Papier dargestellt. Ausgiebig dafür genutztes Zeichengerät war der **Zuse Graphomat Z64**.



©Ochmann

SIEMENS 2002

Programmierung

Viele Menschen sind heutzutage schon mehr oder weniger mit der **Programmierung** eines **PCs** vertraut.



Datenverarbeitungsanlage 2002

Programm

Bei der **Programmierung** einer **damaligen Rechenanlage** war anderes zu bedenken als bei der Programmierung eines PCs.

Vorgehensweise und **Beschränkungen** kommen eher der Programmierung von **Mikro-Controllern** nahe. Es galt immer, die **Ablaufgeschwindigkeit** eines Programms zu **optimieren**, dabei aber **keinesfalls Speicherplatz** zu **verschwenden**.

An der Siemens 2002 wurden nur **imperative Programmiersprachen** genutzt.

Sprachen

```
00050+005550000250.  
00051+000550000260.  
00052+004440000270.
```

```
TEP I  
ADD J  
TAS K
```

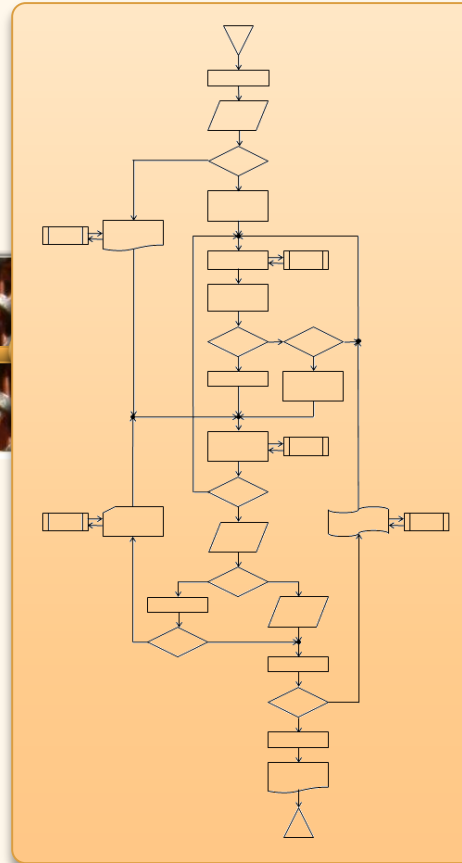
Besonders die Techniker und Wartungsingenieure bevorzugten die Erstellung von Programmen direkt im **Maschinencode**.

Üblich war die **Programmierung** in der Assembler-Sprache **PROSA** (**Pro**grammieren mit **s**ymbolischen **A**dressen), im Sektor Mathematik des HMI auch in der eigenen Weiterentwicklung **POESIA**.

Geringe Speicherkapazität und **langsame Rechenleistung** erforderten einen Programmierstil, der heutzutage verächtlich als **Spaghetti-Code** bezeichnet wird.

Sprachen

Deutlich wird dieser Stil besonders bei großen Programmen. Charakteristisch sind **viele Verzweigungen** mit **Sprüngen** vor- und rückwärts, wobei sich die **Linien** häufig **kreuzen**.



Am HMI war es darüber hinaus möglich, auf der S2002 in einer **höheren Programmiersprache** zu arbeiten.

Sprachen

```
00050+005550000250.  
00051+000550000260.  
00052+004440000270.
```

```
TEP I  
ADD J  
TAS K
```

```
k:=i+j;
```

Ein **ALGOL60-Compiler** ermöglichte erste Schritte im Stil der **strukturierten** Programmierung. Gerade für mathematische Problemstellungen wurde der Programmcode besser lesbar und damit weniger fehleranfällig.

Allerdings musste man dafür die eventuelle **Verschwendung von Speicherplatz** und eine möglicherweise **längere Laufzeit** des Programms in Kauf nehmen.

Als **Universalrechner** konnte die Siemens 2002, je nach verwendetem **Programm**, in kaufmännischen und verwaltenden, wie auch in technisch-wissenschaftlichen Bereichen eingesetzt werden.

Aufgaben

Einfache **kaufmännische Berechnungen** stellten für die Programmierung keine großen Probleme dar. Kaufmännische Berechnungen werden nach endlichen Schemas durchgeführt. Bei der Berechnung wird eine Genauigkeit vorgegeben, die auch auf der S2002 problemlos umzusetzen war.

Organisationsprogramme zur **Datenverwaltung** waren eine größere Herausforderung. Damals wurden theoretische Grundlagen und erste Versionen von Datenbank-Software erarbeitet.

Für viele Problemstellungen in Wissenschaft, Forschung und Technik sind jedoch **algebraische Ausdrücke** auszuwerten.

Numerik

$$I(\alpha) = I_0 * \cos^2(\alpha - \alpha_0) + IF$$

$$J_n(x) = \frac{1}{\pi} * \int_0^{\pi} \cos(x * \sin(t) - n * t) * dt$$

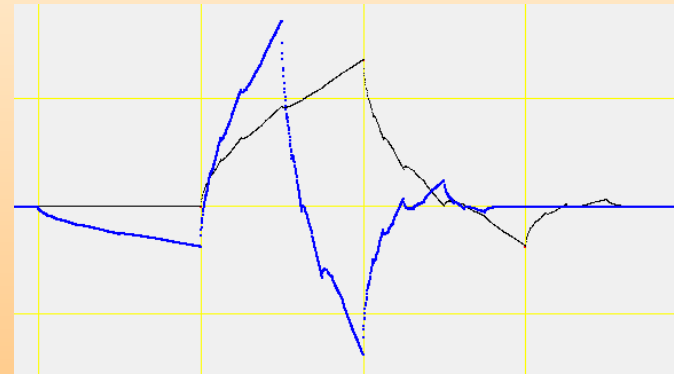
Derartige Ausdrücke können auf einer Rechenanlage prinzipiell nur in einer **numerischen Näherung** berechnet werden.

Numerische Mathematik und **Approximations-Theorie** wurden wichtige Forschungsgebiete, die **auch heute** nicht nur Altbewährtes zur Verfügung stellen, sondern immer wieder wichtige neue Verfahren liefern.

Keine Ingenieursdisziplin kam und kommt ohne **Fourier-Transformationen** aus, kein Schiffbau-Ingenieur ohne seine Sammlung **Numerical Recipes**.

Numerik

Ohne die besonders in den 1980er Jahren beschriebenen verschiedenartigen **Wavelets**, hätten wir weder JPG, noch MP3, noch die am Computer generierten Zeichentrickfilme z. B. der Firma Pixar.



Daubechies-Wavelets
(Ingrid Daubechies, *1954)

Zu **algebraischen Ausdrücken** müssen für die Berechnung mit dem Computer also immer **numerische Näherungen** gefunden werden. Unendliche Reihen müssen bei Erreichen einer vorgegebenen **Genauigkeit** abgebrochen werden.

Näherung

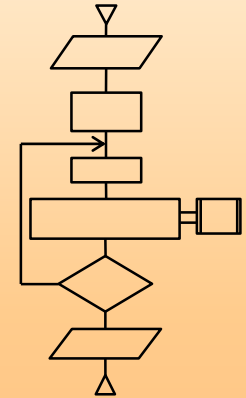
Die Rechenanlage war zwar damals, besonders für komplexe Berechnungen, **schneller als Menschen**, war aber für heutige Verhältnisse unvorstellbar **langsam**. Operationszeiten, die heute im Nano-Sekundenbereich liegen, bewegten sich damals im mehrstelligen Mikro-Sekundenbereich. Gesucht waren damals deshalb besonders **Näherungsverfahren, die schnell konvergierten**. Alternierende Reihen waren dabei wegen der einfachen Restgliedabschätzung für die Genauigkeit praktisch.

$$y = \sin x$$

$$\sin x = \sum_{n=0}^{\infty} (-1)^n \frac{x^{2n+1}}{(2n+1)!}$$

$$\sin x = \sum_{n=0}^{\infty} (-1)^n \frac{x^{2n+1}}{(2n+1)!}$$

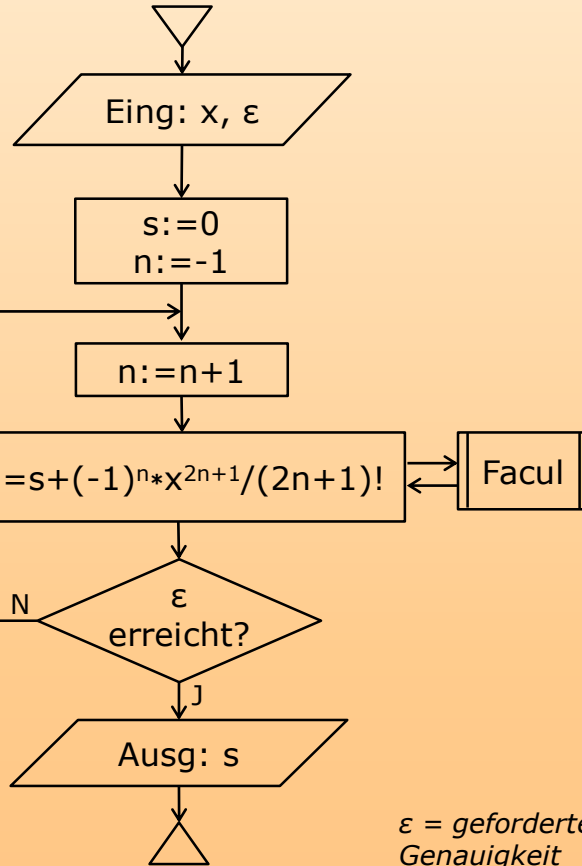
Vor der eigentlichen Programmierung müssen solche numerischen Näherungen aber erst in **Algorithmen** überführt werden.



Algorithmus

Algorithmen sind **Vorgehensweisen**, die später in einer Programmiersprache formuliert, also als Programmbefehle geschrieben werden können. Zur Darstellung von Algorithmen wurden damals **Strukturdiagramme** verwendet.

Wie überall im Leben, so gibt es auch am Computer immer mindestens zwei Möglichkeiten, ein Ziel zu erreichen. Damals war es besonders wichtig, die gefundenen Näherungsverfahren in **schnelle Algorithmen** zu überführen, die zudem **möglichst wenig Speicherplatz** einnehmen durften.



ϵ = geforderte Genauigkeit

$$\sin x = \sum_{n=0}^{\infty} (-1)^n \frac{x^{2n+1}}{(2n+1)!}$$

Varianten

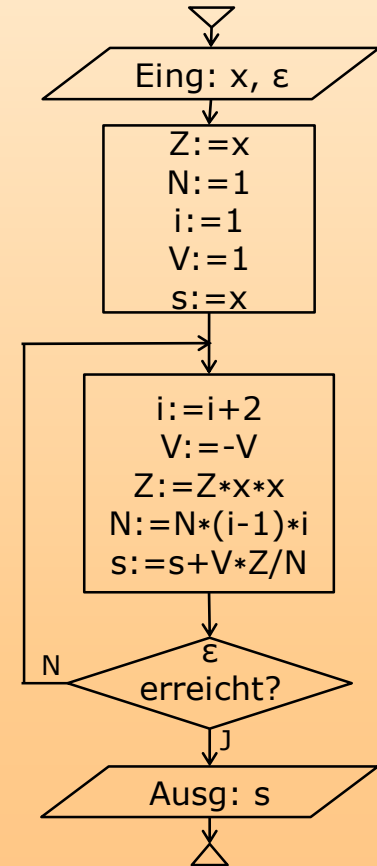
Eng an der **mathematischen Form orientierte Algorithmen** liefern zwar eine übersichtliche, **leicht nachvollziehbare** Vorgehensweise, erfordern aber i. A. **hohe Rechenleistung** (im Beispiel wiederholte Potenzierungen und wiederholte Berechnung der Fakultät), waren für die damaligen Rechenanlagen also meist ungeeignet.

$$\sin x = \sum_{n=0}^{\infty} (-1)^n \frac{x^{2n+1}}{(2n+1)!}$$

Varianten

Auf einen **schnellen Ablauf gezielte Algorithmen** erfordern **mathematische Umformung** – jeder Schritt im höchsten Grade **fehleranfällig**, aber unter damaligen Umständen meist **unabdingbar**.

ε = geforderte Genauigkeit



Ist ein **Algorithmus** formuliert, muss die **Implementation** überdacht werden, die **Umsetzung** in Programmcode. Wieder gibt es verschiedene Möglichkeiten und verschiedene Programmiersprachen.

Umsetzung

Damals genauso wichtig wie ein **schnelles Näherungsverfahren** und ein **schneller Algorithmus** war der **schnelle Programmcode**. Eine höhere Programmiersprache erhöht zwar die Lesbarkeit des Programmcodes, er kommt der Schreibweise des Algorithmus sehr nahe. In einem Programm in **Assemblersprache** kann der/die Programmierer/in jedoch selbst auf die verwendeten Maschinenbefehle Einfluss nehmen – deshalb damals fast **immer** die **Sprache der Wahl**!

Selbst nach der Festlegung auf die Assembler-Sprache waren noch nicht alle **Optimierungsmöglichkeiten** der **Ablaufgeschwindigkeit** ausgereizt.

Operationszeiten

Einheit Wortzeit, **1 Wortzeit = 90 µs**

	Festkomma	Gleitkomma
Addition	1	5
Subtraktion	1	5
Multiplikation	14	16*
Division	39*	36*

*plus weitere Wortzeiten für die Normalisierung

Umsetzung

Um die Ablaufgeschwindigkeit eines Programms zu optimieren, musste auf die **Ausführungszeit** jedes einzelnen Befehls geachtet werden und die **schnellste Variante** zur Ausführung einer Operation gewählt werden. Es musste außerdem genau beachtet werden, wo schnelle **Festkomma-Arithmetik** benutzt werden konnte, wo aber für hohe Rechengenauigkeit mit Nachkommastellen langsame **Gleitkomma-Arithmetik** eingesetzt werden musste.

Besonders bei **großen Projekten** durfte nie die sehr **begrenzte Speicherkapazität** außer Acht gelassen werden!

Umsetzung

Die **Beschränkung** des **Hauptspeichers** auf ursprünglich 3.000, in der zweiten Ausbaustufe **5.000 Worte**, machte es nötig, mit jedem Wort zu geizen. So war es üblich, die **Speicherstellen**, an denen nicht mehr benötigte **Befehle** standen, später als **Datenspeicher** zu nutzen – eine nie versiegende Quelle schwer aufzuspürender **Laufzeitfehler**!

Die zur **Problemlösung** erforderlichen **mathematischen Formeln** und die passenden **Näherungsverfahren** wurden von den **Wissenschaftlern** erarbeitet. Die Wissenschaftler waren am HMI fast ausschließlich Männer.

Menschen

Die **Programmentwicklung**, also das Erarbeiten der geeigneten **Algorithmen**, die **Umsetzung** in Programmcode und das **Austesten** der Programme, wurde am HMI von den **mathematisch-technischen Assistentinnen** durchgeführt. Die mathematisch-technischen Assistentinnen, im Englischen würde man den Begriff **Software-Engineer** benutzen, waren am HMI fast ausschließlich Frauen.

Beispiele aus dem Jahresbericht 1969 des Sektors Mathematik am HMI, die Arbeit an der Siemens 2002 beinhalteten,
eigene Projekte des Sektors Mathematik:

Projekte

Programmsystem Lineare Optimierung

Statistik-System COVAR

Zeichenprogrammsystem für Graphomat

Programm zur Verwaltung des Siemens-Schrifttums

Beispiele aus dem Jahresbericht 1969 des Sektors Mathematik am HMI, die Arbeit an der Siemens 2002 beinhalteten, **Projekte anderer Bereiche** unter **Beteiligung** des Sektors Mathematik.

Projekte

Grafische Darstellung von γ -Spektren (HMI, Sektor Kernchemie)

Monte-Carlo-Rechnung für ein Ge(Li)- γ -Spektrometer mit NaJ-Anti-Compton-Einrichtung (HMI, Sektor Kernchemie)

Iterative Ermittlung der Geschwindigkeitskonstanten der Reaktion $\text{NO}_2 + \text{OH} \rightarrow \text{HO NO}_2$ (HMI, Sektor Strahlenchemie)

Beispiele aus dem Jahresbericht 1969 des Sektors Mathematik am HMI, die Arbeit an der Siemens 2002 beinhalteten, **Projekte der Freien Universität** unter **Beteiligung** des Sektors Mathematik.

Projekte

Auswertprogramm der Radiosondenaufstiege (FU, Institut für Meteorologie und Geophysik)

Fourier-Transformationen von Autocovarianzfunktionen zur Analyse von Elektroencephalogrammen (FU, Psychiatrische und Neurologische Klinik)

Auswertung von etwa 10000 Strahlenschutzuntersuchungen von etwa 3000 Personen (FU, Institut für Arbeitsmedizin)

Beispiele aus dem Jahresbericht 1969 des Sektors Mathematik am HMI, die Arbeit an der Siemens 2002 beinhalteten, **Projekte der Technischen Universität** unter **Beteiligung** des Sektors Mathematik.

Projekte

Statistische Auswertung der Ausleihvorgänge (TU, Universitätsbibliothek)

Berechnung von Varianz- und Kovarianzanalysen sowie genotypische und phänotypische Korrelationen für Leistungsdaten beim Schwein (TU, Institut für Tierzüchtung und Haustiergenetik)

Zeichnung von ISO-Linien (TU, Institut für Mechanik)

Der Einfluss verschiedener Aufzuchtformen auf die Eiqalität (TU, Institut für Tierzüchtung und Haustiergenetik)

Beispiele aus dem Jahresbericht 1969 des Sektors Mathematik am HMI, die Arbeit an der Siemens 2002 beinhalteten, **Projekte des Deutschen Instituts für Wirtschaftsforschung** (DIW) mit und ohne **Beteiligung** des Sektors Mathematik:

Projekte

Anlagevermögensrechnung

Monetäres Modell der Bundesrepublik Deutschland für kurzfristige Prognosen

Aufstellung projektionsfähiger Funktionen für Einkommensschichtungen

Versuchsrechnungen zum Projekt: Nachwuchskräftebedarf in Berlin-West
(Lineare Programmierung und Simplex-Methode)

Beispiele aus dem Jahresbericht 1969 des Sektors Mathematik am HMI, die Arbeit an der Siemens 2002 beinhalteten, **Projekte ohne Beteiligung** des Sektors Mathematik.

Projekte

Kristallstrukturberechnungen (FU, Lehrstuhl für Kristallographie)
Mößbauer-Effekt (HMI, Sektor Kernphysik)
Auswirkung von Fluglärm auf die Bevölkerung (FU, Psychologisches Institut)
Stabilität gleitgelagerter Rotoren (TU, Lehrstuhl für Konstruktionslehre)
Integration (FU, Institut für theoretische Physik)
Auswertung von Flugzeitmessungen (HMI, Abteilung Reaktorphysik)

SIEMENS 2002

Programmbeispiele

Emulation der **Siemens 2002** auf dem PC unter Windows von **Volker Dahrendorf**



Datenverarbeitungsanlage 2002

Emulation

Für alle Beispiele wurde die **Emulation** der **Siemens 2002** auf dem PC unter Windows von **Volker Dahrendorf** benutzt.

Volker Dahrendorf gehörte zu den ersten Schülern an der Dreilindenschule in Berlin, die in der Informatik-AG die Siemens 2002 für den Unterricht modifizierten und nutzten.

Emulation der **Siemens 2002** auf dem PC
unter Windows von **Volker Dahrendorf**



Datenverarbeitungsanlage 2002

3. Etage

Dieses **Emulationsprogramm** kann in der 3. Etage des Computer-Museums
von jedem/r Besucher/in benutzt werden.

Es gibt **6 automatisch ablaufende Demonstrationen** und die Möglichkeit,
die Siemens 2002 **selbst virtuell** zu bedienen.

Allerdings **muss** im **Maschinencode** gearbeitet werden.
Es stehen **weder Assembler noch Compiler** zur Verfügung.

Emulation der **Siemens 2002** auf dem PC unter Windows von **Volker Dahrendorf**



Datenverarbeitungsanlage 2002

Bedienung

Dieses Emulationsprogramm wird genauso **bedient**, wie damals die S2002 – nur eben **virtuell**. Statt echte Tasten am Bedientisch zu drücken und Drehschalter zu drehen, wird in der Emulation mit den **Maustasten** auf Tasten und Schalter geklickt.

Statt echte Lochstreifen, Papier, usw. zu erzeugen, wird mit **Dateien** gearbeitet.

Emulation der **Siemens 2002** auf dem PC unter Windows von **Volker Dahrendorf**



Datenverarbeitungsanlage 2002

Beispiele

Die Beispiele sollen einen kurzen Einblick in den **Umgang** mit der S2002 und die **grundlegenden Vorgehensweisen** geben.

3 der Beispiele werden auch mit längeren Erläuterungen als **PowerPoint-Präsentationen** vorgestellt. Die anderen 3 stehen nur als automatisch ablaufende Demonstrationen am **Emulations-PC** zur Verfügung.

Programm-Beispiel **1**:
Manuelle Programm-Eingabe,
Festkomma-Addition



Manuell

Beispiel 1 demonstriert die Vorgehensweise bei der **manuellen** Programm-Eingabe am **Drehschalterfeld**.

Diese direkteste Art der Eingabe wurde beim Testen von Programmen benutzt, um z. B. Befehlscode und/oder Daten zu ergänzen oder zu modifizieren.

Hier soll ein Mini-Demo-Programm eingegeben werden. Es soll **zwei Werte addieren**. Benutzt wird dabei **Festkomma-Arithmetik**.

Programm-Beispiel 2:
Programm-Eingabe im **ZÜGIG-Modus**,
Gleitkomma-Addition



Zügig

Beispiel 2 demonstriert die Vorgehensweise bei der normalerweise genutzten Art der Programm-Eingabe per **Lochstreifen**. Hier wird ein Lochstreifen erstellt, der den Programmcode in Maschinensprache in **Zügig-Codierung** enthält. Der Lochstreifen wird dann im **Zügig-Modus** eingelesen, der Maschinencode direkt in den Kernspeicher schreibt.

Das Mini-Demo-Programm soll mit **Gleitkomma-Arithmetik** (unnormalisiert) **zwei Werte addieren**.

Programm-Beispiel 3: **Zählschleife**



Datenverarbeitungsanlage 2002

Zählschleife

Beispiel 3 demonstriert die Implementation einer Zählschleife, einer häufig benutzten Steuerungsstruktur (control structure):
Zähler auf **Anfangswert** setzen, in einer Schleife Zähler **inkrementieren** und die Schleife verlassen, wenn der vorgegebene **Endwert** erreicht ist.

In der Schleife stehen normalerweise wichtige Anweisungen, die z. B. bei jedem Durchlauf für ein anderes Element ausgeführt werden sollen. Hier im **Beispiel** soll einfach nur von 1 bis 10 gezählt werden, in Schritten von 1.

Programm-Beispiel 4-6: **Peripherie**



Datenverarbeitungsanlage 2002

Peripherie

Beispiel 4 demonstriert die Ausgabe von Lochkarten.

Beispiel 5 demonstriert den Zugriff auf ein Magnetband.

Beispiel 6 demonstriert die Anzeige auf dem Analoysichtgerät.



SIEMENS 2002

Beispiel 1 Manuell

Für alle Beispiele wurde die **Emulation** der **Siemens 2002** auf dem PC unter Windows von **Volker Dahrendorf** benutzt.



Manuell

Beispiel 1 demonstriert die Vorgehensweise bei der **manuellen** Programm-Eingabe am **Drehschalterfeld**.

Diese direkteste Art der Eingabe wurde beim Testen von Programmen benutzt, um z. B. Befehlscode und/oder Daten zu ergänzen oder zu modifizieren.

Hier soll ein Mini-Demo-Programm eingegeben werden. Es soll **zwei Werte addieren**. Benutzt wird dabei **Festkomma-Arithmetik**.

Beispiel 1: A plus B ergibt C, in Basic: **C=A+B**, in Algol: **C:=A+B;**

Manuell

in **Assembler**, bei Entscheidung für **Festkomma-Arithmetik**
(dabei steht im Folgenden die Abkürzung **AR** für das akkumulative Register):

TEP	A	Lade das AR mit dem Inhalt der Speicherzelle A
ADD	B	Addiere zum AR den Inhalt der Speicherzelle B
TAS	C	Schreibe den Inhalt des AR in die Speicherzelle C
STP		Halte an - fertig

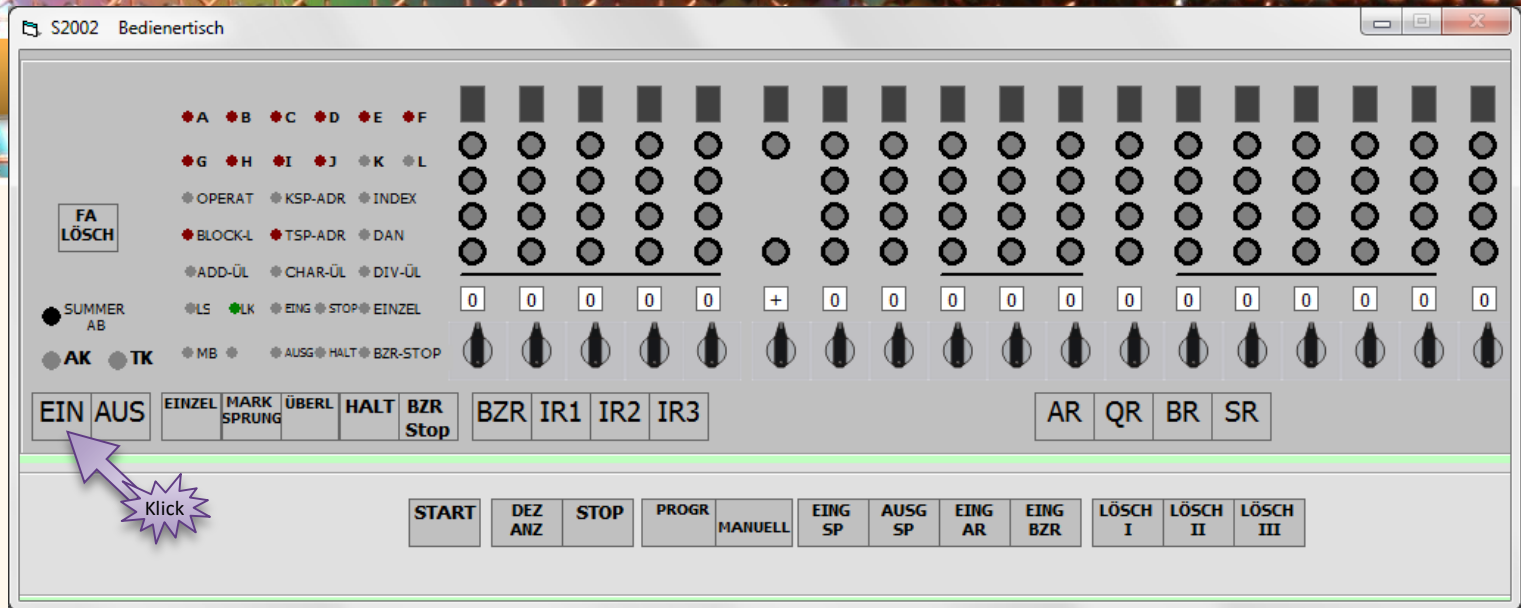
Die Assembler-Befehle müssen in **Maschinencode** übersetzt werden und für die **symbolischen** Namen der Speicherzellen A, B und C müssen **absolute** Speicheradressen festgelegt werden.

Manuell

Adresse	Inhalt	Erläuterung
00001	000000000005	Speicheradresse für A, hier der Wert 5
00002	000000000007	Speicheradresse für B, hier der Wert 7
00003		Speicheradresse für das Ergebnis C
00010	005550000010	Befehl TEP A
00011	000550000020	Befehl ADD B
00012	004440000030	Befehl TAS C
00013	002000000000	Befehl STP

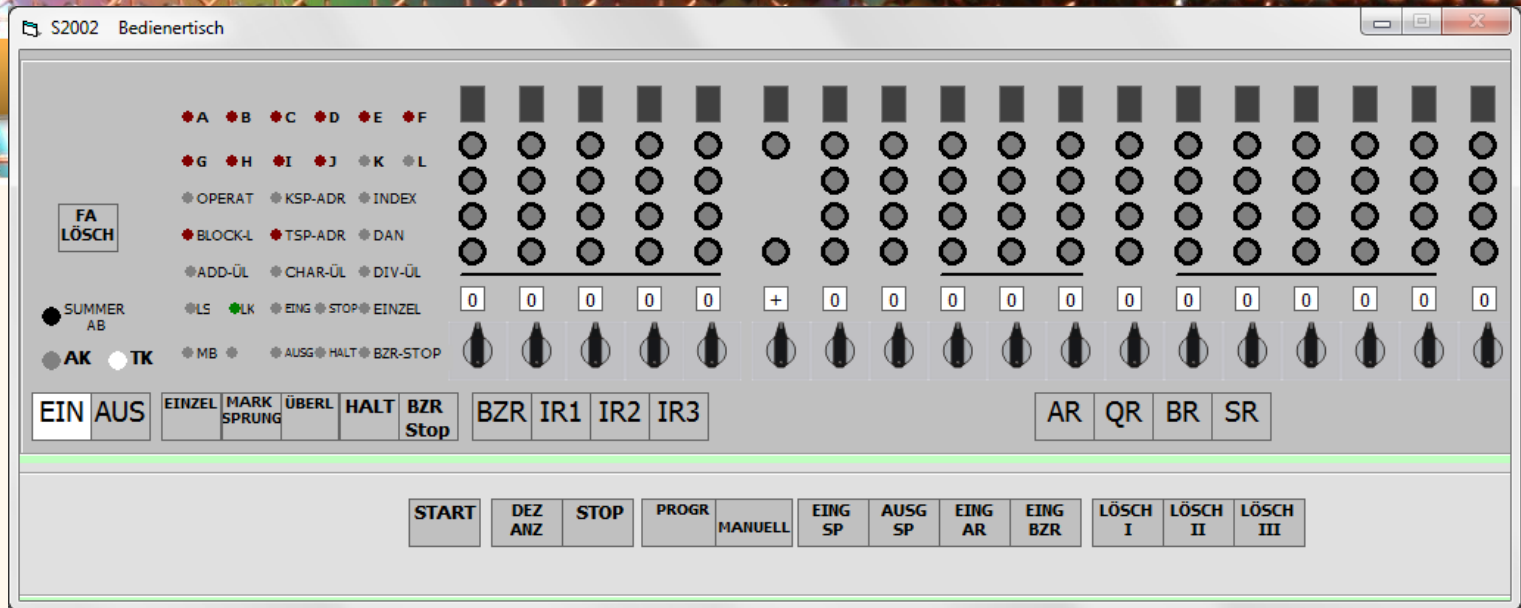
Die Rechenanlage wird am Bedientisch eingeschaltet (**EIN**).

Manuell



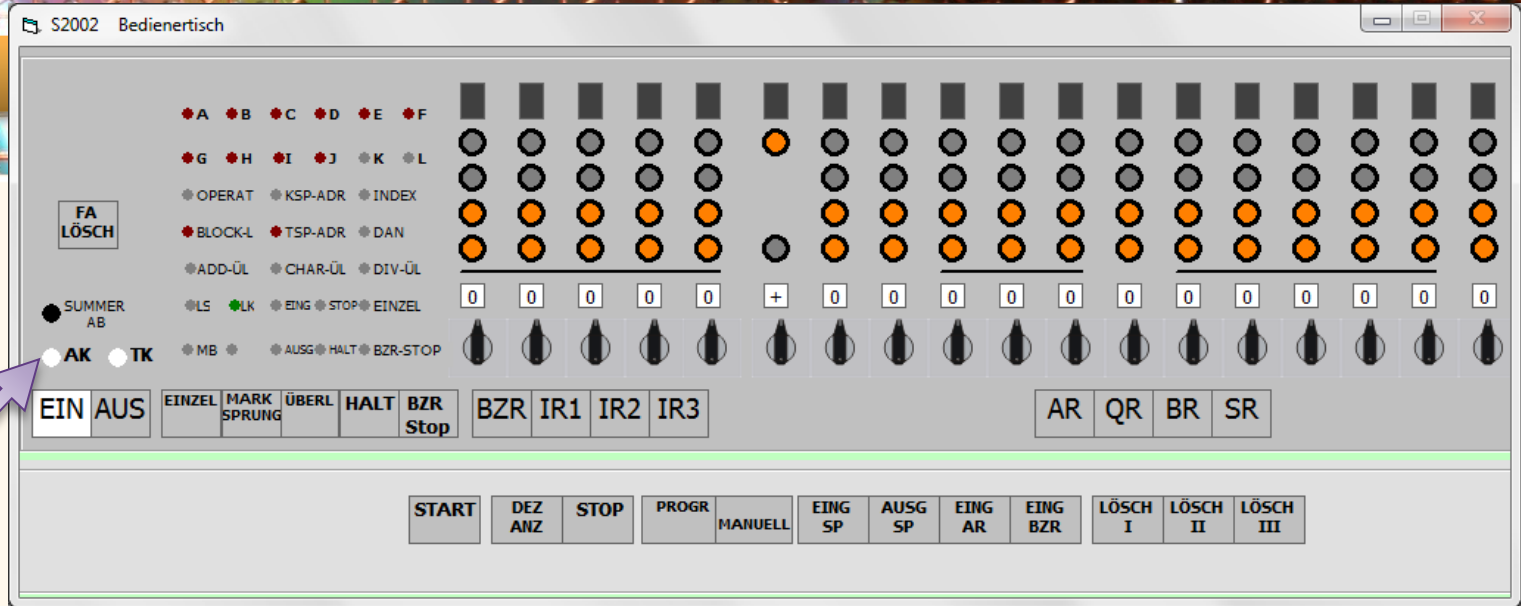
Der Einschaltvorgang dauert einen Moment,
denn die **mehrstufige Spannungseinschaltung** braucht einige Zeit.

Manuell



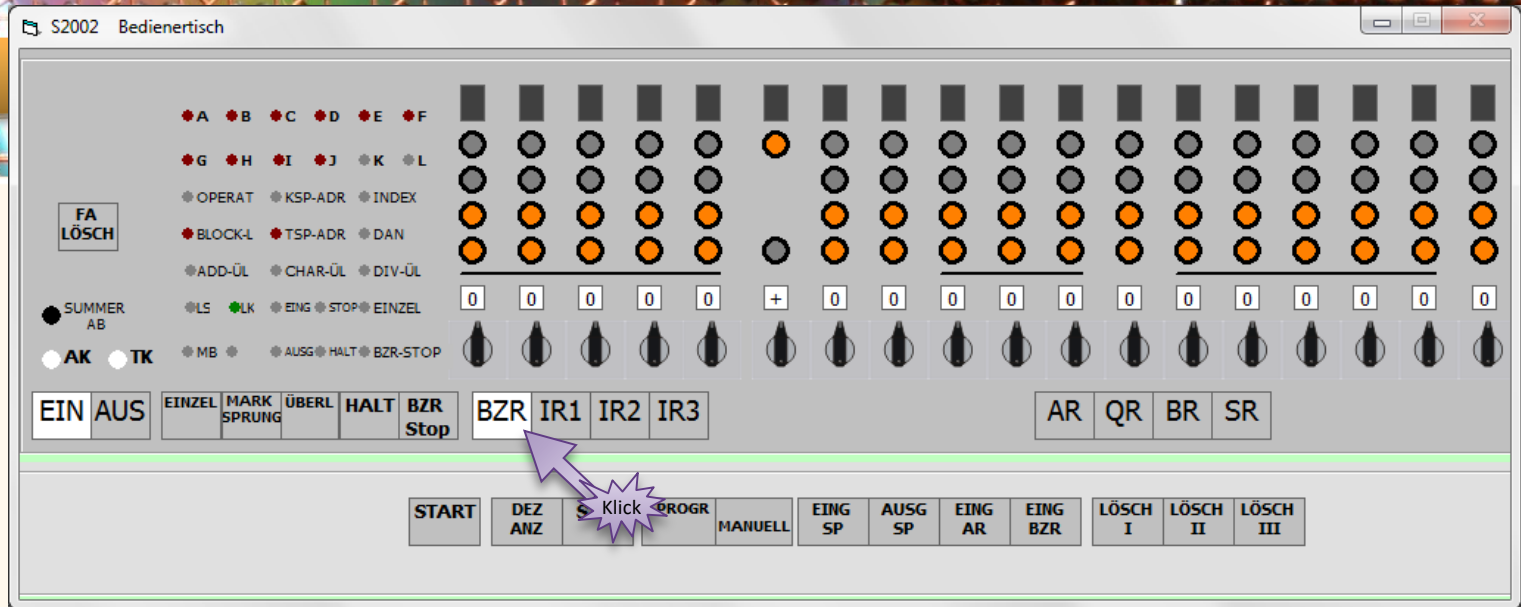
Anlage klar (**AK**) signalisiert, dass die Rechenanlage betriebsbereit ist.

Manuell



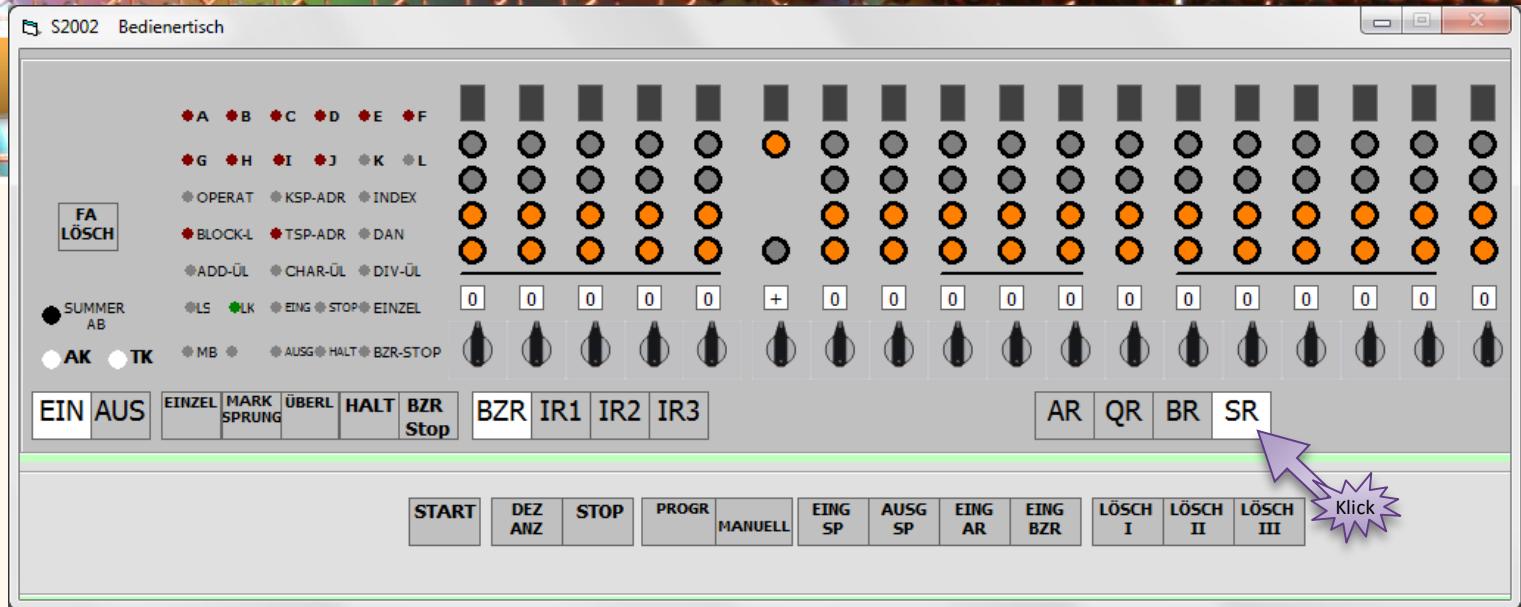
Bei den 5-stelligen Registern
Anzeige des Befehlszählers (**BZR**) einschalten:

Manuell



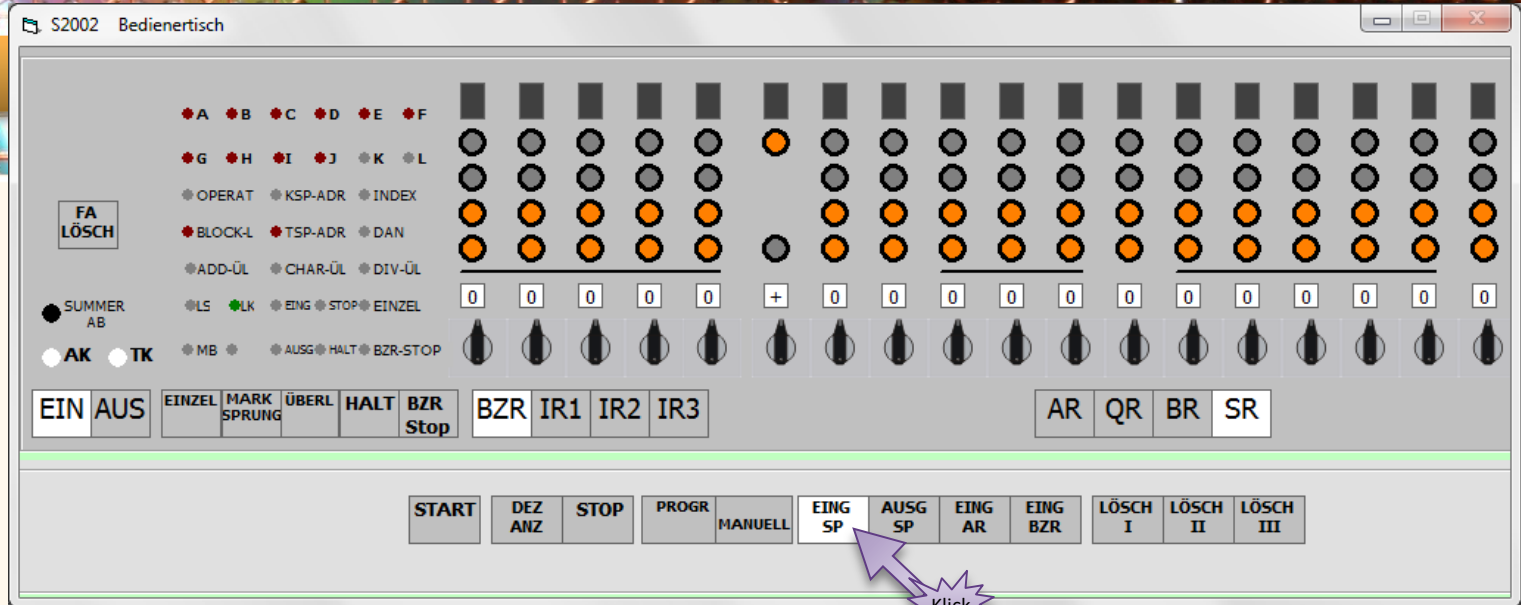
Bei den 12-stelligen Registern
Anzeige des Speicherregisters (**SR**) einschalten:

Manuell



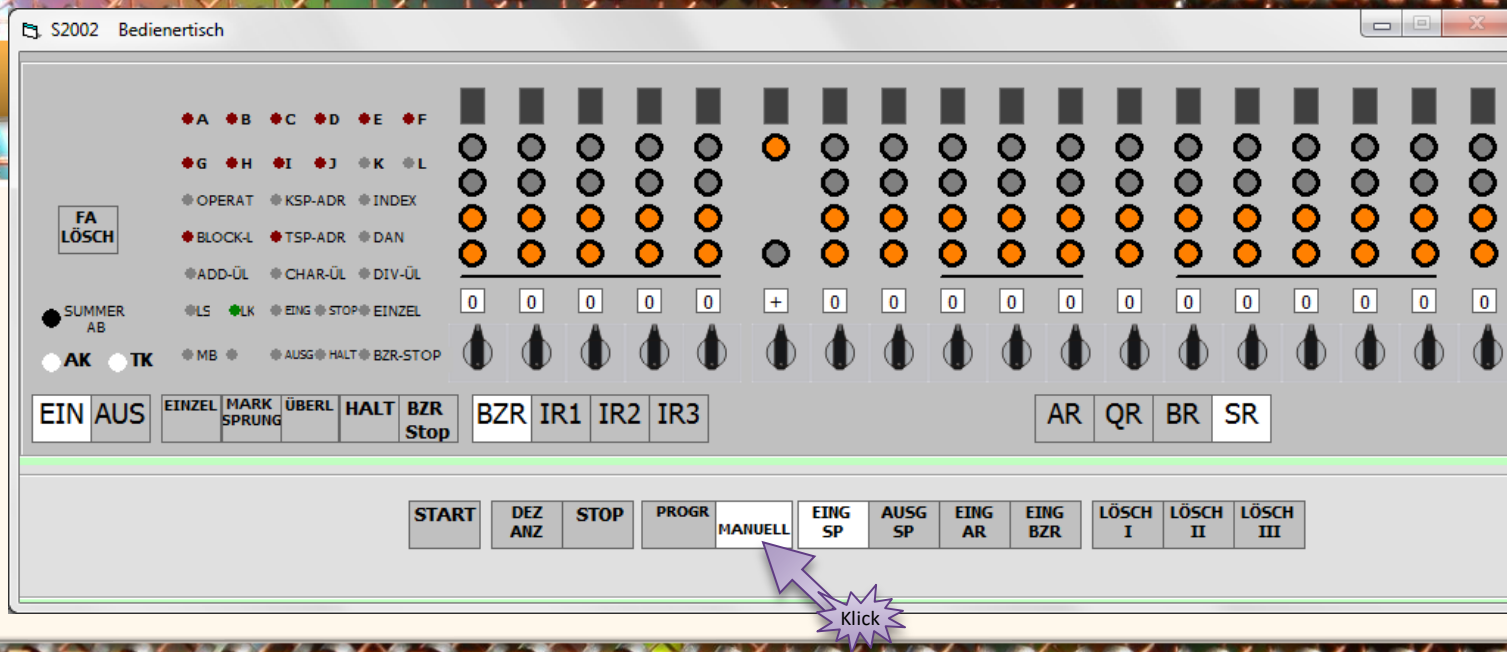
Als Aktion Eingabe in den Speicher (**EING SP**) einschalten:

Manuell



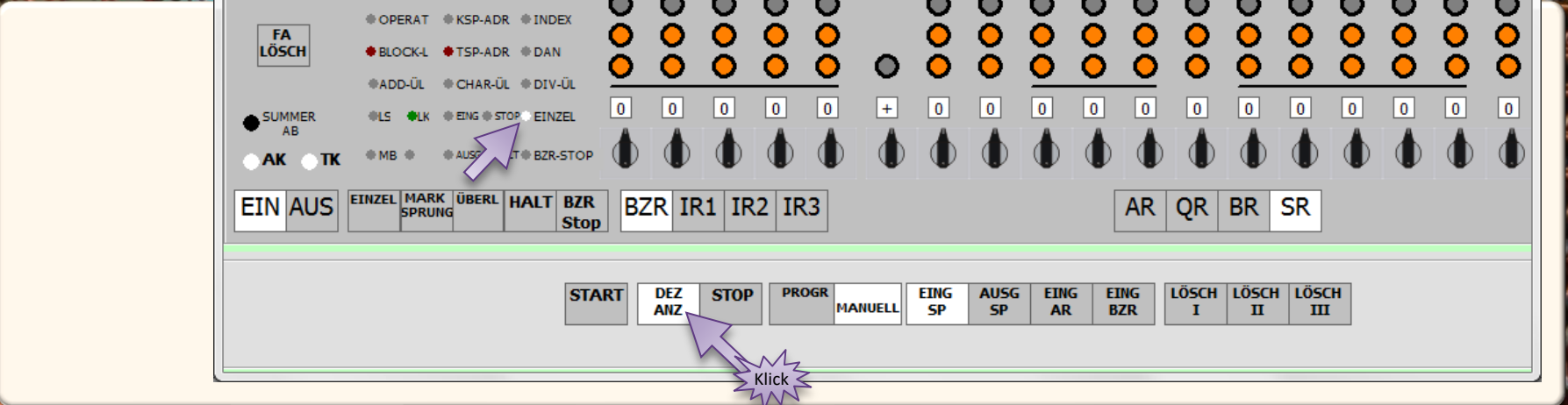
Den manuellen Modus (**MANUELL**) einschalten:

Manuell



Dezimalanzeige (**DEZ ANZ**) einschalten,
das bewirkt immer die gleichzeitige Aktivierung des Einzelschrittmodus'.

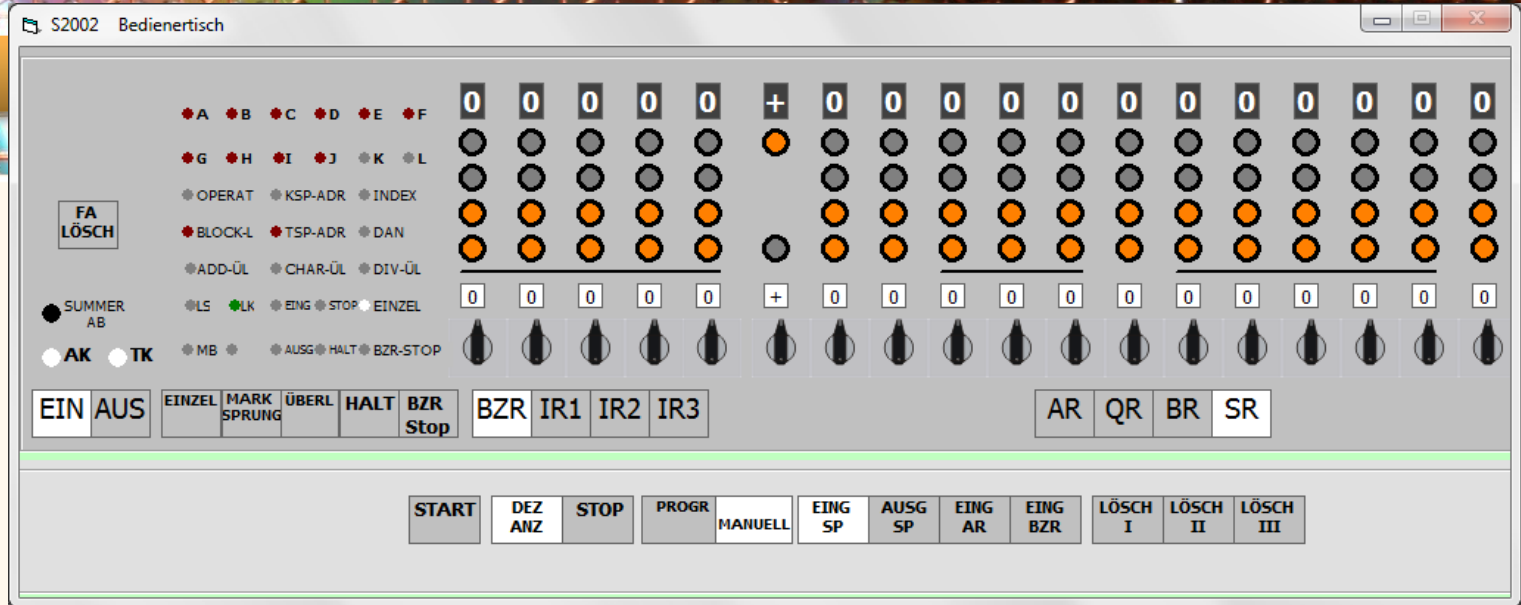
Downloaded from <http://ajph.org/> on November 10, 2014



Nun werden **nacheinander** die Daten und Befehle

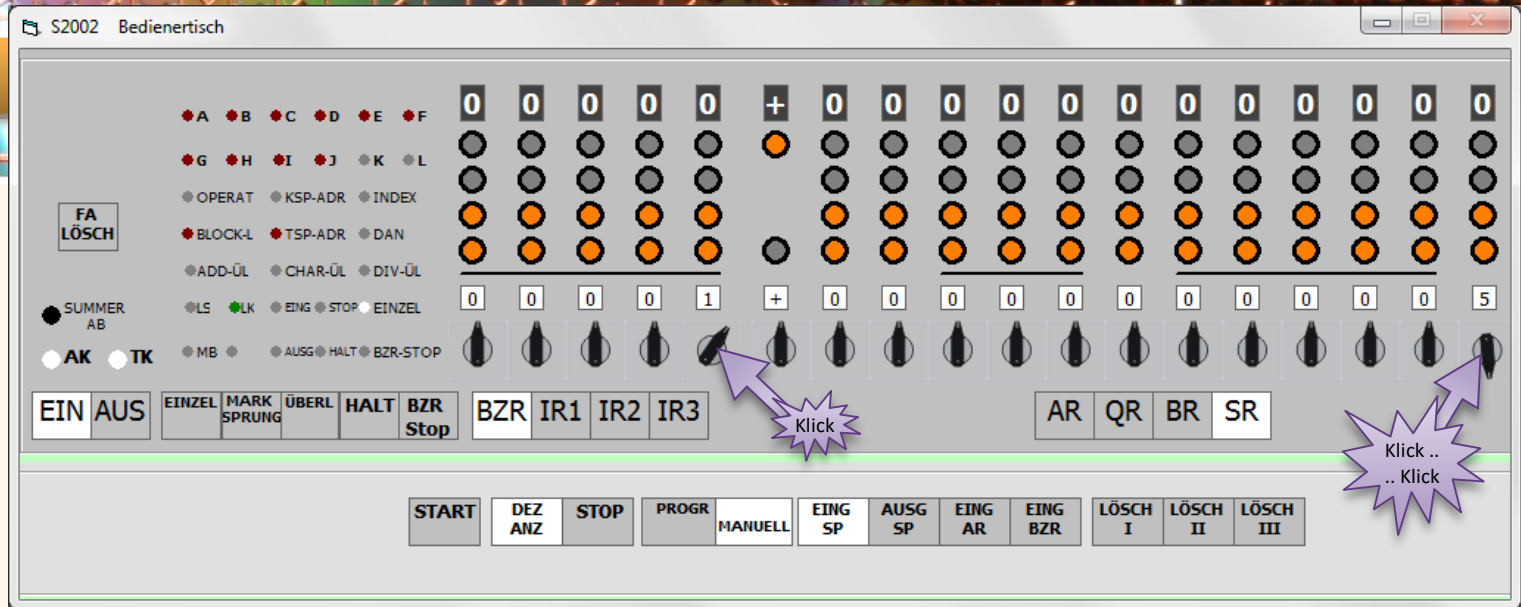
- erst am **Drehschalterfeld eingestellt** und
- dann in die richtigen Speicherzellen **eingegeben** (geschrieben).

Manuell



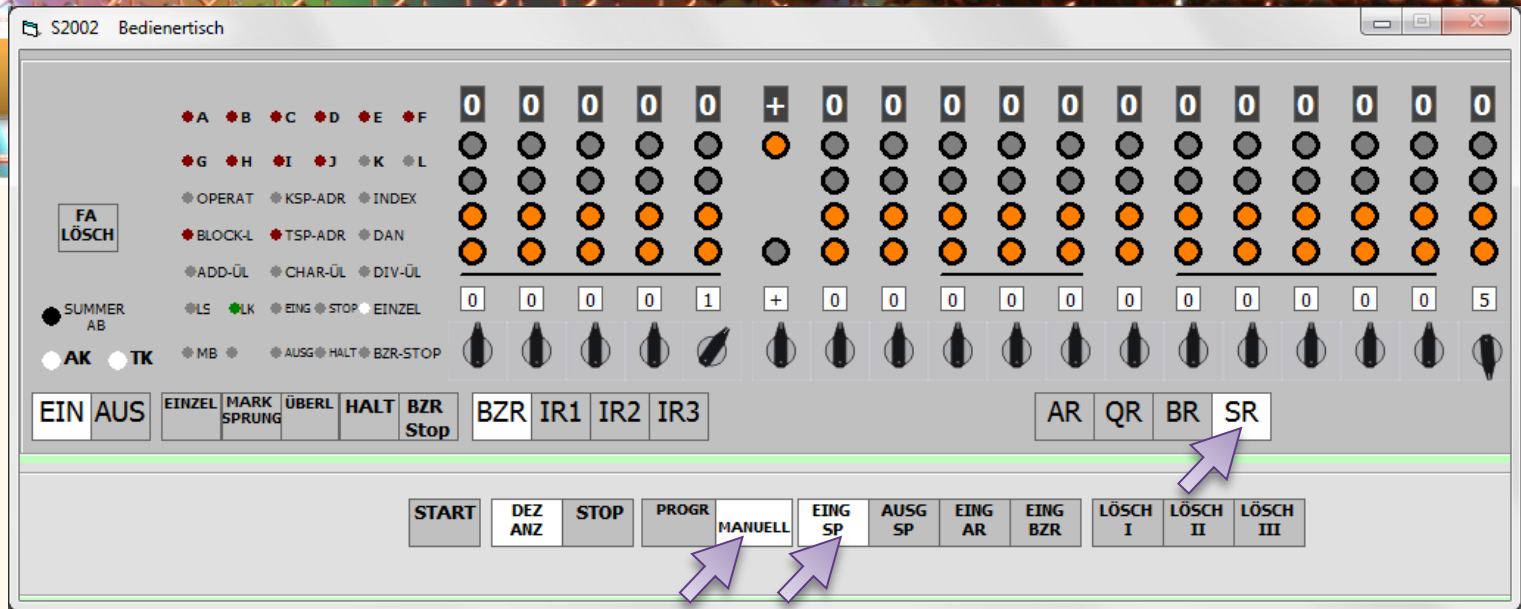
Adresse 1 soll für A den **Wert** 5 aufnehmen, **einstellen**:

Manuell



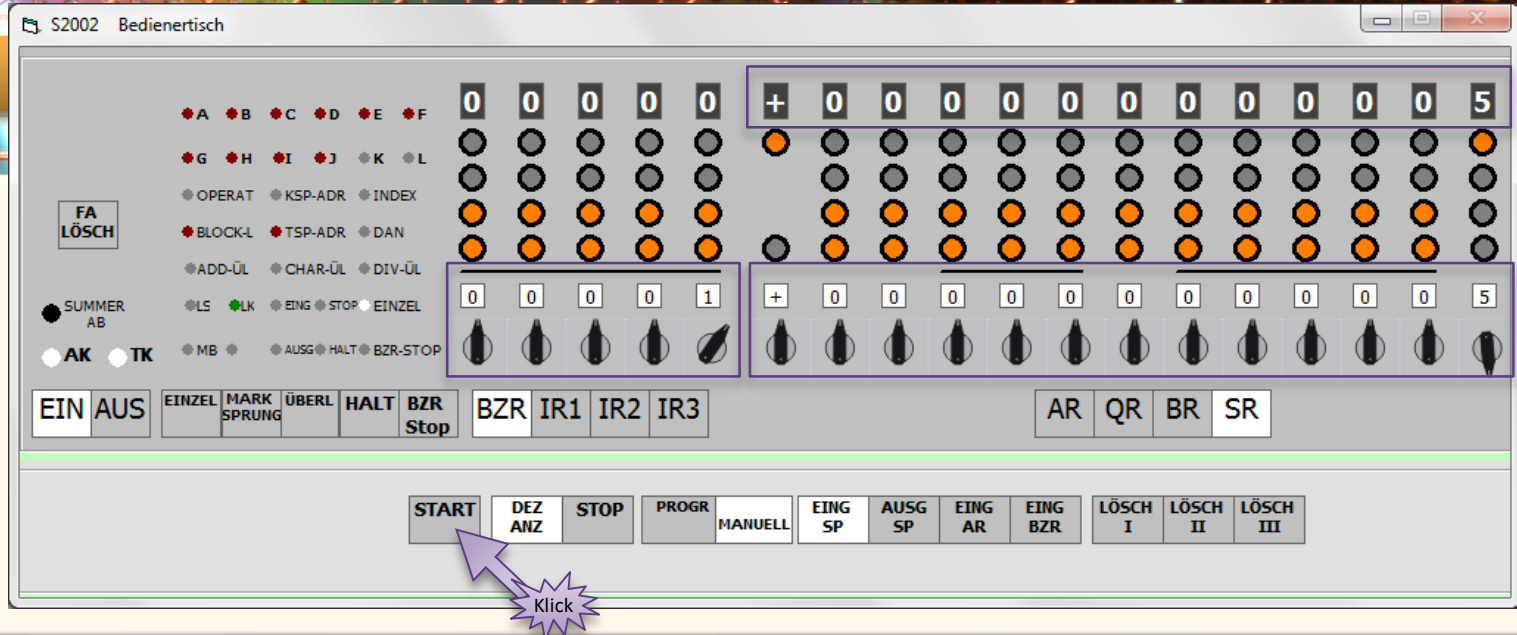
Kontrolle: Eingabe in den Speicher (**EING SP**) und manueller Modus (**MANUELL**) sind eingestellt, auch die Anzeige des Speicherregisters (**SR**):

Manuell



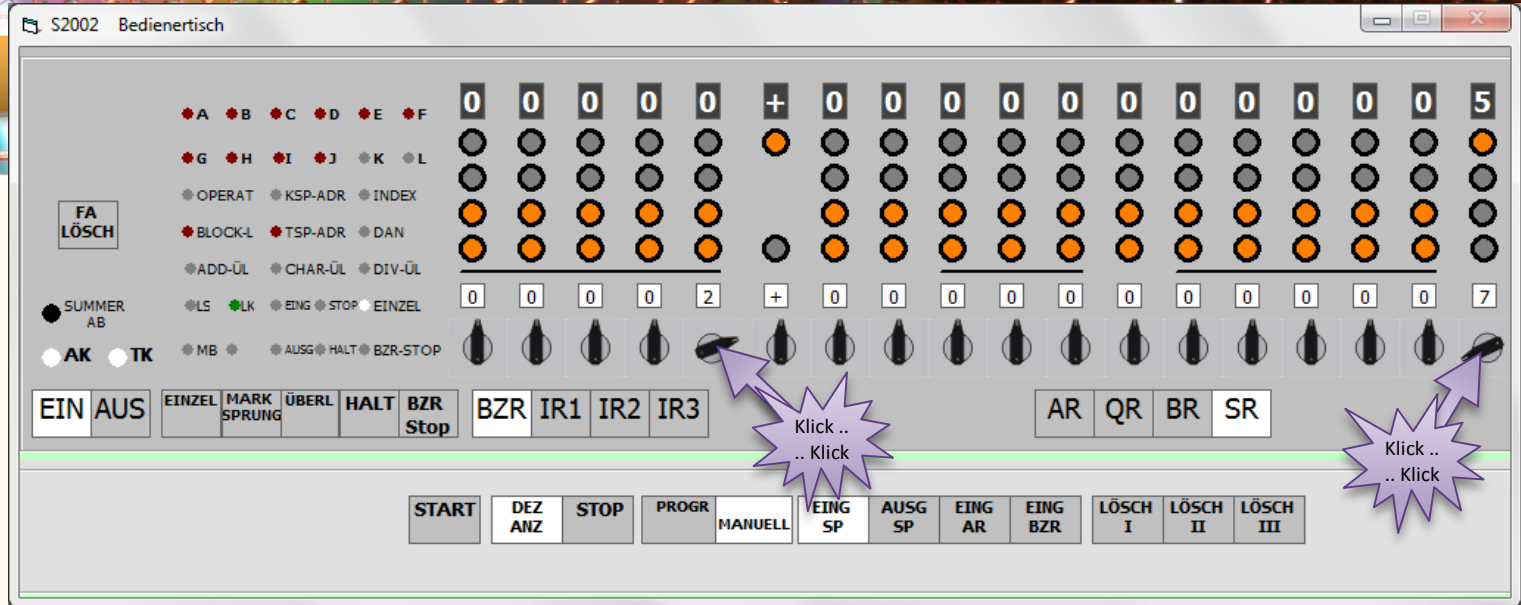
Also bewirkt das Drücken der **Start-Taste** die **Eingabe in den Speicher**, **schreibt** also den 12-stelligen Wert (5) an der eingestellten Speicheradresse (1) **in den Speicher**, und der Inhalt des Speicherregisters wird angezeigt:

Manuell



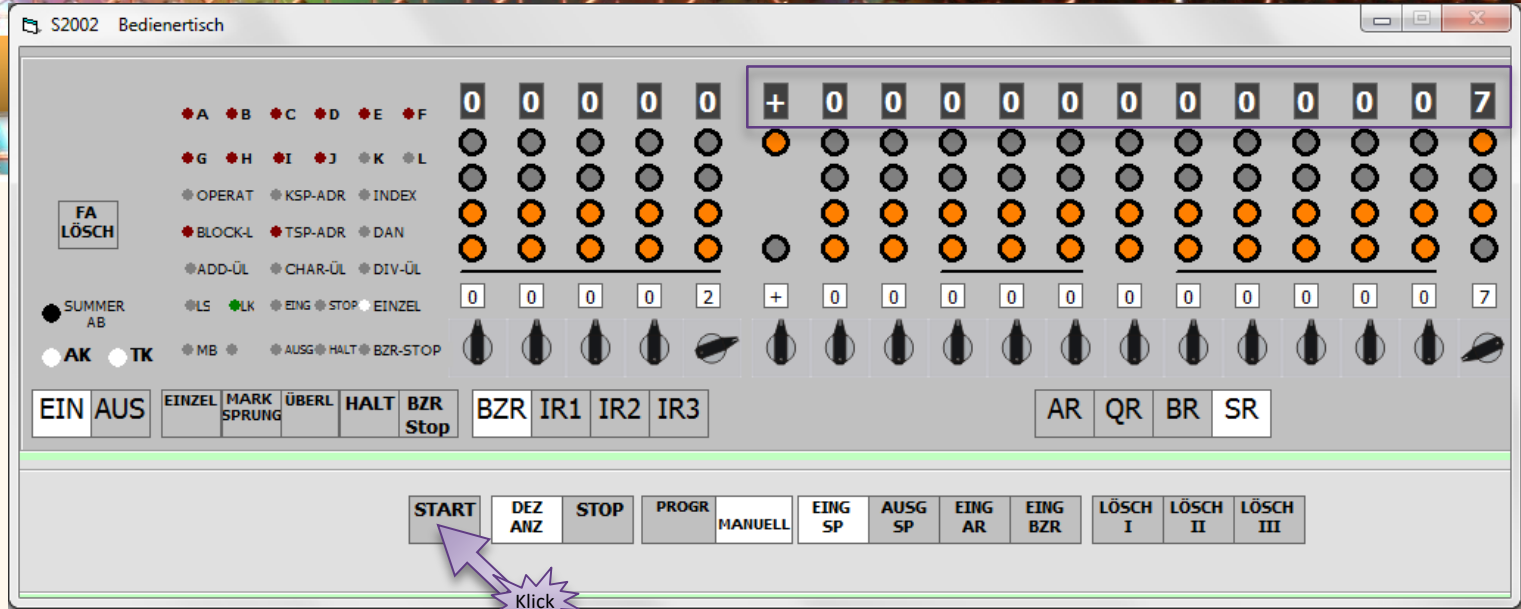
Adresse 2 soll für B den **Wert** 7 aufnehmen, **einstellen**:

Manuell



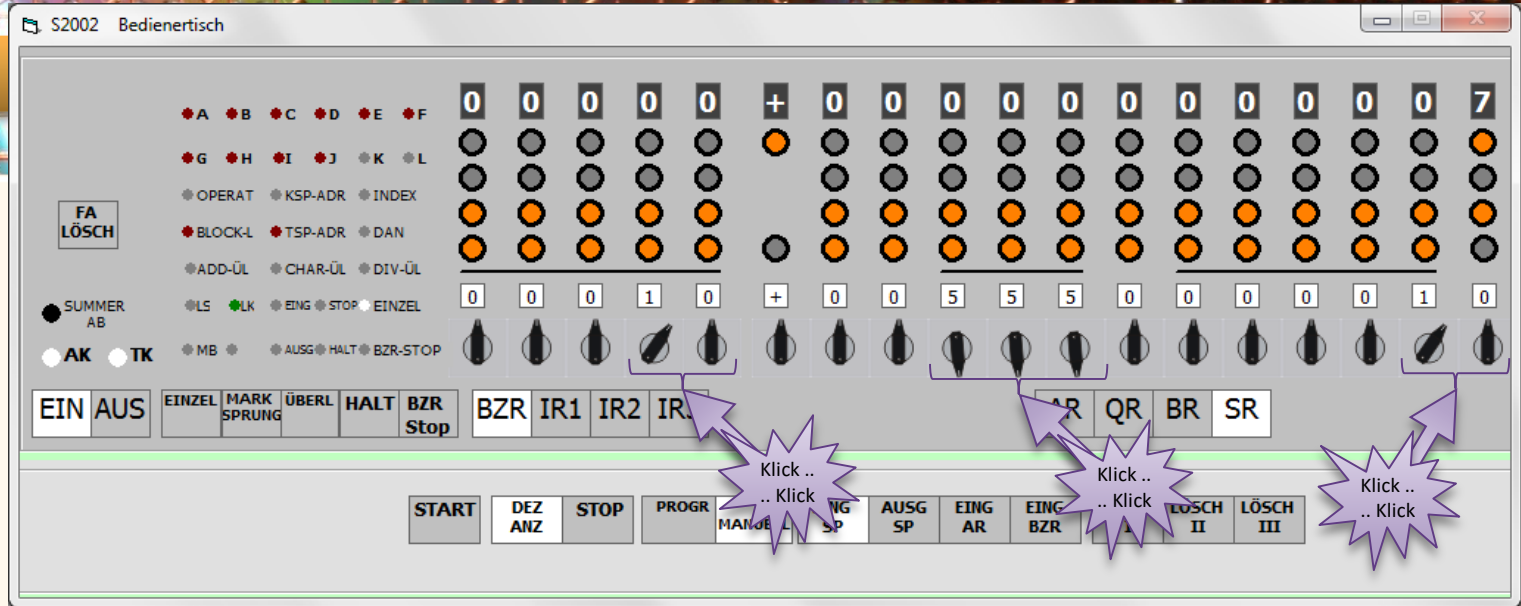
In den **Speicher** schreiben:

Manuell



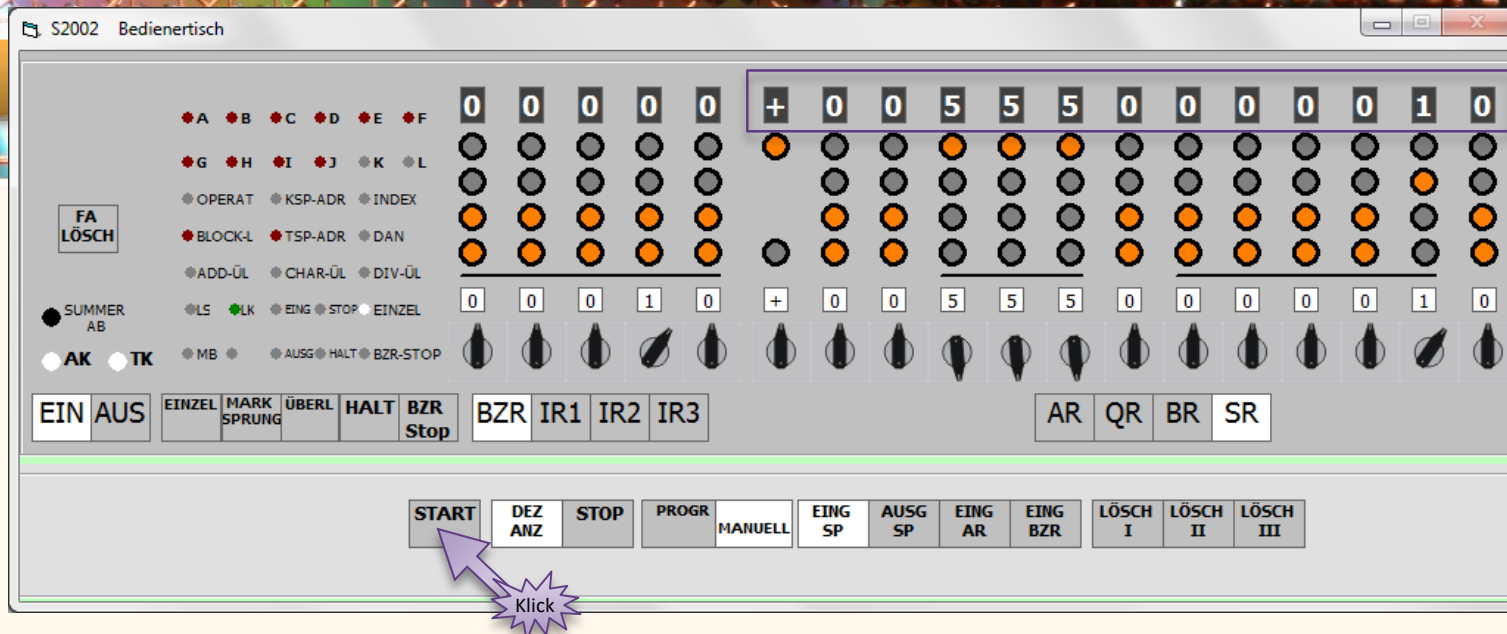
Adresse 3 soll später das Ergebnis aufnehmen, jetzt also keine Eingabe.
Adresse 10 soll den Befehl 00555000010 (TEP A) aufnehmen, **einstellen**:

Manuell



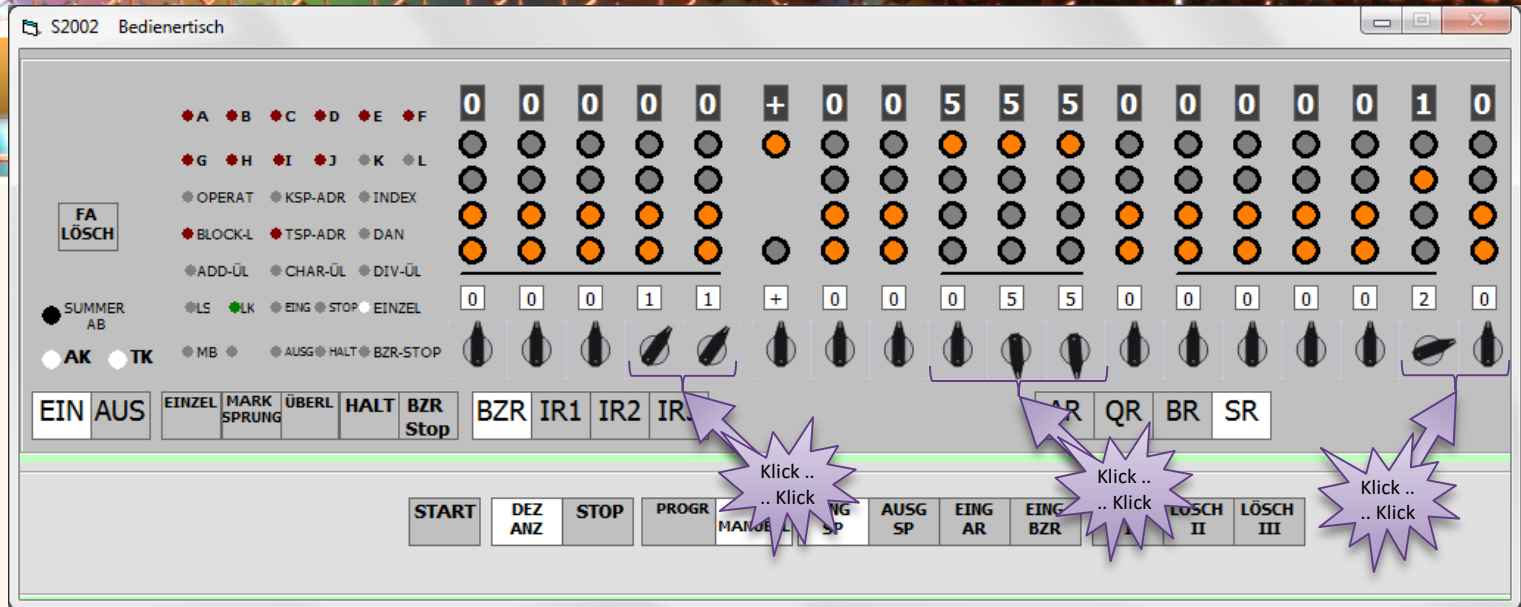
In den **Speicher schreiben**:

Manuell



Adresse 11 soll den Befehl 00055000020 (ADD B) aufnehmen, **einstellen**:

Manuell

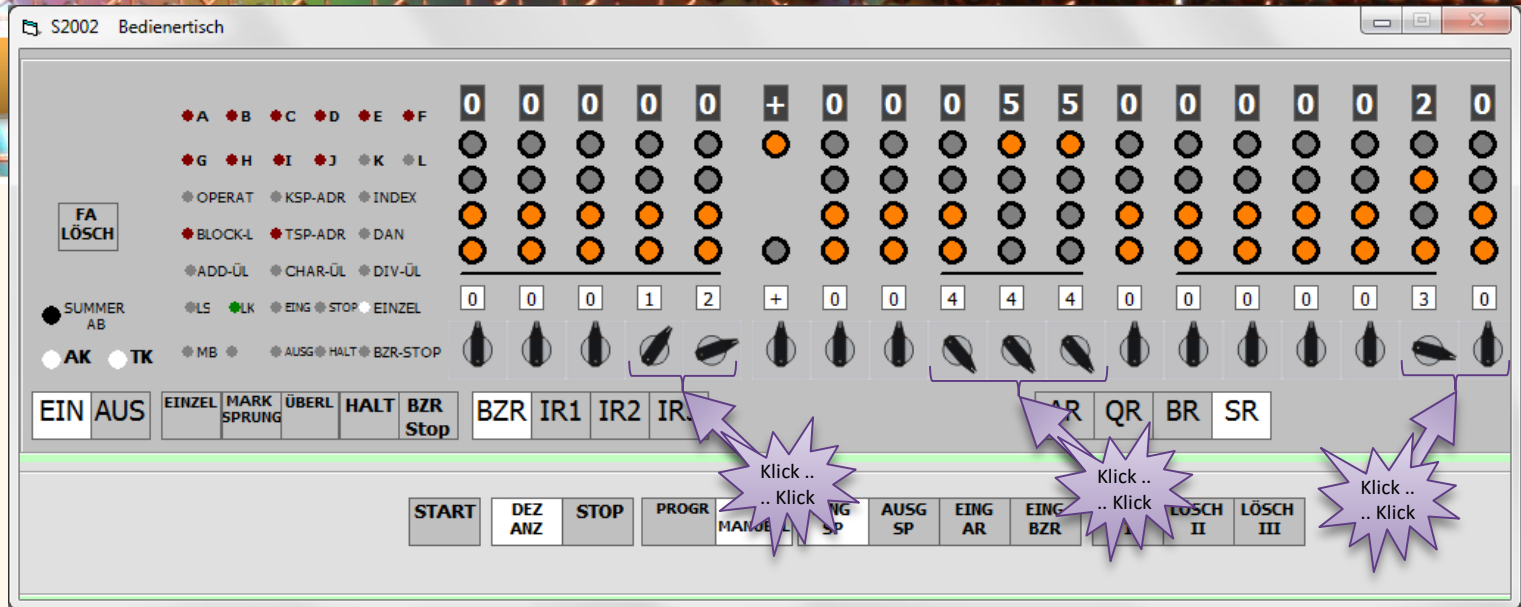


Manuell



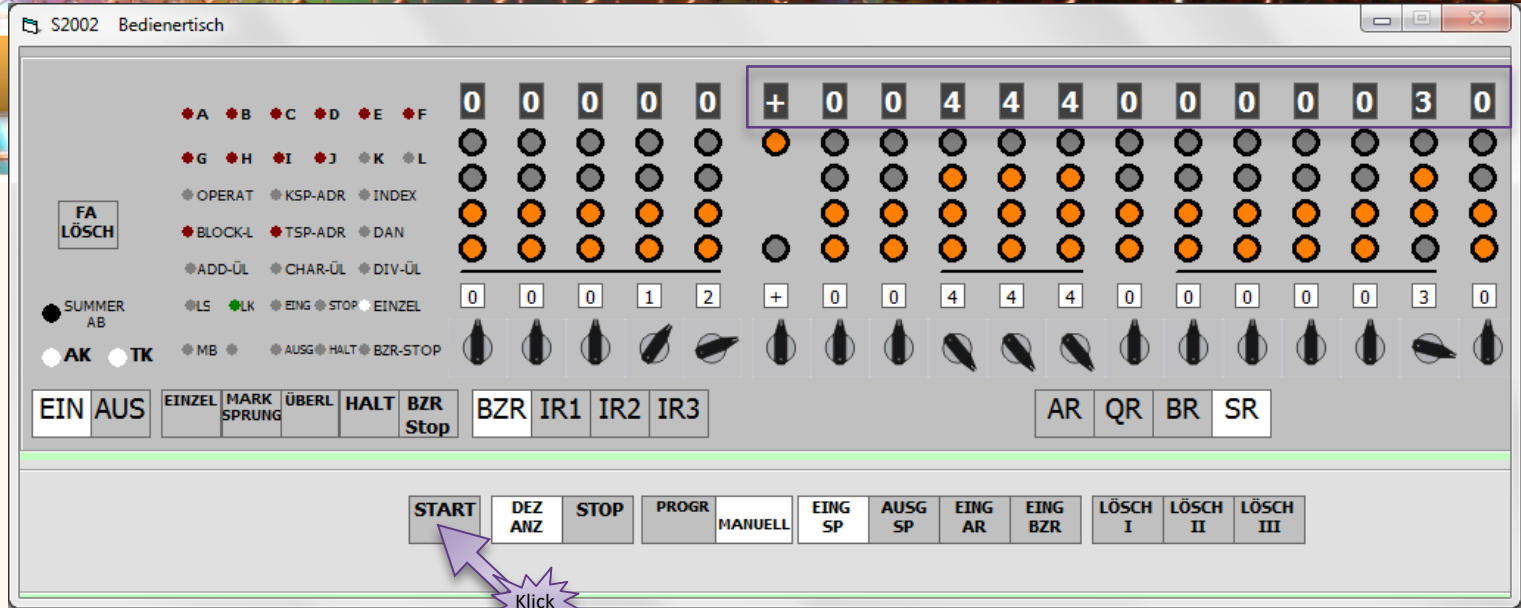
Adresse 12 soll den Befehl 00444000030 (TAS C) aufnehmen, **einstellen**:

Manuell



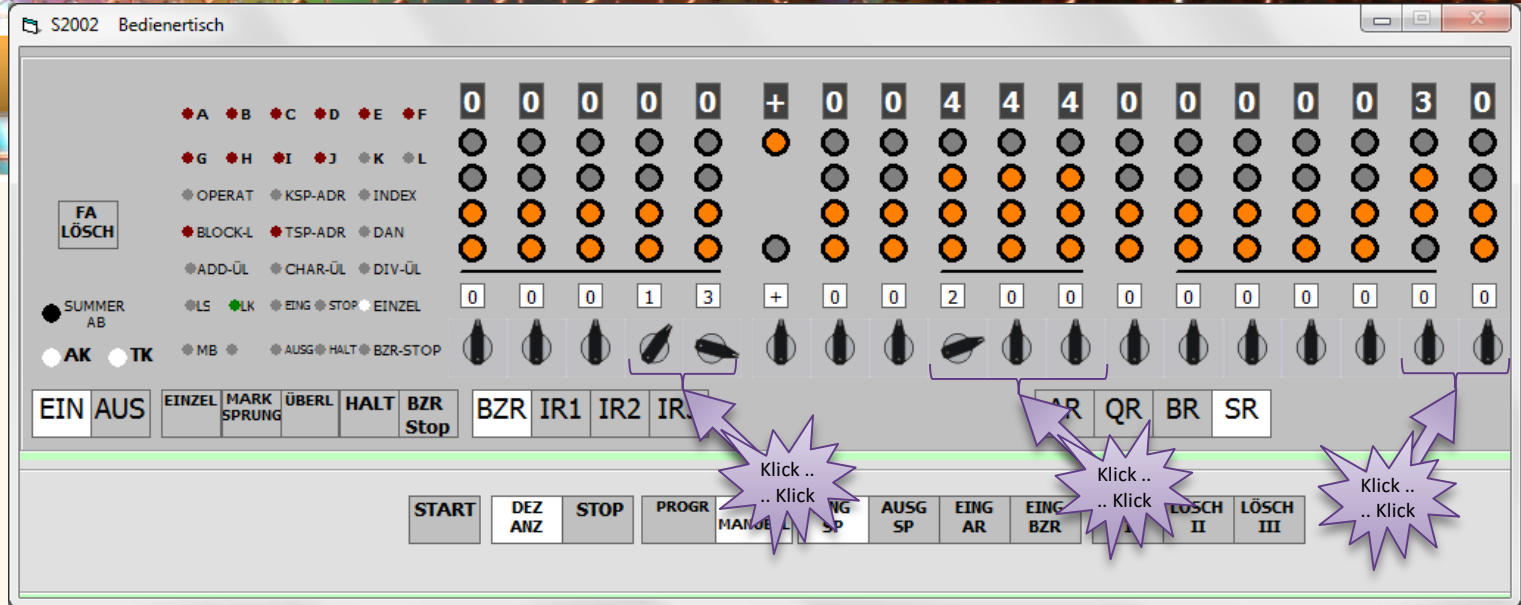
In den **Speicher schreiben**:

Manuell



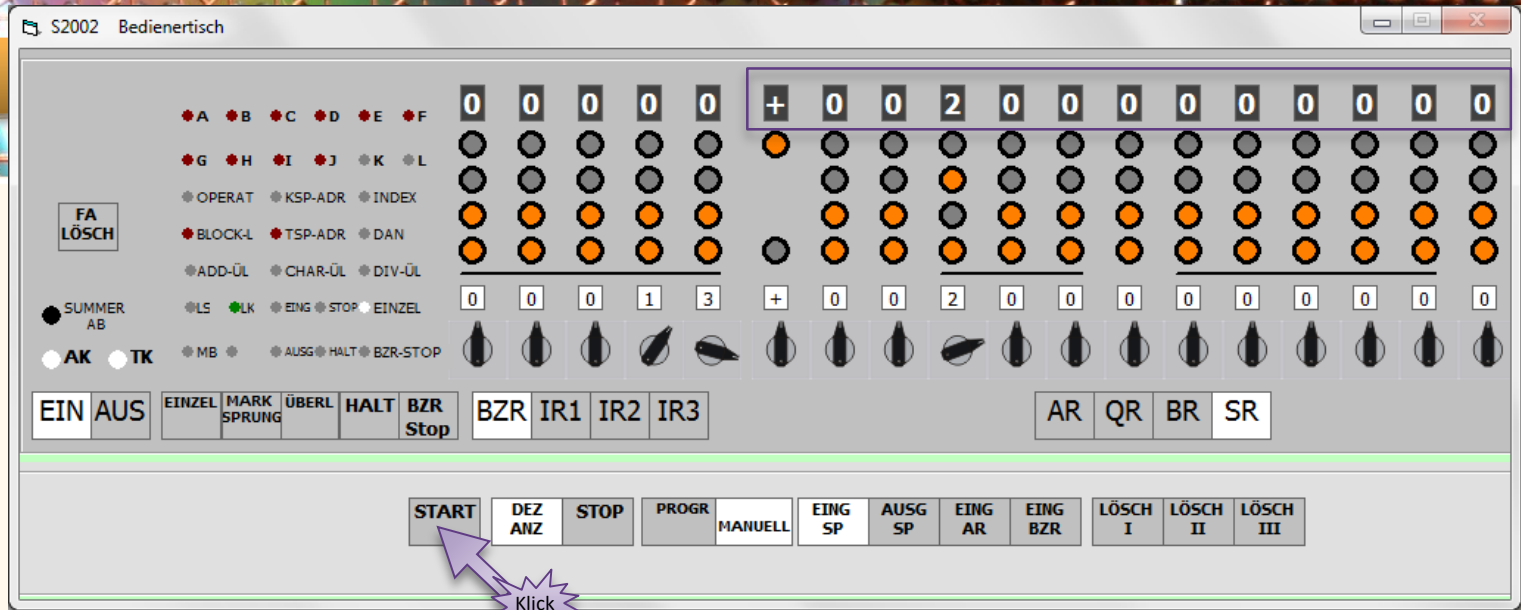
Adresse 13 soll den Befehl 002000000000 (STP) aufnehmen, **einstellen**:

Manuell



In den **Speicher schreiben**:

Manuell



Das **Programm** steht nun im **Kernspeicher**, es kann **ablaufen**.

Manuell

Für den Programm-Ablauf muss erst die **Startadresse**, das ist die Adresse des ersten Befehls, also Adresse 10, in das **BZR** (Befehlszählregister) eingegeben werden, dann muss in den **Programm-Modus** umgeschaltet werden, bevor das Drücken der **START-Taste** zum Programm-Ablauf führt.

Startadresse einstellen, das ist die Speicheradresse 10:
(das 12-stellige Drehschalterfeld kann, aber muss nicht auf 0 gesetzt werden)

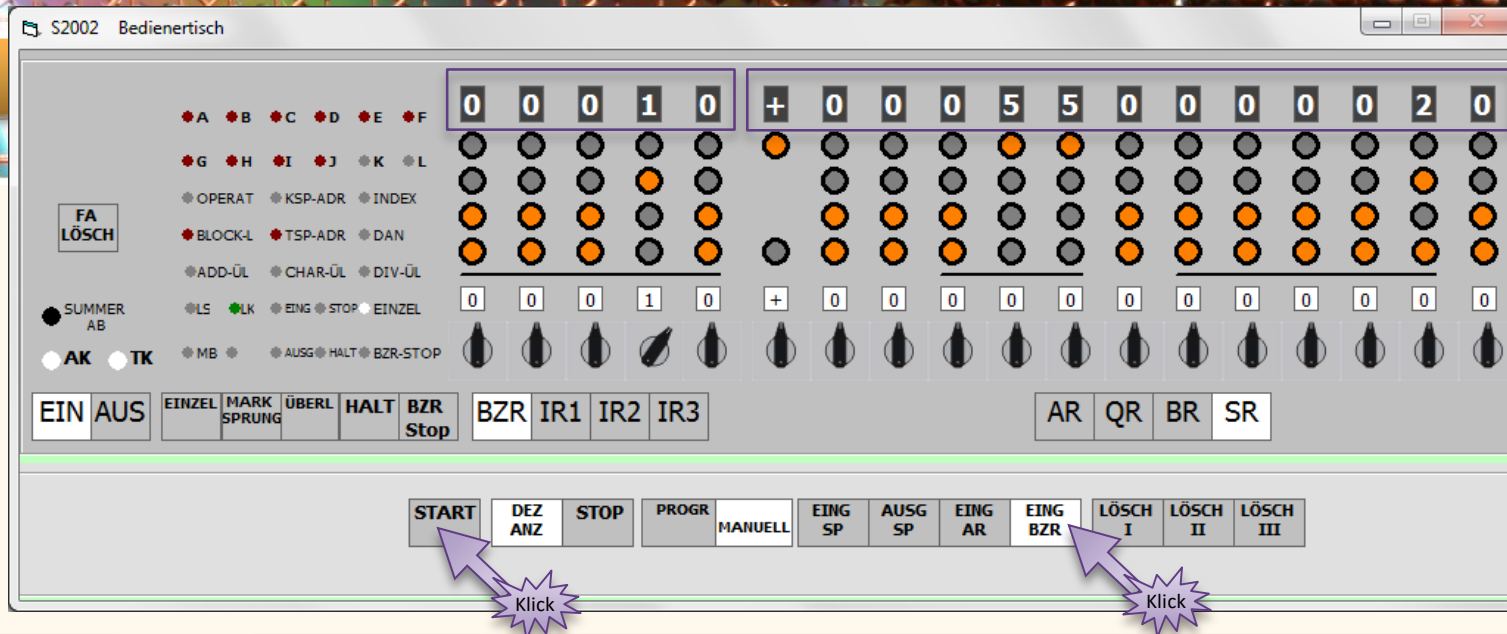
Manuell



EING BZR einschalten und mit **START** Adresse in das **BZR schreiben**.

Anzeige von BZR und SR sind eingeschaltet, in der 5-stelligen Anzeige erscheint die 10, in der 12-stelligen Anzeige der erste Befehl.

Manuell



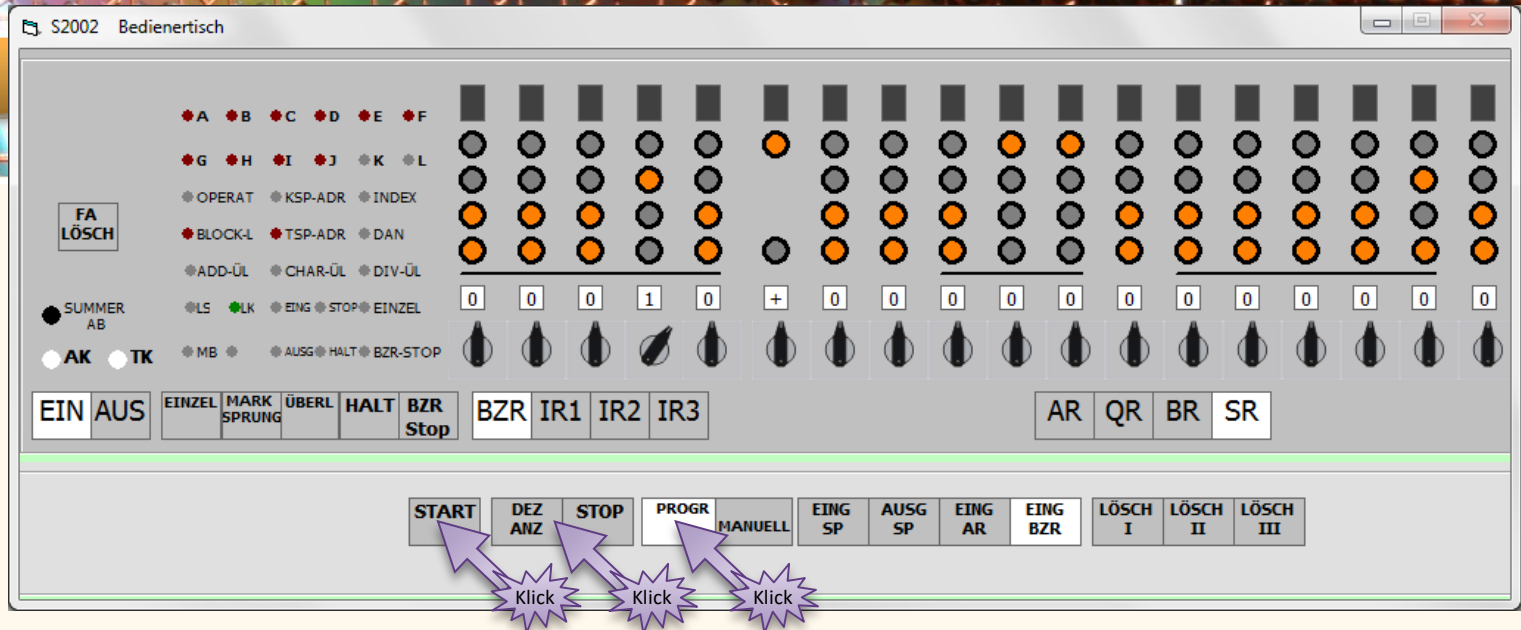
Das Programm kann **schrittweise** oder **in einem Zug** ablaufen.

Manuell

Soll das Programm **in einem Zug** ablaufen,
muss der **Einzelschrittmodus deaktiviert** werden,
das bedeutet: auch die **Dezimalanzeige** muss **ausgeschaltet** werden.

Dezimal-Anzeige (**DEZ ANZ**) ausschalten,
Programm-Modus (**PROGR**) einschalten,
und mit **START** das Programm starten:

Manuell

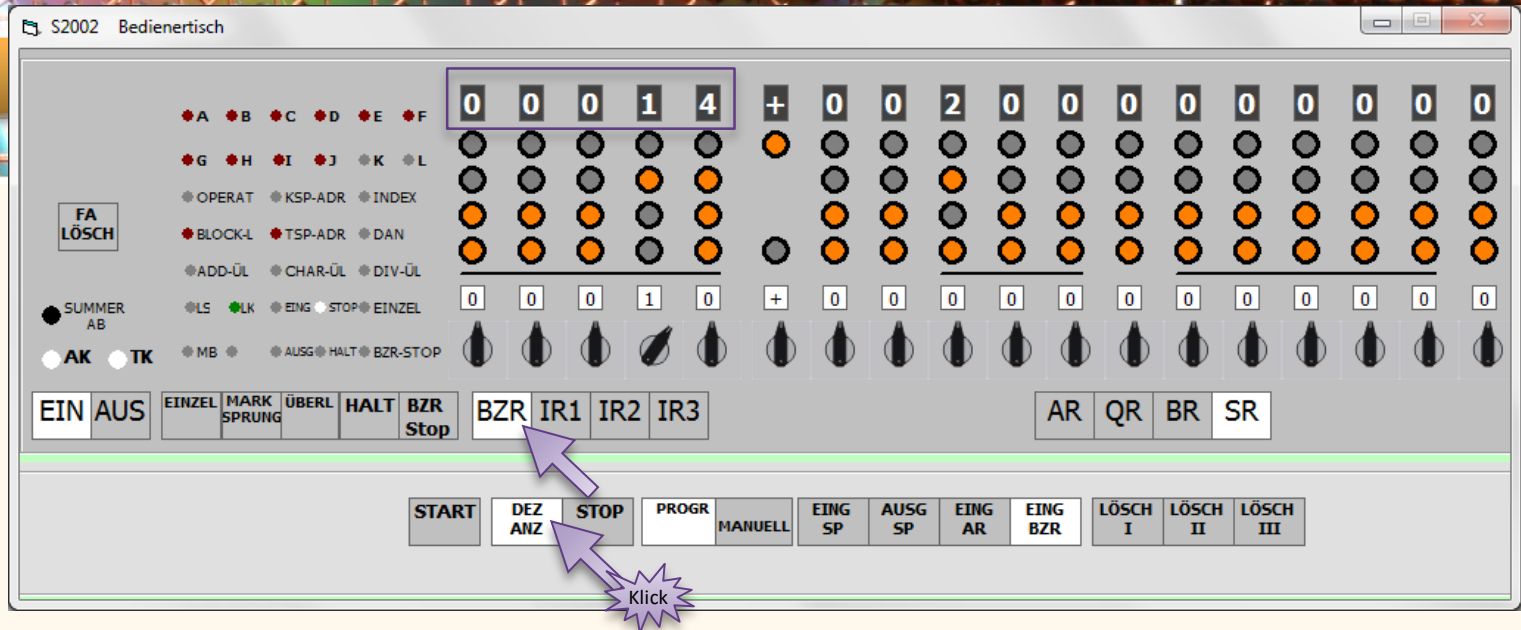


Die Anzeige (**STOP**) zeigt an, dass der Befehl **STP** abgearbeitet wurde.

The image shows a control panel for a machine, labeled "S2002 Bedientisch". It features a grid of buttons and indicators. At the top, there are buttons labeled A through L, each with a corresponding colored dot. Below these are buttons for "OPERAT", "KSP-ADR", "INDEX", "BLOCK-L", "TSP-ADR", "DAN", "ADD-ÜL", "CHAR-ÜL", and "DIV-ÜL". A "SUMMER AB" button is on the left, and "AK" and "TK" buttons are below it. A "FA LÖSCH" button is in the top left corner. A row of buttons labeled "EINZEL", "MARK SPRUNG", "ÜBERL", "HALT", "BZR Stop", "BZR", "IR1", "IR2", "IR3", "AR", "QR", "BR", and "SR" is located in the middle. At the bottom, there are buttons for "START", "DEZ ANZ", "STOP", "PROGR", "MANUELL", "EING SP", "AUSG SP", "EING AR", "EING BZR", "LÖSCH I", "LÖSCH II", and "LÖSCH III". A mouse cursor is pointing at the "EING STOP" button. The panel also includes several indicator lights and a row of buttons labeled 0 through 9, each with a corresponding colored dot.

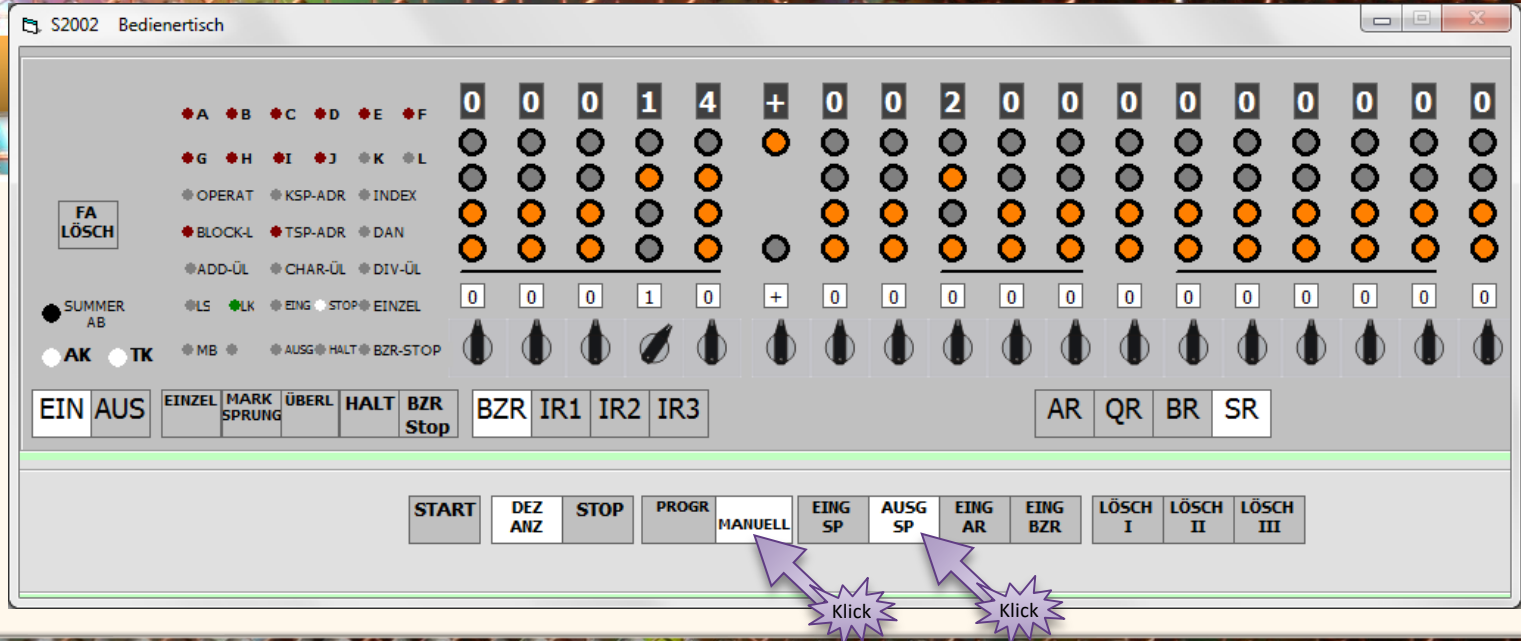
Dezimal-Anzeige (**DEZ ANZ**) für einfaches Ablesen einschalten.
Kontrolle: Die Anzeige des Befehlszählregisters (**BZR**) ist eingeschaltet.
Es steht auf Adresse 14, also sind alle Befehle abgearbeitet worden.

Manuell



Vorbereitung für die Anzeige des Ergebnisses:
Umschalten in den manuellen Modus (**MANUELL**).
Aktion Ausgabe Speicher (**AUSG SP**) einschalten.

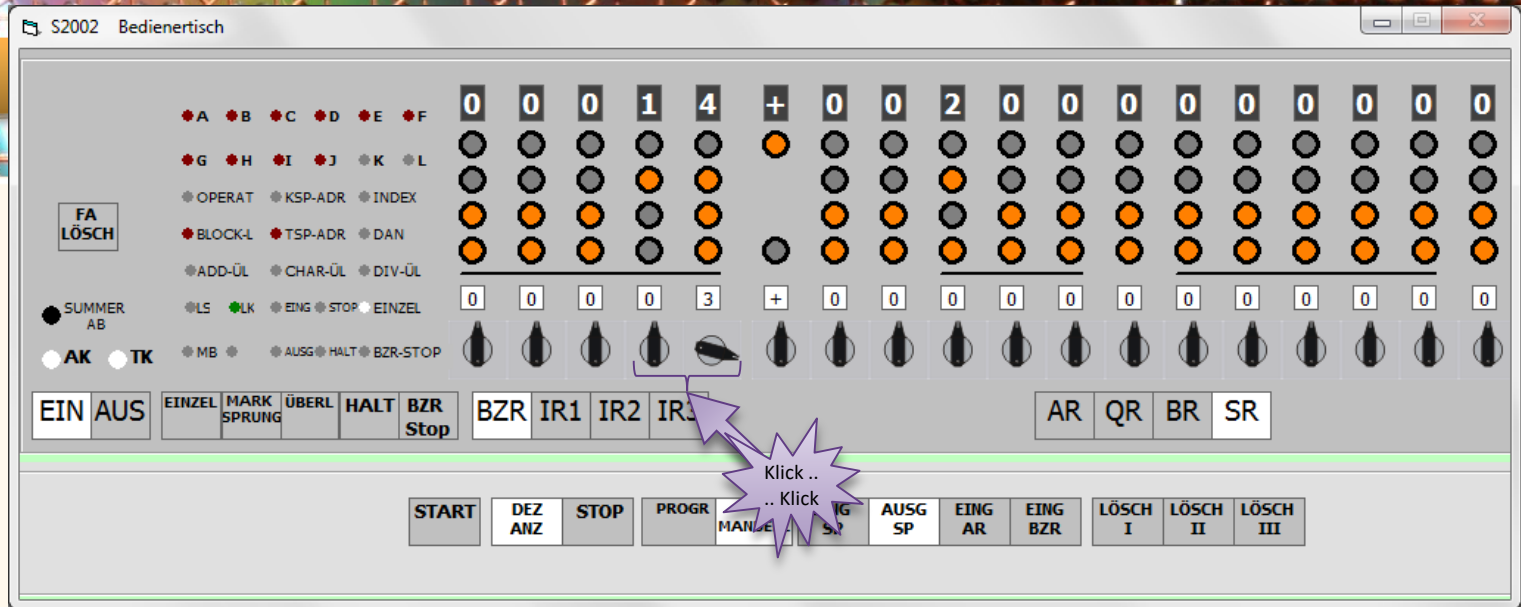
Manuell



In Speicherzelle C, also an Speicheradresse 3 sollte das Ergebnis stehen.

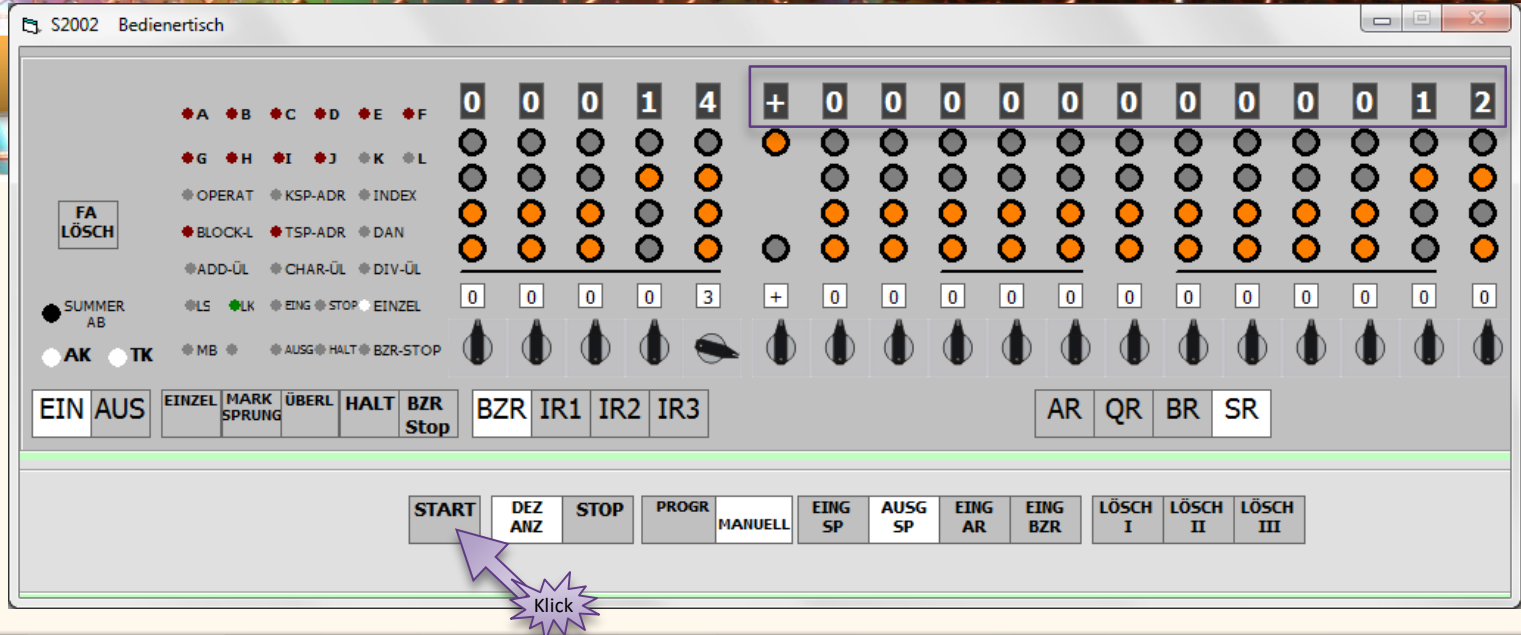
Adresse einstellen:

Manuell



Mit **START** den Inhalt anzeigen – im Speicher steht 12 – richtig!

Manuell



Die **Größenordnung** des Ergebnisses 12 ist dabei **Interpretationssache**!

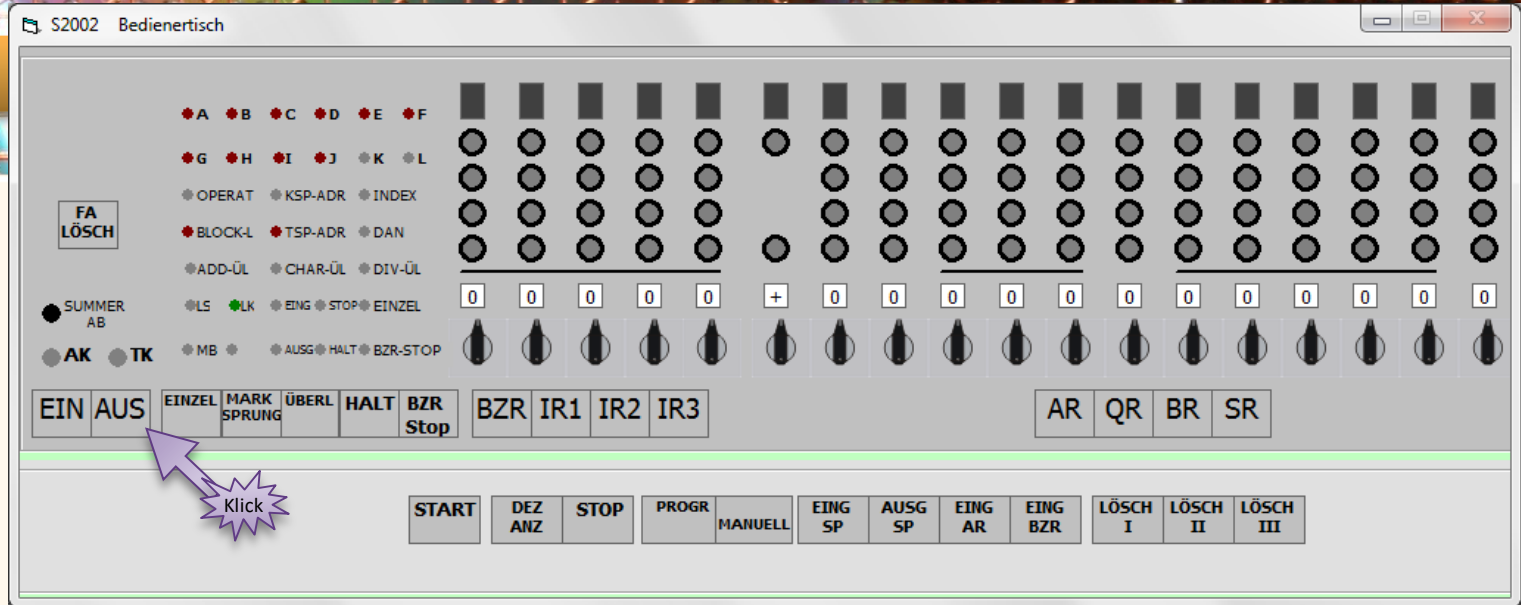
Manuell

Vereinbarung ist bei der Festkomma-Arithmetik der Siemens 2002:
Das Komma steht ganz **links** vor der höchstwertigen Stelle!

Also hat das Beispiel nicht **$5+7=12$** berechnet,
sondern **$5*10^{-12}+7*10^{-12}=12*10^{-12}$** !

Die Rechenanlage kann wieder ausgeschaltet werden (**AUS**),
vorher sollten aber immer alle Drehschalter auf 0 gesetzt werden.

Manuell



SIEMENS 2002

Beispiel 2 Zügig

Für alle Beispiele wurde die **Emulation** der **Siemens 2002** auf dem PC unter Windows von **Volker Dahrendorf** benutzt.



Zügig

Beispiel 2 demonstriert die Vorgehensweise bei der normalerweise genutzten Art der Programm-Eingabe per **Lochstreifen**. Hier wird ein Lochstreifen erstellt, der den Programmcode in Maschinensprache in **Zügig-Codierung** enthält. Der Lochstreifen wird dann im **Zügig-Modus** eingelesen, der Maschinencode direkt in den Kernspeicher schreibt.

Das Mini-Demo-Programm soll mit **Gleitkomma-Arithmetik** (unnormalisiert) **zwei Werte addieren**.

Beispiel 2: A plus B ergibt C, in Basic: **C=A+B**, in Algol: **C:=A+B;**

Zügig

in **Assembler**, bei Entscheidung für **Gleitkomma-Arithmetik**
(dabei steht im Folgenden die Abkürzung **AR** für das akkumulative Register):

TEP	A	Lade das AR mit dem Inhalt der Speicherzelle A
GAU	B	Addiere unnormalisiert zum AR den Inhalt der Speicherzelle B
TAS	C	Schreibe den Inhalt des AR in die Speicherzelle C
STP		Halte an - fertig

Die Addition soll in **Gleitkomma-Arithmetik** durchgeführt werden

Zügig

Die Zahlen, hier im Beispiel 5 und 7, müssen also korrekt mit **Mantisse** und **Charakteristik** in die Speicherzellen geschrieben werden:

Der Wert **5**, d. h. $5 \cdot 10^0$ muss als **$0,5 \cdot 10^1$** gespeichert werden, in S2002-Schreibweise ist das **500000000051**.

Der Wert **7**, d. h. $7 \cdot 10^0$ muss als **$0,7 \cdot 10^1$** gespeichert werden, in S2002-Schreibweise ist das **700000000051**.

Die Assembler-Befehle müssen in **Maschinencode** übersetzt werden und für die **symbolischen** Namen der Speicherzellen A, B und C müssen **absolute** Speicheradressen festgelegt werden.

Zügig

Adresse	Inhalt	Erläuterung
00001	500000000051	Speicheradresse für A, hier der Wert $0,5 \cdot 10^1$
00002	700000000051	Speicheradresse für B, hier der Wert $0,7 \cdot 10^1$
00003		Speicheradresse für das Ergebnis C
00010	005550000010	Befehl TEP A
00011	000750000020	Befehl GAU B
00012	004440000030	Befehl TAS C
00013	002000000000	Befehl STP

Nun kann der **Lochstreifen** erstellt werden.
Damals standen dafür **Fernschreiber** (z. B. vom Typ **Siemens T100**) (offline)
in einem separaten Raum zur Verfügung.

Zügig

Auch bei der **Emulation** kann der Fernschreiber benutzt werden, ohne dass
die S2002 eingeschaltet wird. Der Fernschreiber kann jedoch auch (online) als
Ausgabegerät Blattschreiber genutzt werden.

Das **virtuelle Eintippen** am Fernschreiber wird simuliert durch Übernahme
einer vorbereiteten Text-Datei.

Die Eingabe am Fernschreiber – bei der Emulation die vorbereitete Textdatei – muss genau den **Vorschriften** für den Aufbau des **Züig-Lochstreifens** (Formulars) folgen.

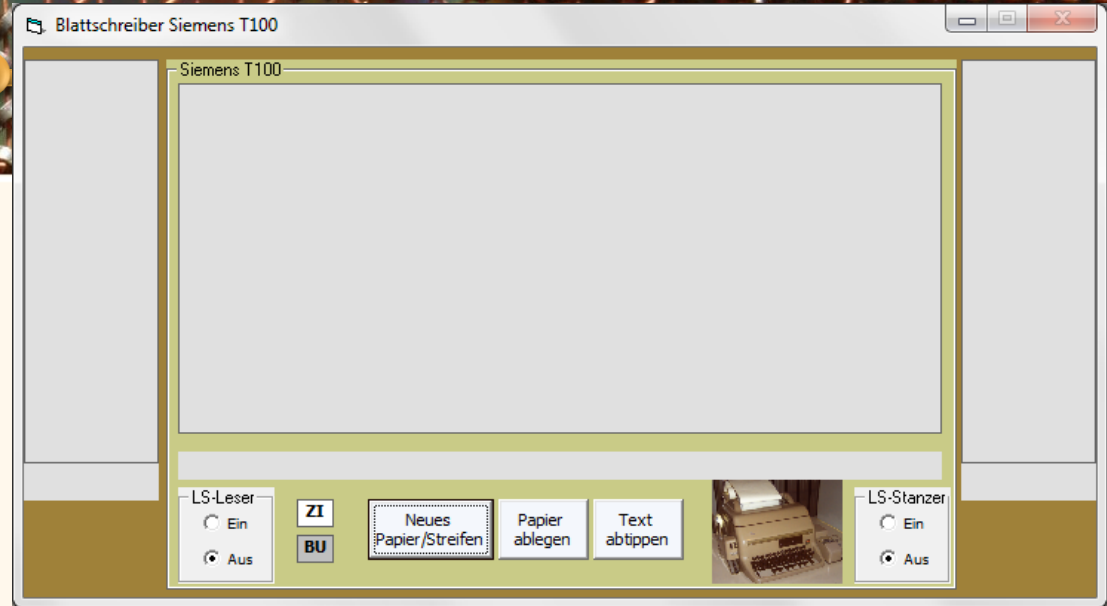
Züig

```
00001+500000000051.A  
+700000000051.B  
00010+005550000010.BEFEHL TEP A  
+000750000020.BEFEHL GAU B  
+004440000030.BEFEHL TAS C  
+002000000000.BEFEHL STP  
00010+ ENDE .ENDE DES CODES, BZR FUER PROGRAMMSTART
```

Fernschreiber (Blattschreiber Siemens T100) auswählen.

Zügig

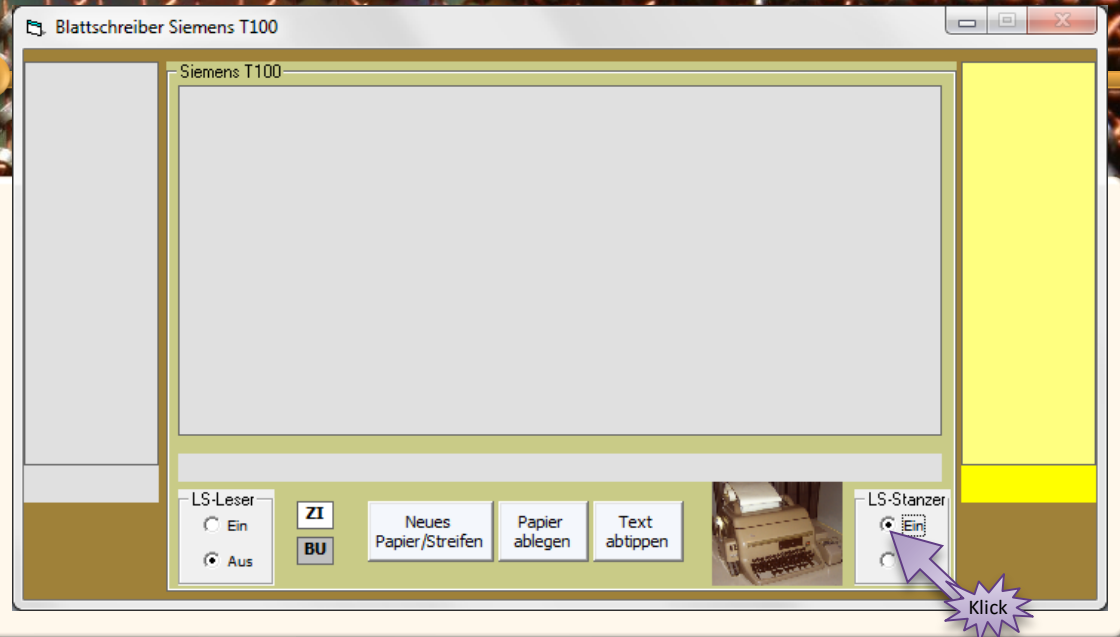
In der Emulation das
Fenster des Fernschreibers
auswählen.



Lochstreifen-**Stanzer** einschalten.

Zügig

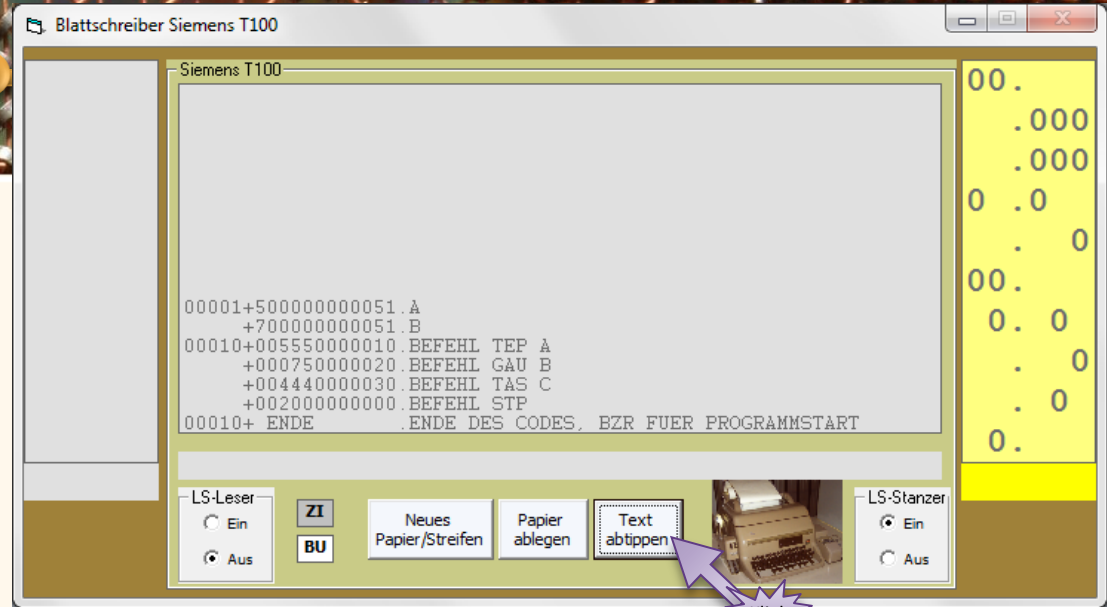
Linksklick auf **LS-Stanzer**
Ein schaltet die Produktion
des virtuellen Lochstreifens
ein.



Maschinencode im Zügig-Format virtuell eintippen.

Zügig

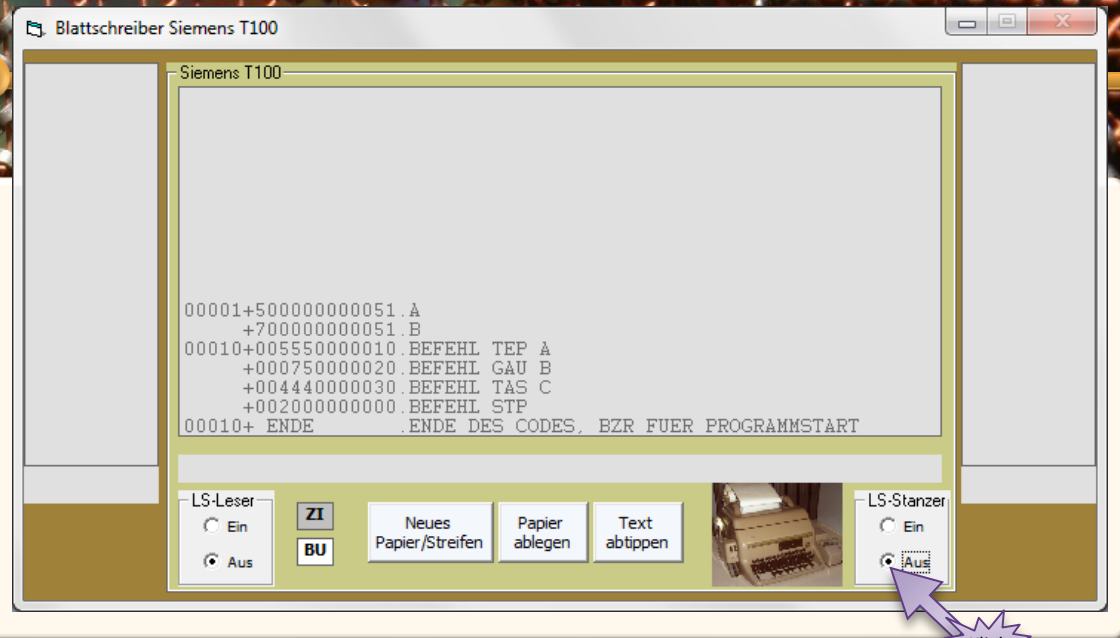
Linksklick auf Schaltfläche **Text abtippen** öffnet den Dialog zur Auswahl der vorbereiteten Text-Datei, die dann Zeichen für Zeichen eingelesen und angezeigt wird.



Lochstreifen abreißen.

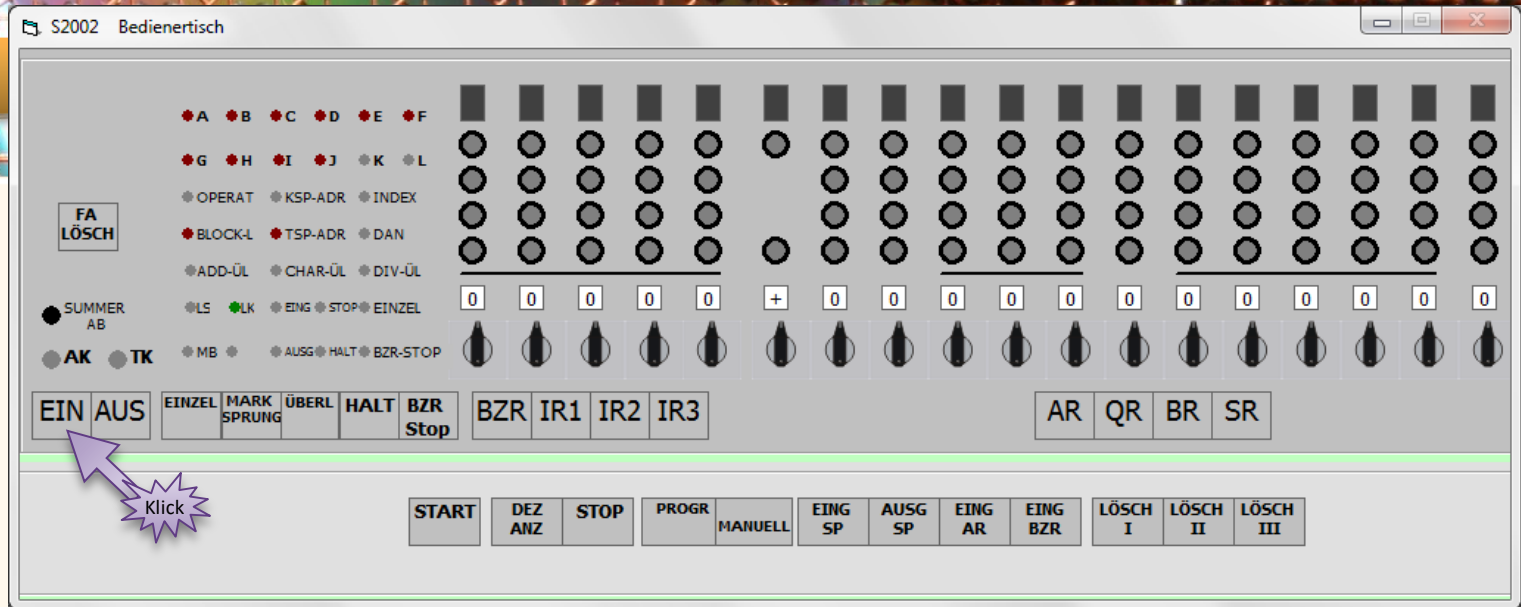
Zügig

Linksklick auf **LS-Stanzer Aus** öffnet den Dialog zum Speichern der Lochstreifendatei.



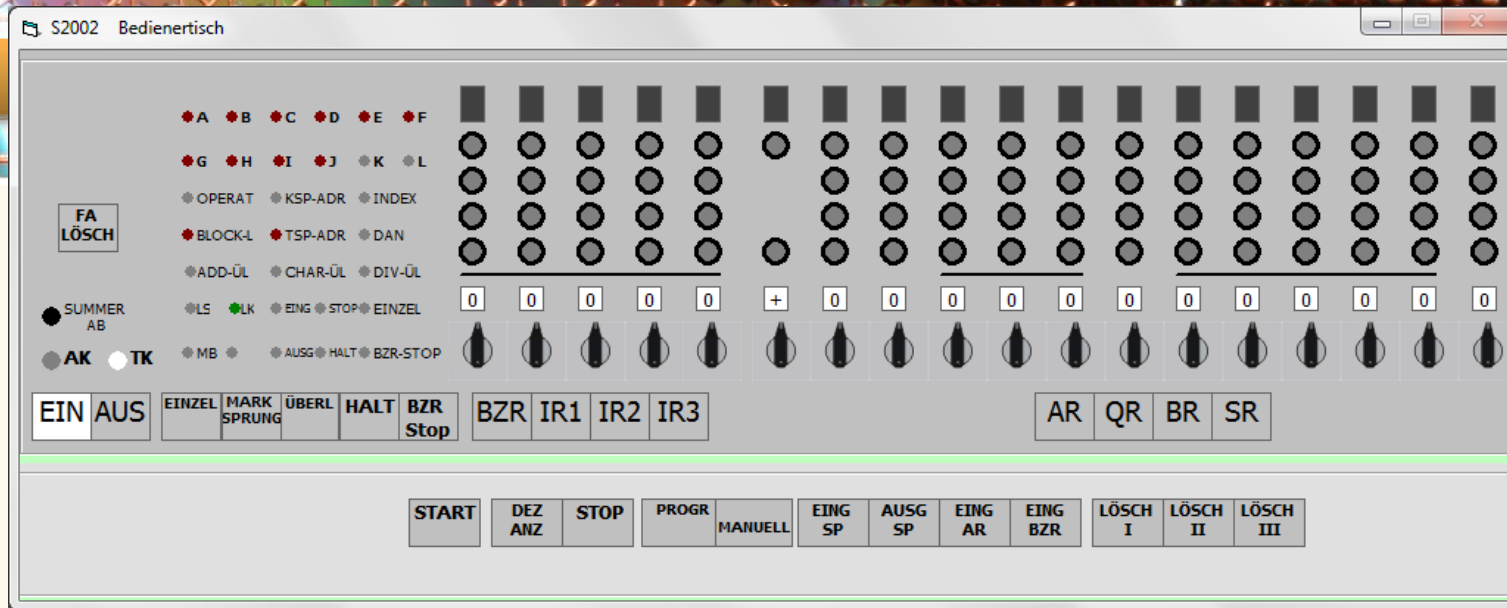
Rechenanlage einschalten (**EIN**), warten bis die Anlage betriebsbereit ist.

Zügig



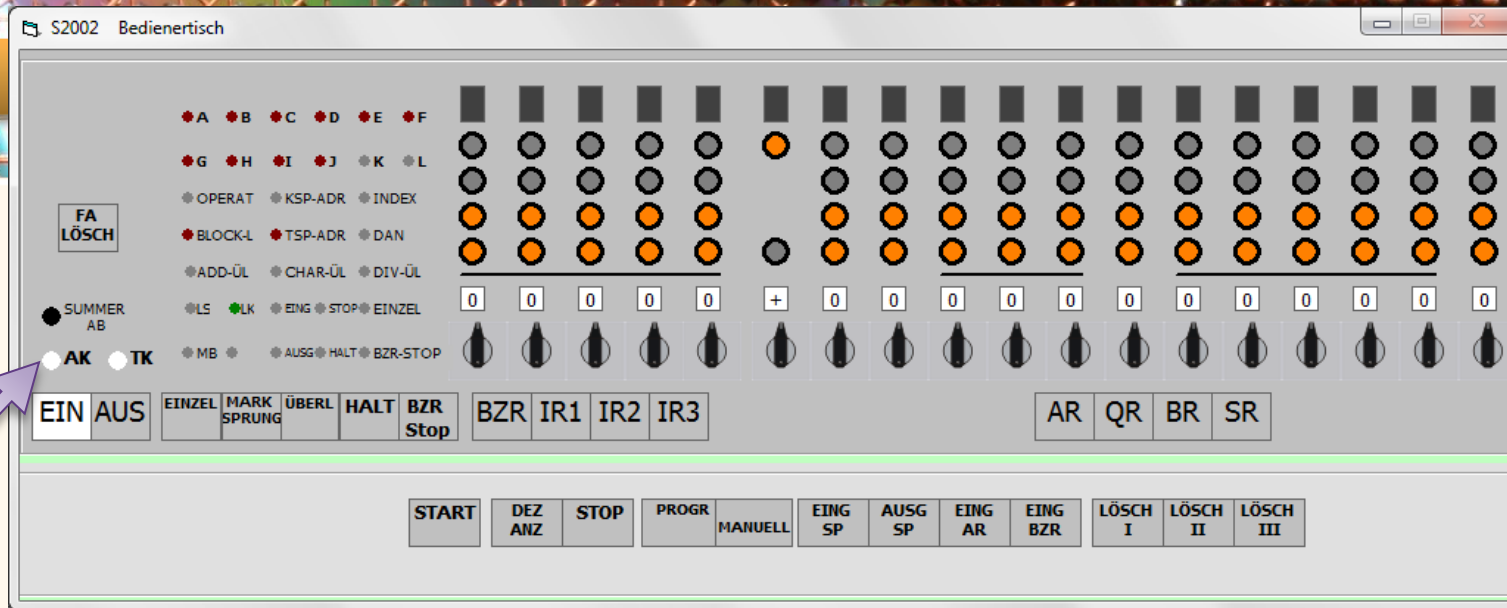
Rechenanlage einschalten (**EIN**), warten bis die Anlage betriebsbereit ist.

Zügig



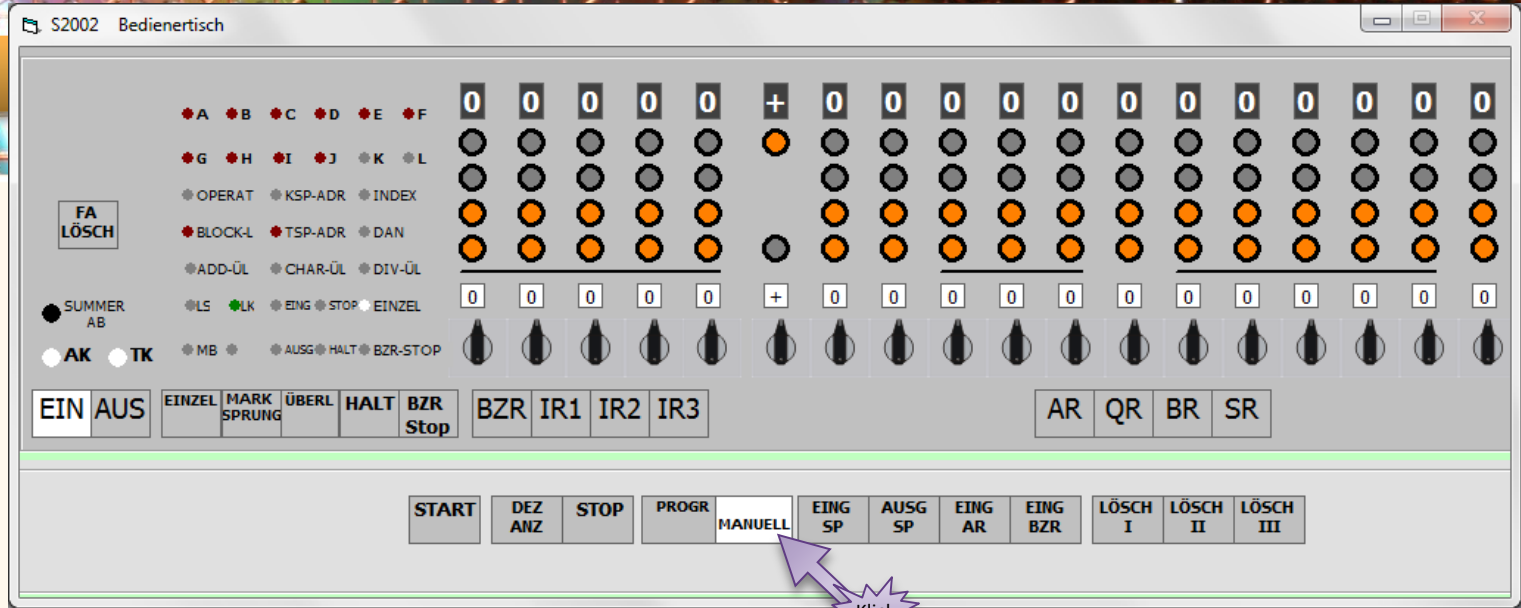
Rechenanlage einschalten (**EIN**), warten bis die Anlage betriebsbereit ist.

Zügig



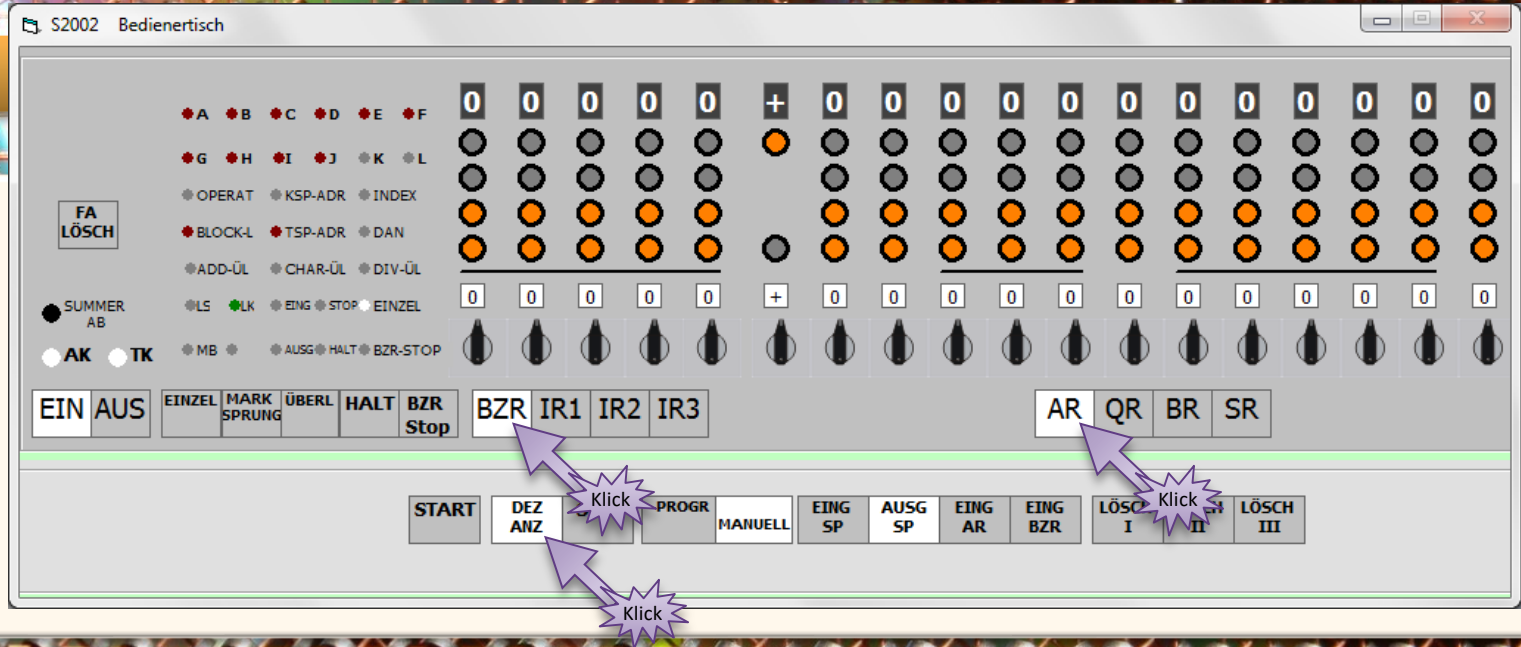
Manuellen Modus (**MANUELL**) einschalten.

Zügig



DEZ ANZ sowie Anzeige von **BZR** und **AR** sind praktisch, aber nicht nötig.

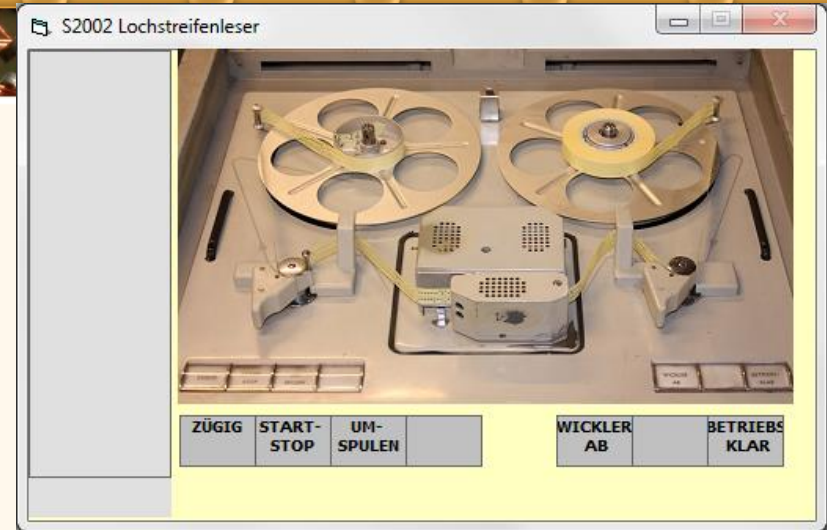
Zügig



Lochstreifenleser auswählen.

Zügig

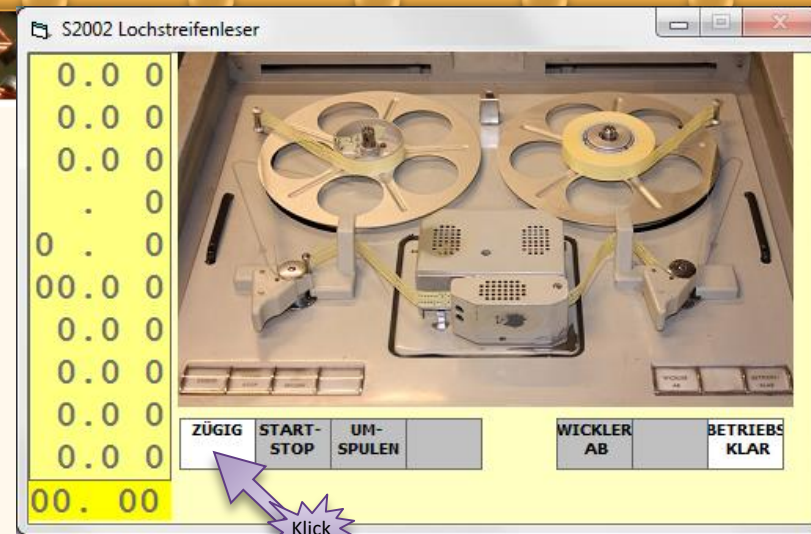
In der Emulation das Fenster des
Lochstreifenlesers auswählen.



Lochstreifen einlegen und **Zügig-Modus** einschalten.

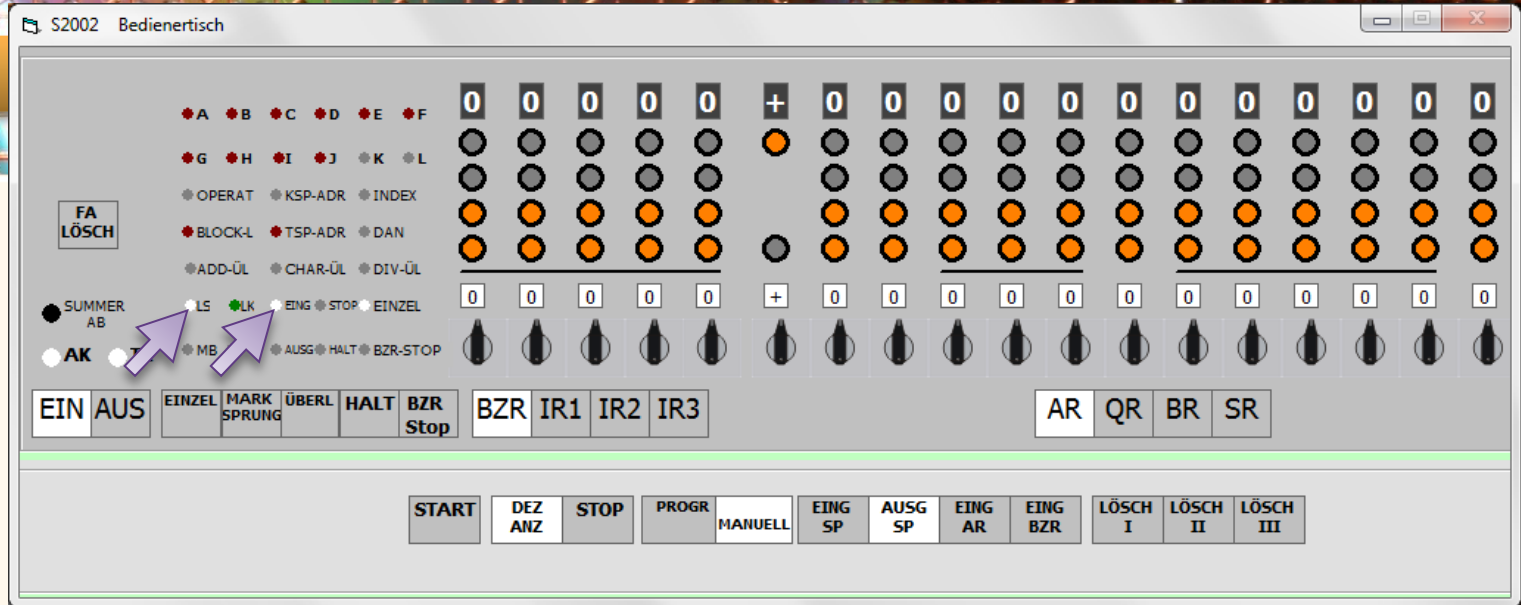
Zügig

Linksklick auf Schaltfläche
ZÜGIG öffnet den Dialog
zur Auswahl der
Lochstreifen-Datei.



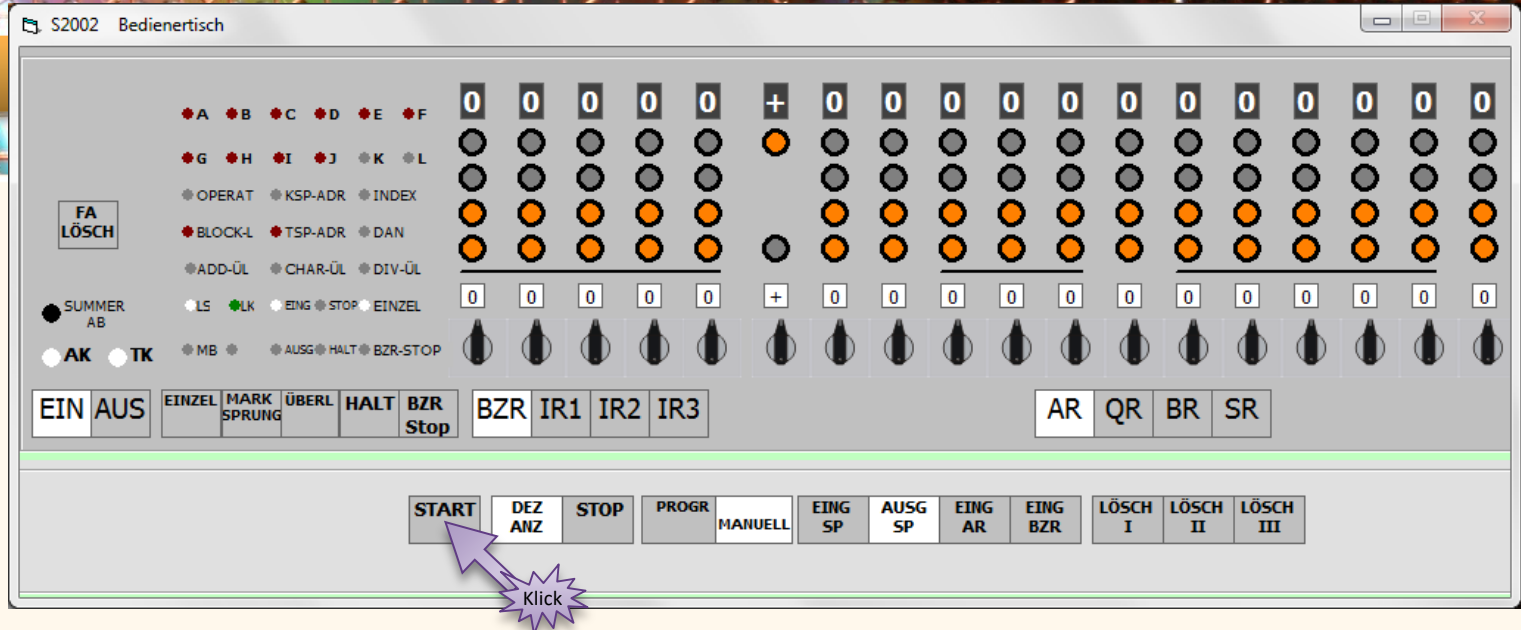
Am Bedientisch leuchten die Anzeige-Lämpchen **LS** und **EING** auf.

Zügig



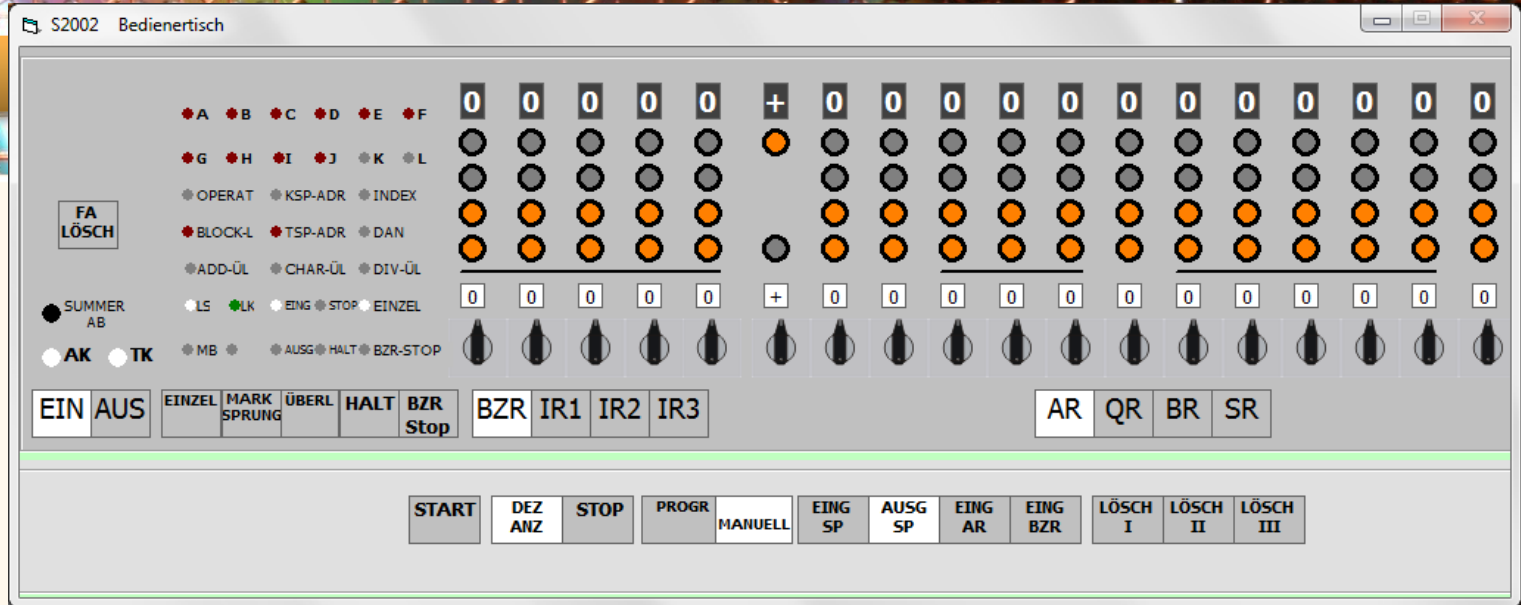
START startet das Einlesen des Lochstreifens im Zügig-Modus.

Zügig



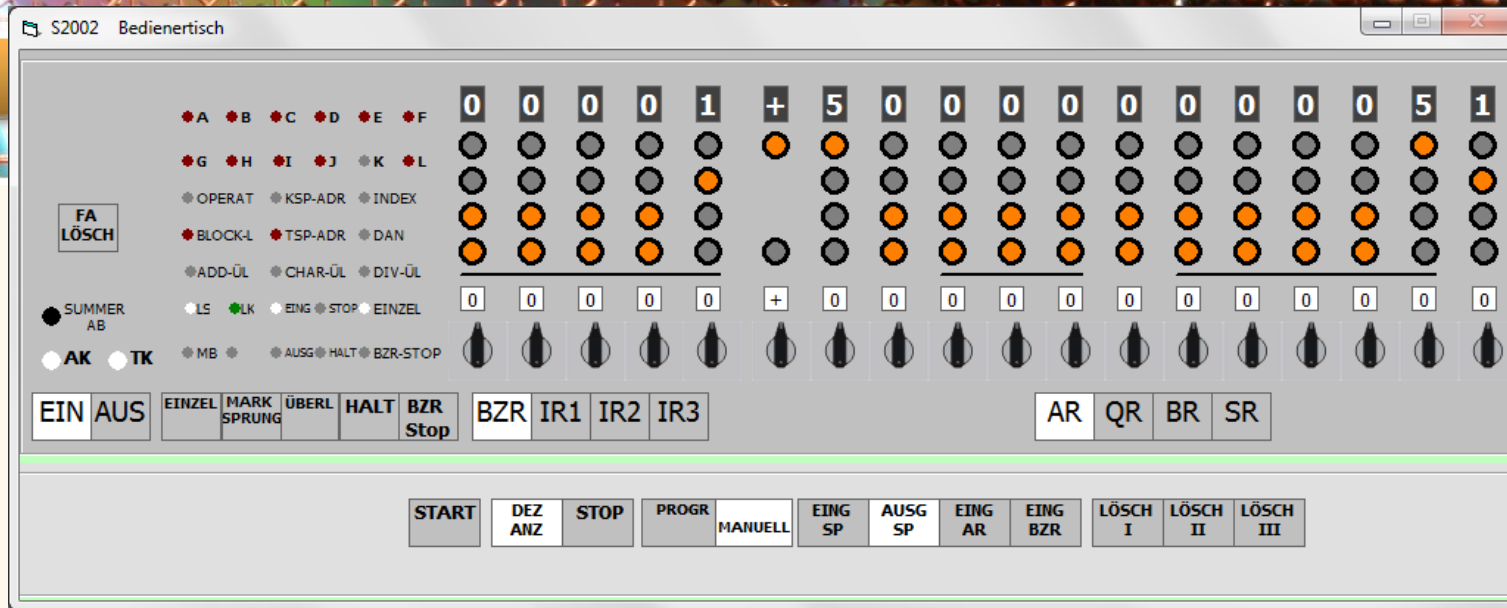
Während des Einlesens laufen die Adressen und Inhalte durch die Anzeige.

Zügig



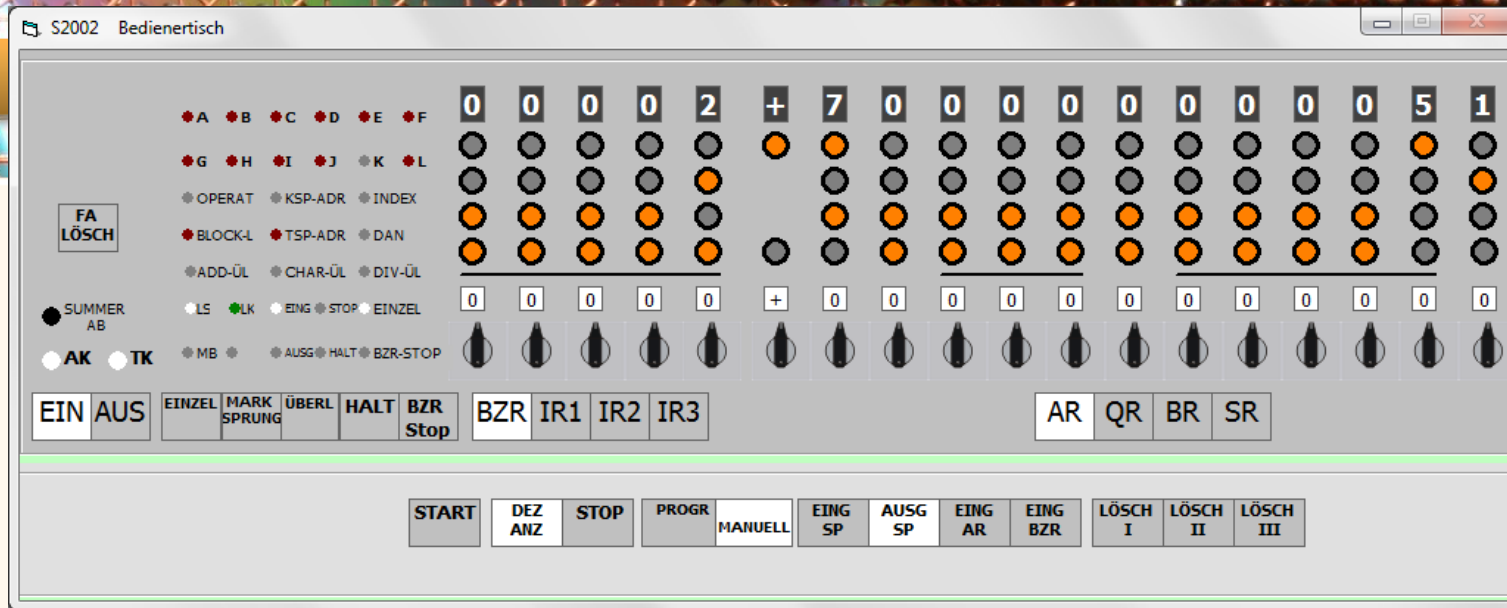
Während des Einlesens laufen die Adressen und Inhalte durch die Anzeige.

Zügig



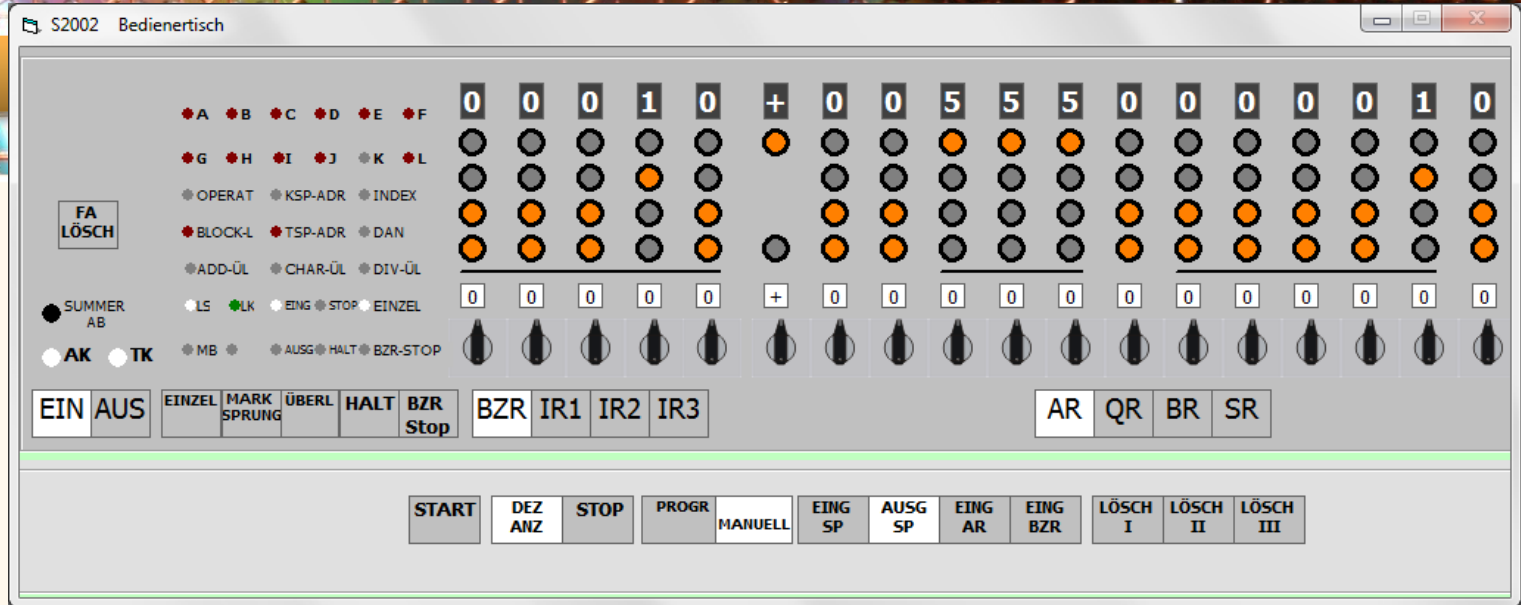
Während des Einlesens laufen die Adressen und Inhalte durch die Anzeige.

Zügig



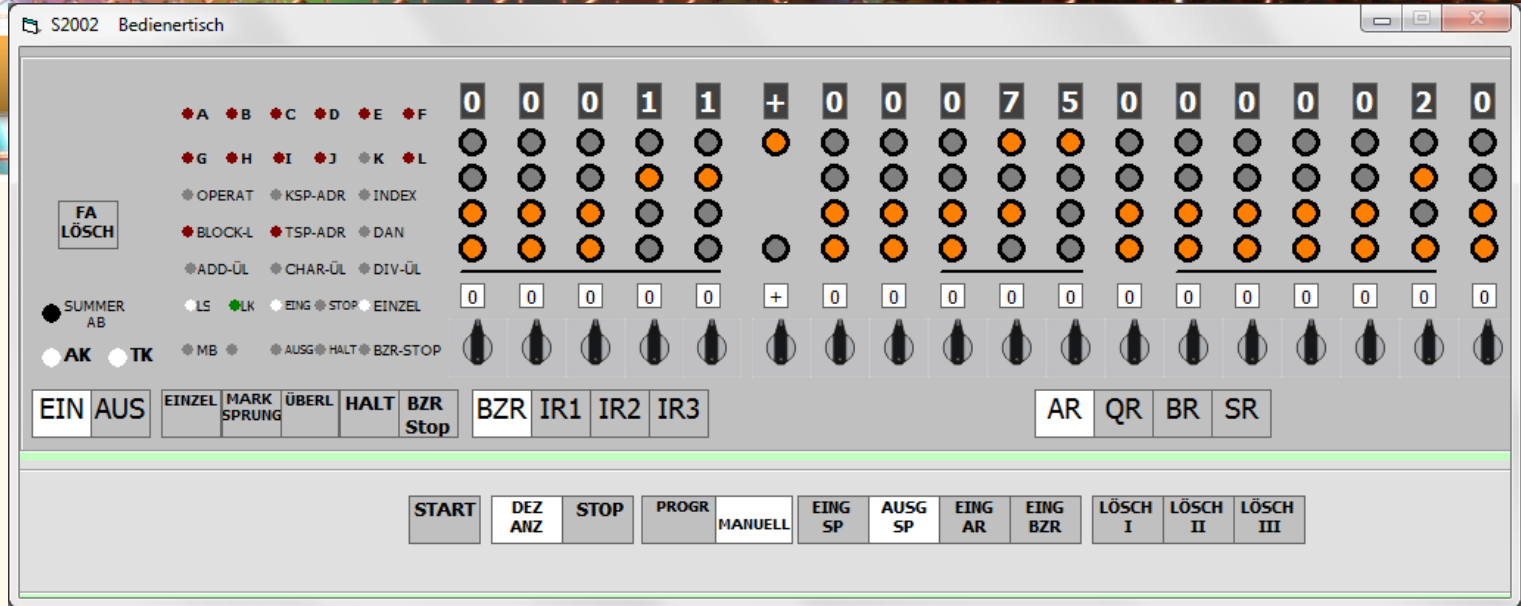
Während des Einlesens laufen die Adressen und Inhalte durch die Anzeige.

Zügig



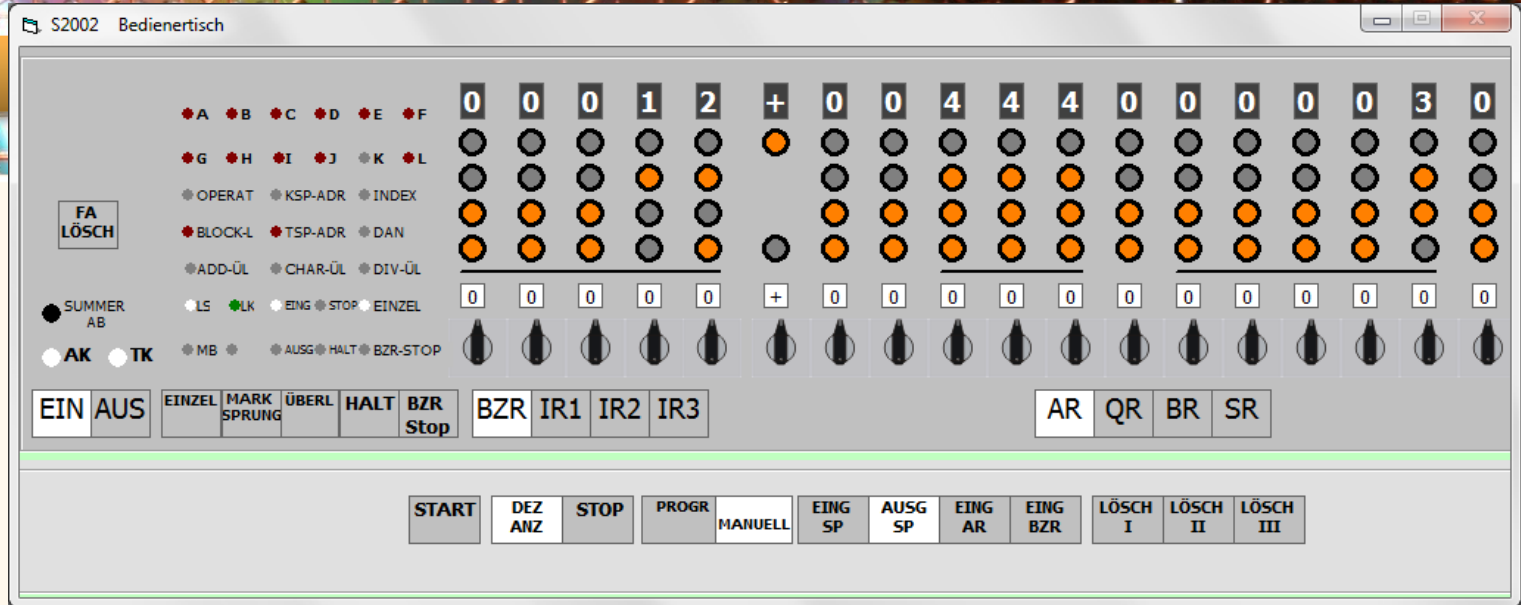
Während des Einlesens laufen die Adressen und Inhalte durch die Anzeige.

Zügig



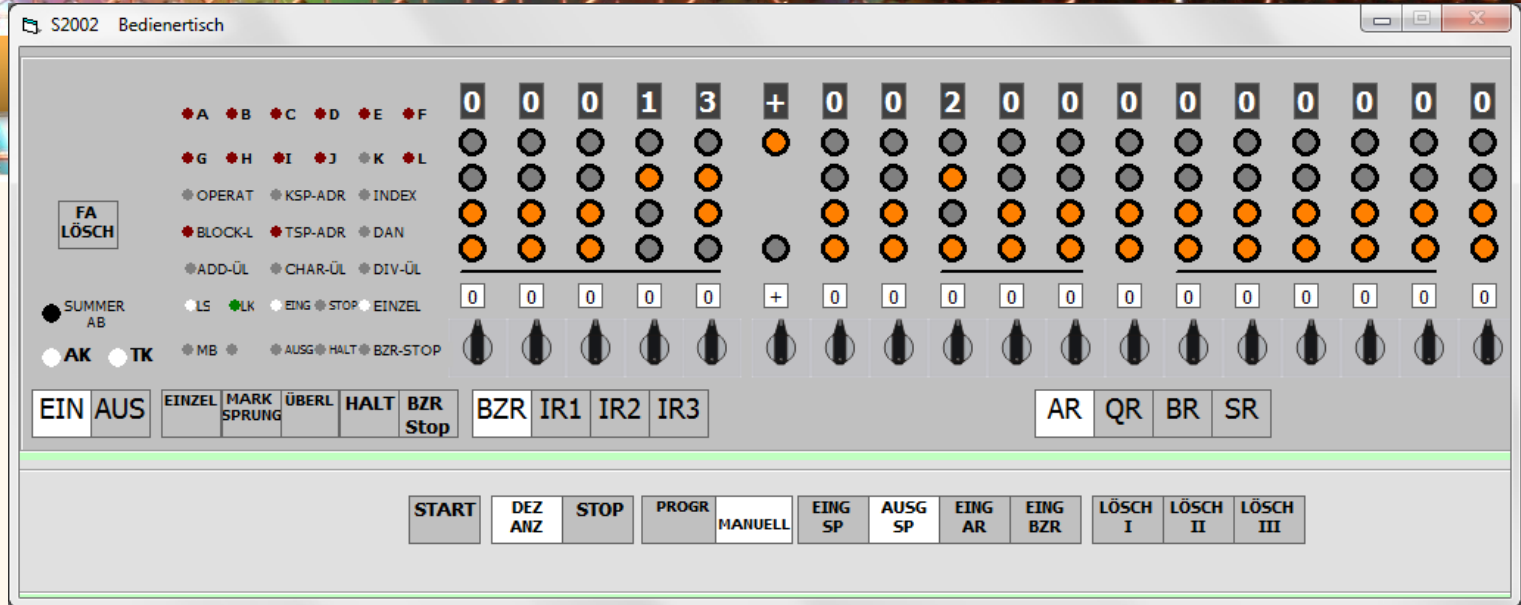
Während des Einlesens laufen die Adressen und Inhalte durch die Anzeige.

Zügig



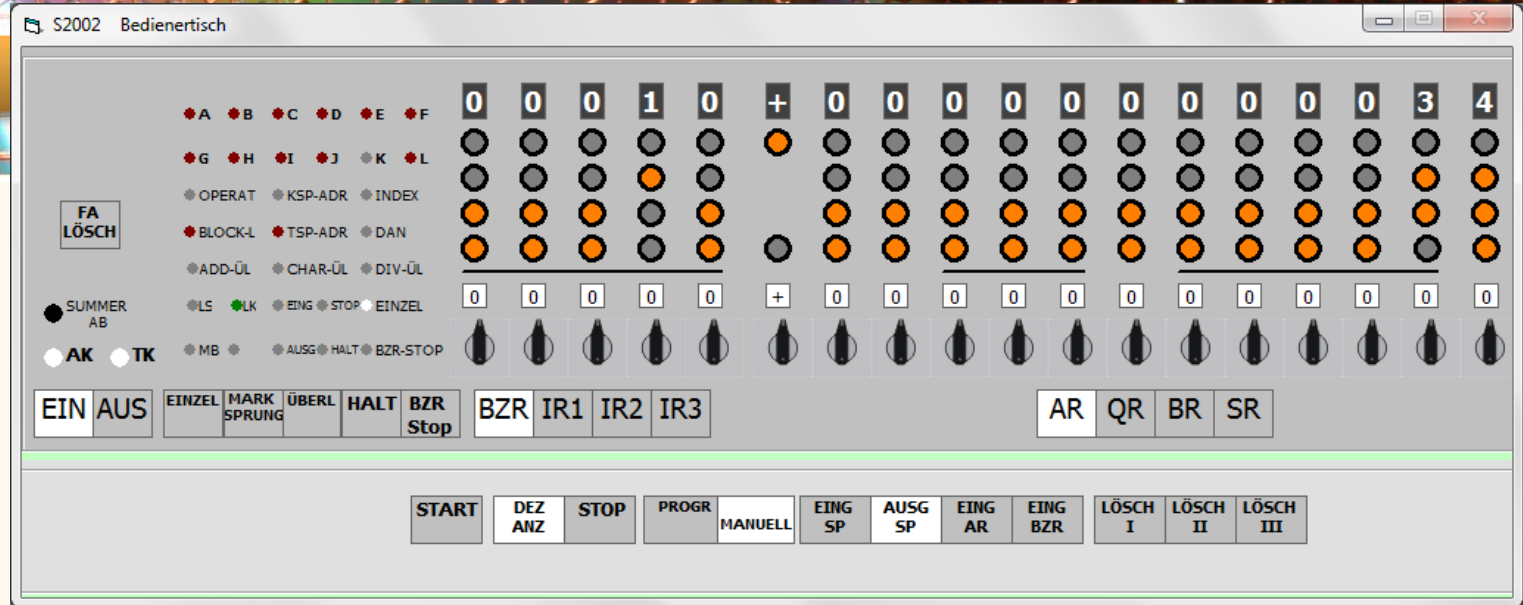
Während des Einlesens laufen die Adressen und Inhalte durch die Anzeige.

Zügig



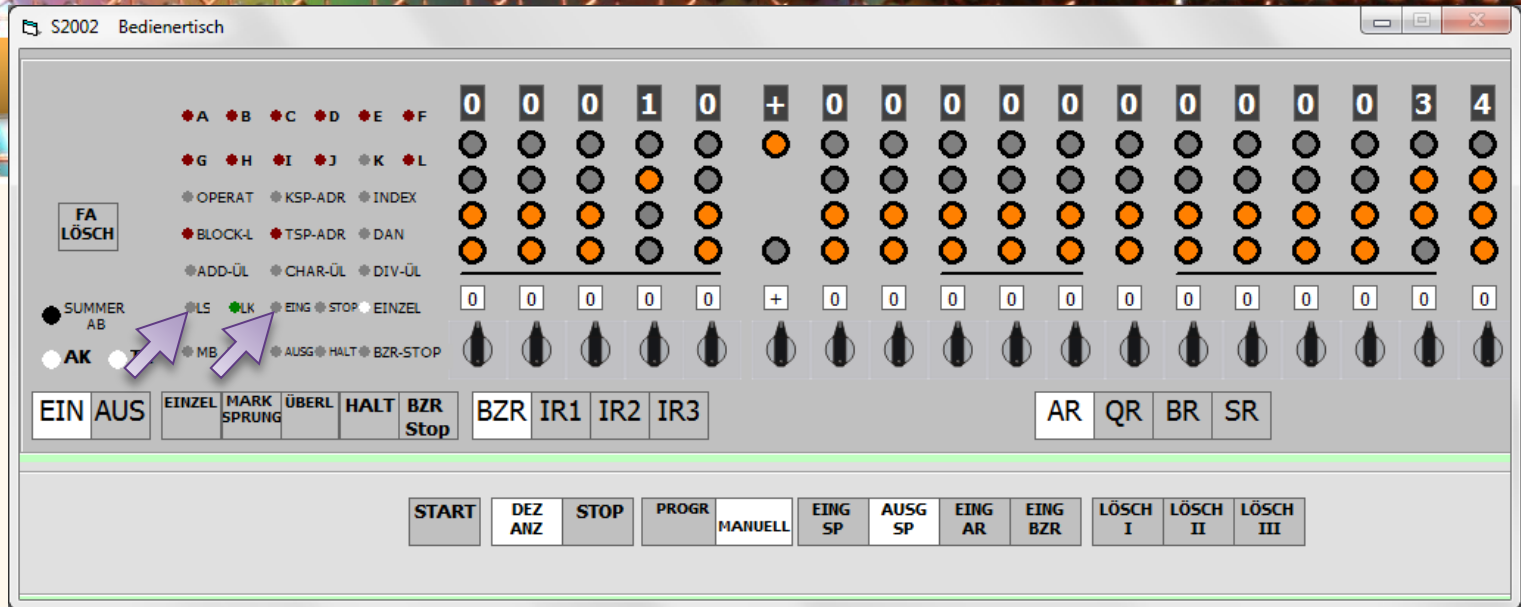
Während des Einlesens laufen die Adressen und Inhalte durch die Anzeige.

Zügig



Ist die Lochstreifen-Eingabe beendet,
erlöschen die Anzeige-Lämpchen **LS** und **EING**.

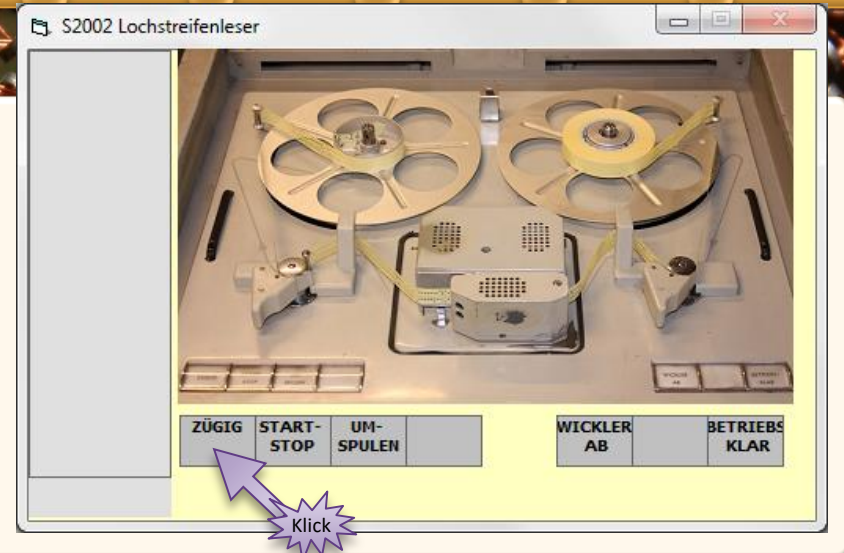
Zügig



Zügig-Eingabe am Lochstreifenleser ausschalten.

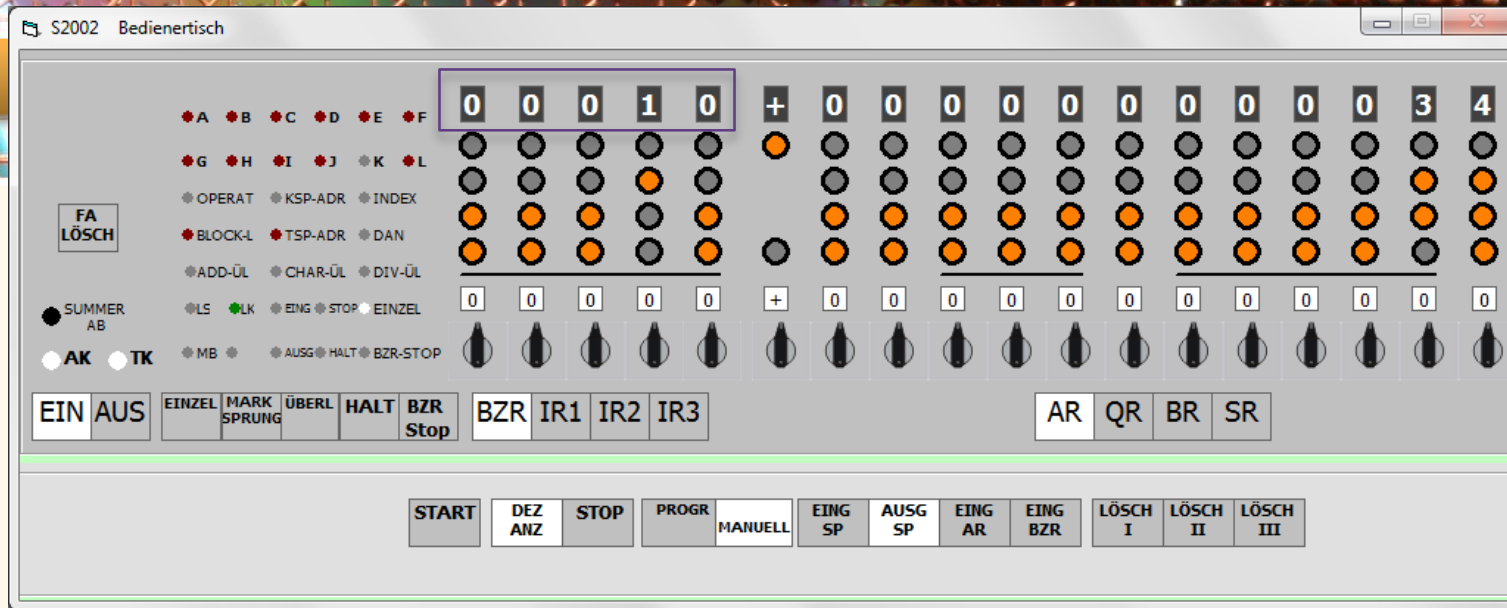
Zügig

Linksklick auf Schaltfläche **ZÜGIG**
legt die eingelesenen Daten im
Kernspeicher ab.



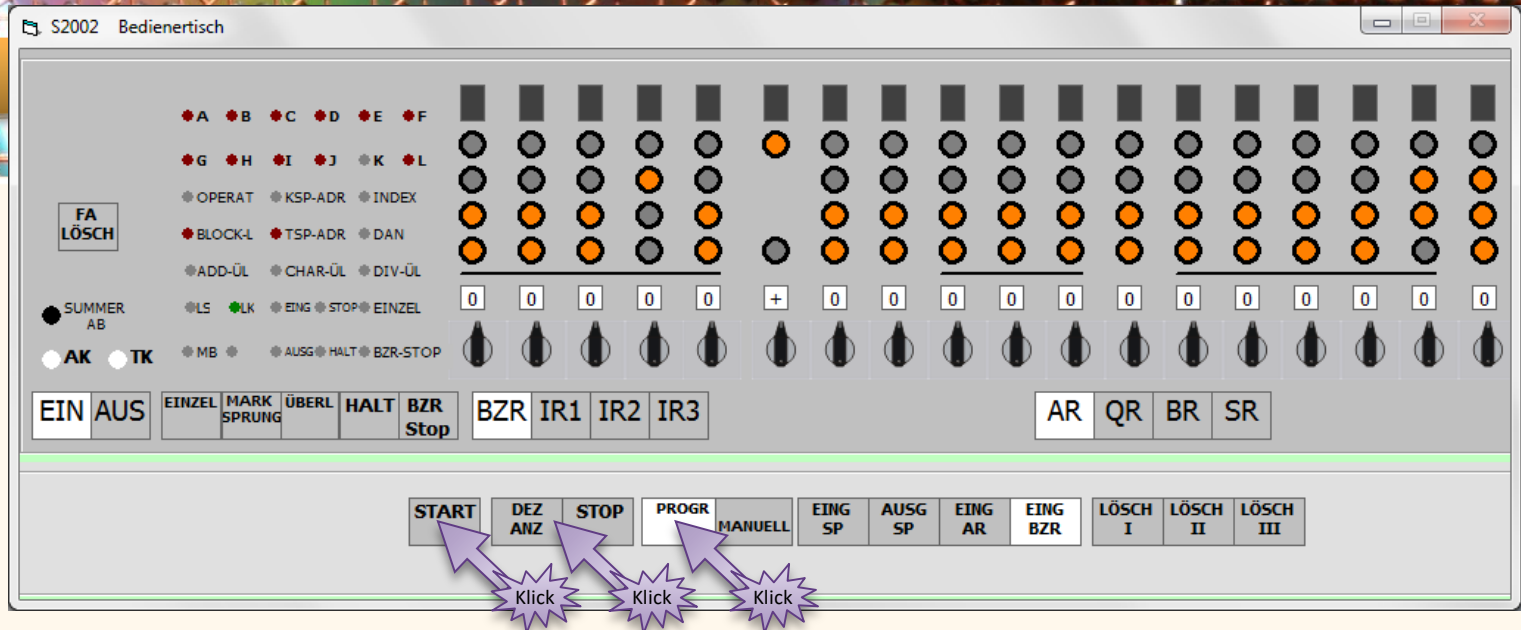
Das **BZR** ist durch den **Züigig-Code** schon auf die **Startadresse** eingestellt.

Züigig



Dezimal-Anzeige (**DEZ ANZ**) ausschalten,
Programm-Modus (**PROGR**) einschalten,
und mit **START** das Programm starten.

Zügig

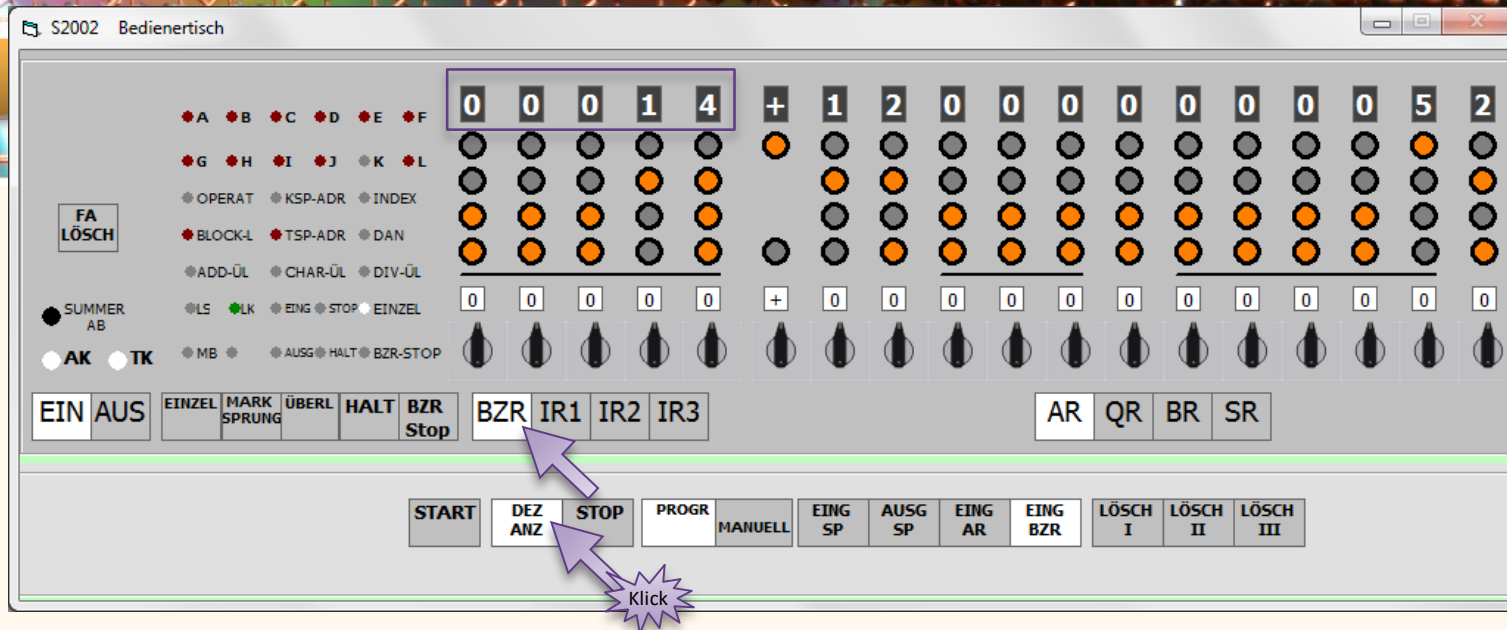


Die Anzeige (**STOP**) zeigt an, dass der Befehl **STP** abgearbeitet wurde.

The image shows a detailed view of the S2002 control panel. At the top, there's a title bar with the text 'S2002 Bedientisch'. Below this, the panel is divided into several sections. On the left, there are buttons for 'FA LÖSCH', 'SUMMER AB', 'AK', and 'TK'. In the center, there's a row of indicator lights labeled 'A' through 'L', each with a corresponding letter and a small red dot. Below these are buttons for 'OPERAT', 'KSP-ADR', 'INDEX', 'BLOCK-L', 'TSP-ADR', 'DAN', 'ADD-ÜL', 'CHAR-ÜL', 'DIV-ÜL', 'LS', 'LK', 'EING', 'STOP', 'EINZEL', 'MB', 'AUSG', 'HA', and 'BZR-STOP'. A mouse cursor is pointing at the 'STOP' button. To the right of these buttons are several digital displays showing '0' or '+'. Further right, there are buttons for 'EIN', 'AUS', 'EINZEL', 'MARK SPRUNG', 'ÜBERL', 'HALT', 'BZR Stop', 'BZR', 'IR1', 'IR2', 'IR3', 'AR', 'QR', 'BR', and 'SR'. At the bottom, there's a row of buttons for 'START', 'DEZ ANZ', 'STOP', 'PROGR', 'MANUELL', 'EING SP', 'AUSG SP', 'EING AR', 'EING BZR', 'LÖSCH I', 'LÖSCH II', and 'LÖSCH III'.

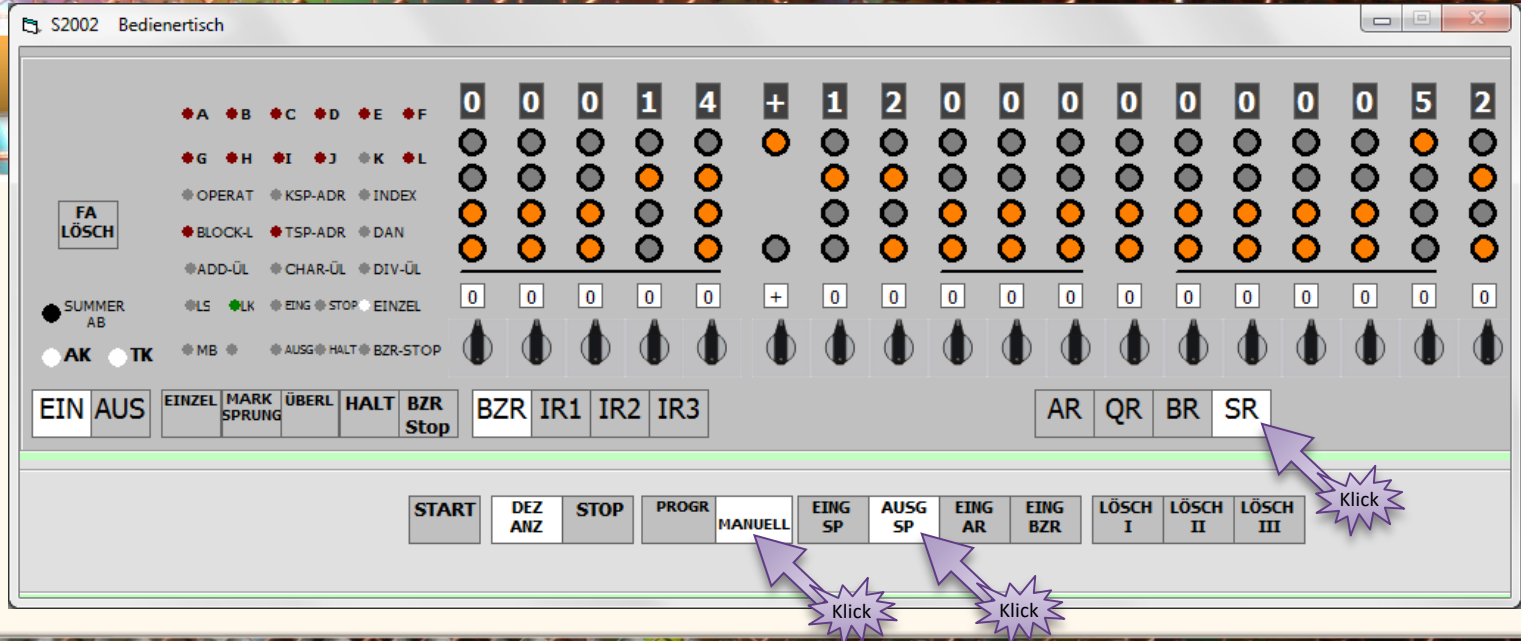
Dezimal-Anzeige (**DEZ ANZ**) für einfaches Ablesen einschalten.
Kontrolle: Die Anzeige des Befehlszählregisters (**BZR**) ist eingeschaltet.
Es steht auf Adresse 14, also sind alle Befehle abgearbeitet worden.

Zügig



Vorbereitung für die Anzeige des Ergebnisses:
Umschalten in den manuellen Modus (**MANUELL**).
Aktion Ausgabe Speicher (**AUSG SP**), Anzeige Speicher (**SR**) einschalten.

Zügig



In Speicherzelle C, also an Speicheradresse 3 sollte das Ergebnis stehen.

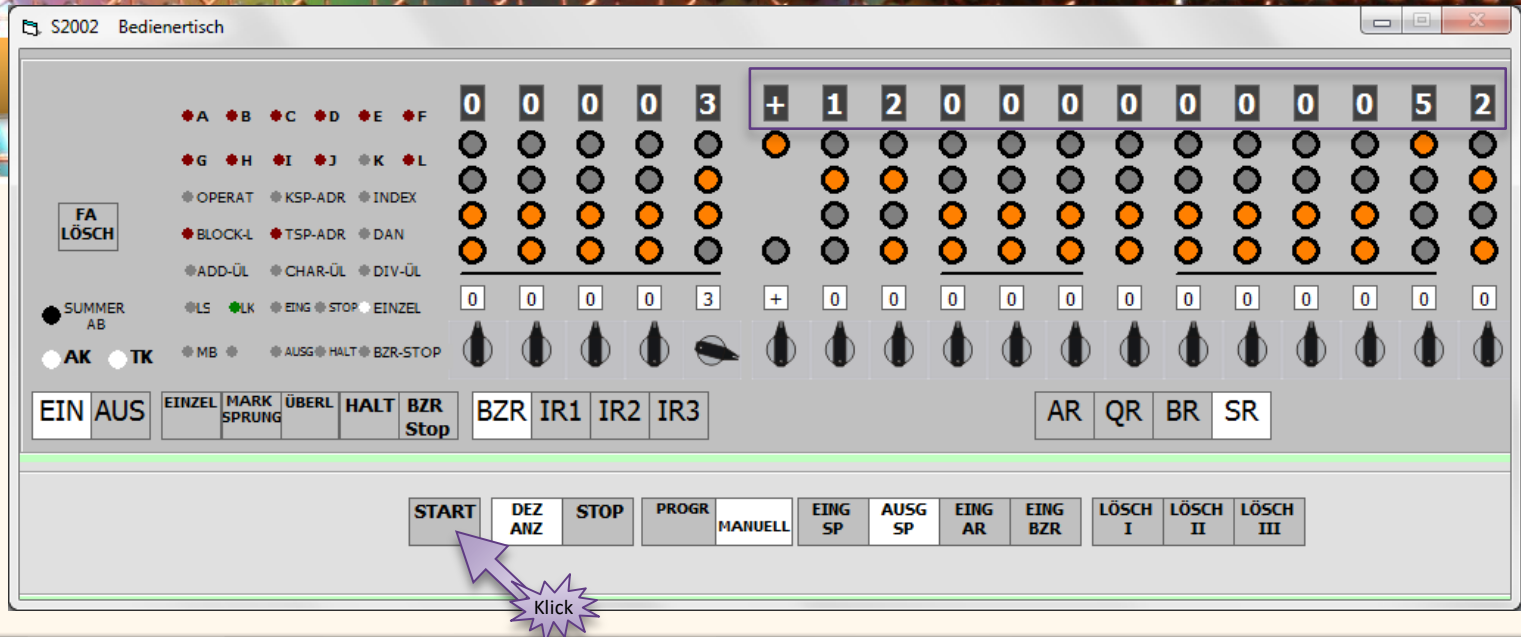
Adresse einstellen.

Zügig



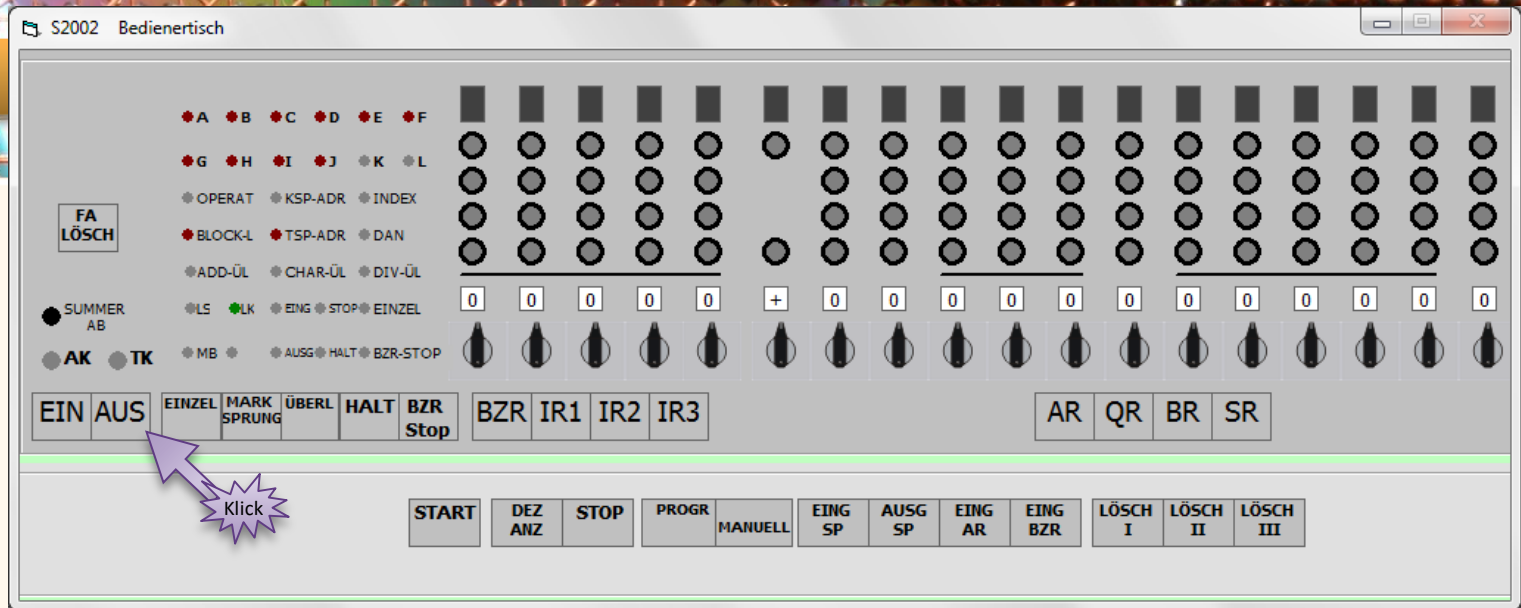
Mit **START** den Inhalt anzeigen – im Speicher steht 120..052 – richtig!
Das ist **0,12*10²**, also der Wert **12**.

Zügig



Die Rechenanlage kann wieder ausgeschaltet werden (**AUS**),
vorher sollten aber immer alle Drehschalter auf 0 gesetzt werden.

Zügig



SIEMENS 2002

Beispiel 3 Zählschleife

Für alle Beispiele wurde die **Emulation** der **Siemens 2002** auf dem PC unter Windows von **Volker Dahrendorf** benutzt.



Zählschleife

Beispiel 3 demonstriert die Implementation einer Zählschleife, einer häufig benutzten Steuerungsstruktur (control structure): Zähler auf **Anfangswert** setzen, in einer Schleife Zähler **inkrementieren** und die Schleife verlassen, wenn der vorgegebene **Endwert** erreicht ist.

In der Schleife stehen normalerweise wichtige Anweisungen, die z. B. bei jedem Durchlauf für ein anderes Element ausgeführt werden sollen. Hier im **Beispiel** soll einfach nur von 1 bis 10 gezählt werden, in Schritten von 1.

Beispiel 3:

Zähler auf **Anfangswert** setzen, in einer Schleife Zähler **inkrementieren**, und die Schleife verlassen, wenn der vorgegebene **Endwert** erreicht ist.

Zählschleife

Es gab auf der S2002 nur **Fest- und Gleitkomma-Arithmetik**, keine Ganzzahlen im Sinne des heute bekannten Datentyps Integer. Natürlich können Zahlen in Festkomma-Darstellung immer auch als ganze Zahlen interpretiert werden. Dennoch erfordert das Inkrementieren des Wertes in einer Speicherzelle immer aufwändige 3 Schritte:

Lade das AR aus einer Speicherzelle, **Addiere** das Inkrement zum AR, **Schreibe** den Inhalt des AR in die Speicherzelle.

Es gab auf der S2002 aber auch 3 **Indexregister**. Ihr Inhalt wurde als Adresse interpretiert und diente der Adress-Modifikation in vielen Befehlen.

Zählschleife

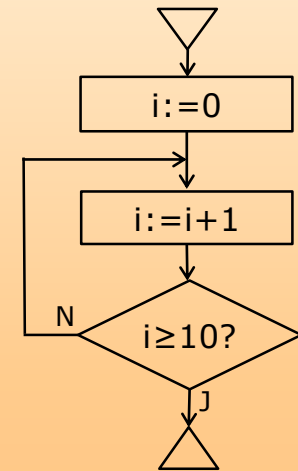
Auf diese Weise konnten, wie in der **Mathematik**, indizierte Datenmengen (Datenfelder wie **Vektoren** und **Matrizen**) Element für Element angesprochen (adressiert) werden. Dabei wurden ein oder mehrere Indexregister inkrementiert. Die Einheit **1 Speicherwort=1 Datenelement** erforderte zudem keine weiteren Zusatzberechnungen bei der Verwendung der Indexregister in Zählschleifen.

Beispiel 3: Zählschleife (ohne weitere Anweisungen) mit Indexregister

Zählschleife

Assemblercode:

LIA	0/1	Lade das IR1 mit dem Wert 0 (eigentlich: Adresse 0)	
S:	ADI	1/1	Addiere 1 zum IR1
IGR	10/1	Wert (Adresse) in IR1 größer oder gleich 10?	
SPR	S	Springe nach S	
STP		Halte an – fertig	



Wenn nicht, wird der **nächste** Befehl ausgeführt, **sonst** der **übernächste**

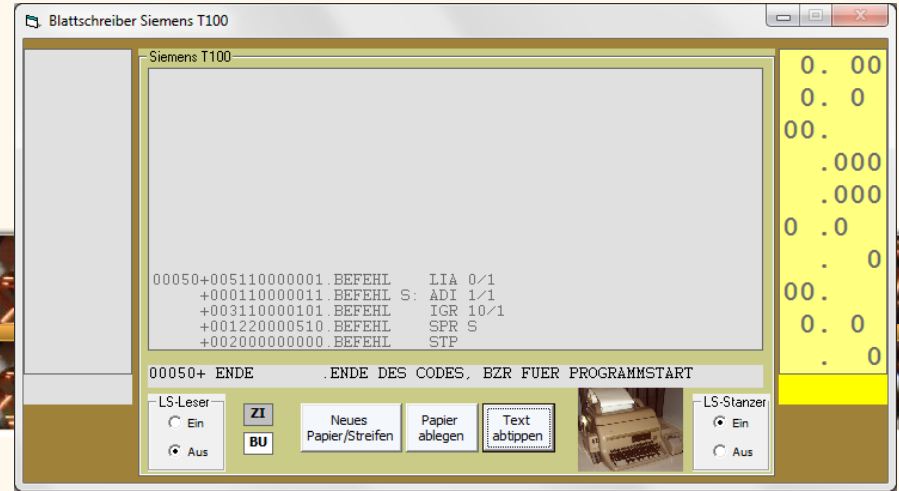
Die Assembler-Befehle müssen in **Maschinencode** übersetzt werden und für den **symbolische** Namen der Sprungmarke S muss eine **absolute** Speicheradresse festgelegt werden.

Zählschleife

Adresse	Inhalt	Erläuterung		
00050	005110000001	Befehl	LIA	0/1
00051	000110000011	Befehl	S: ADI	1/1
00052	003110000101	Befehl	IGR	10/1
00053	001220000510	Befehl	SPR	S
00054	002000000000	Befehl	STP	

Am Fernschreiber wird der
Züig-Lochstreifen erstellt.

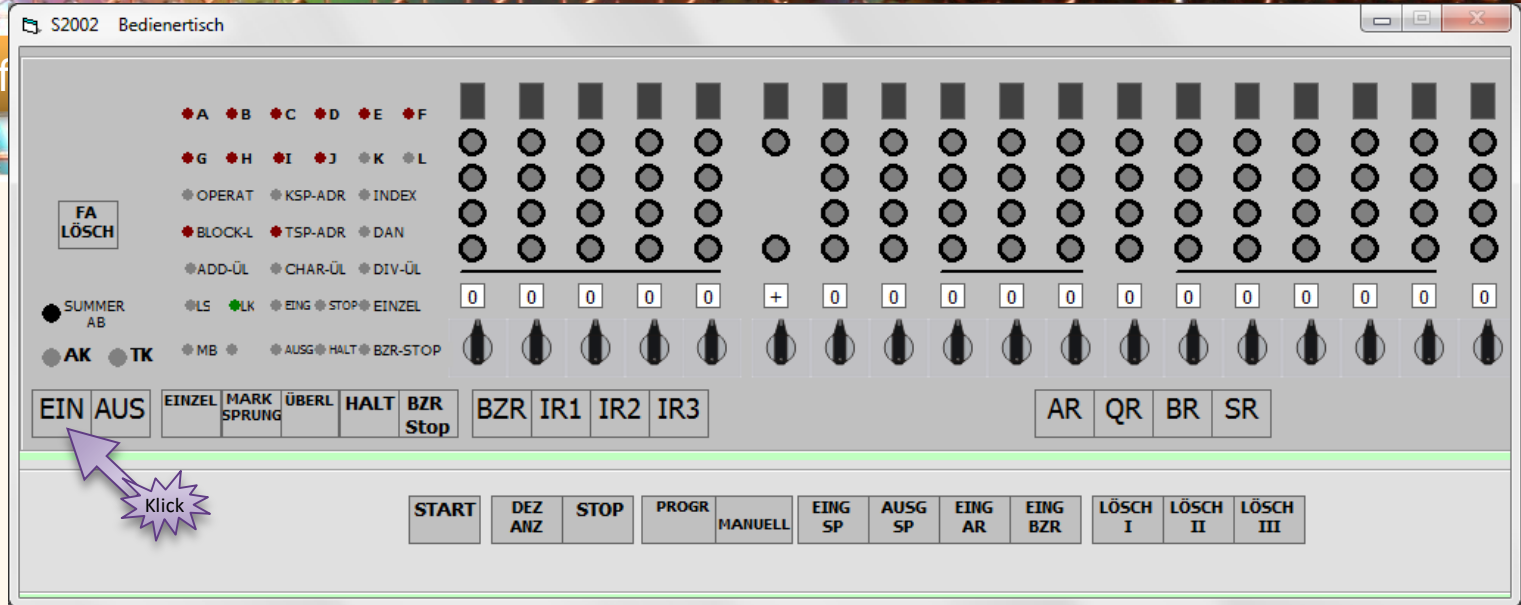
Zählschleife



```
00050+005110000001.BEFEHL    LIA 0/1
      +000110000011.BEFEHL S: ADI 1/1
      +003110000101.BEFEHL    IGR 10/1
      +001220000510.BEFEHL    SPR S
      +002000000000.BEFEHL    STP
00050+ ENDE      .ENDE DES CODES, BZR FUER PROGRAMMSTART
```

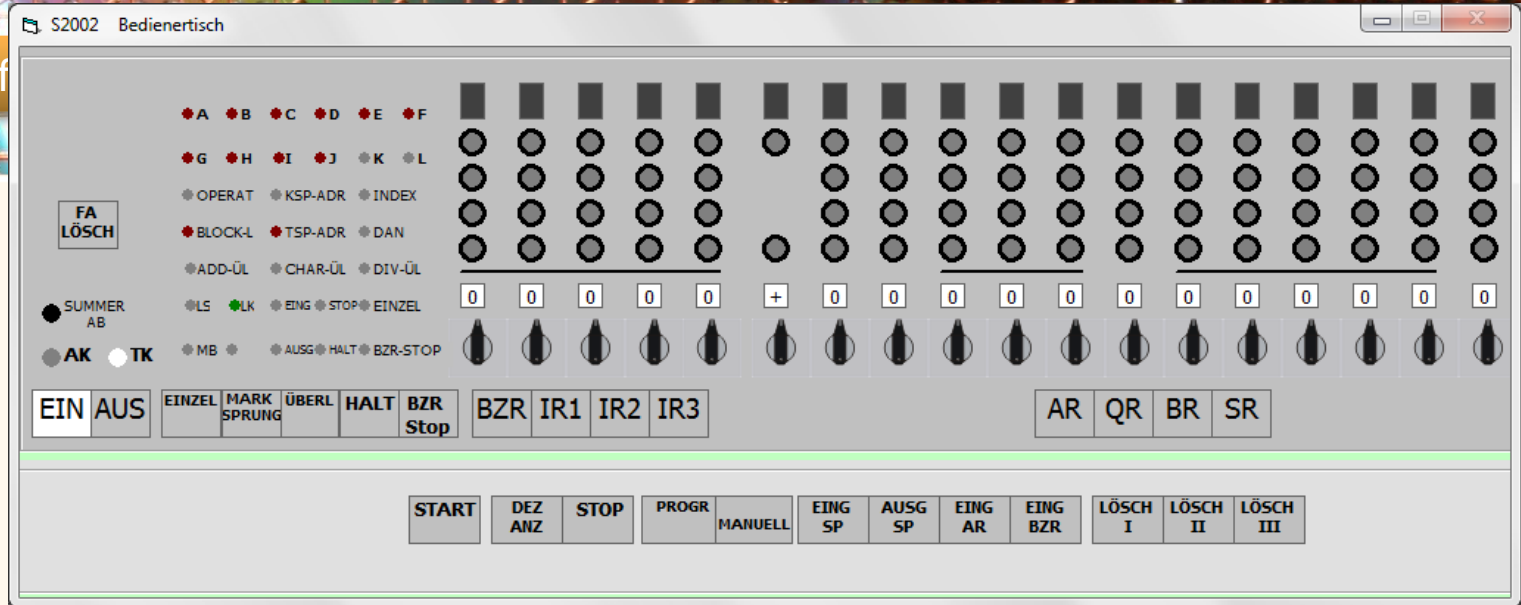
Rechenanlage einschalten (**EIN**), warten bis die Anlage betriebsbereit ist.

Zählschleife



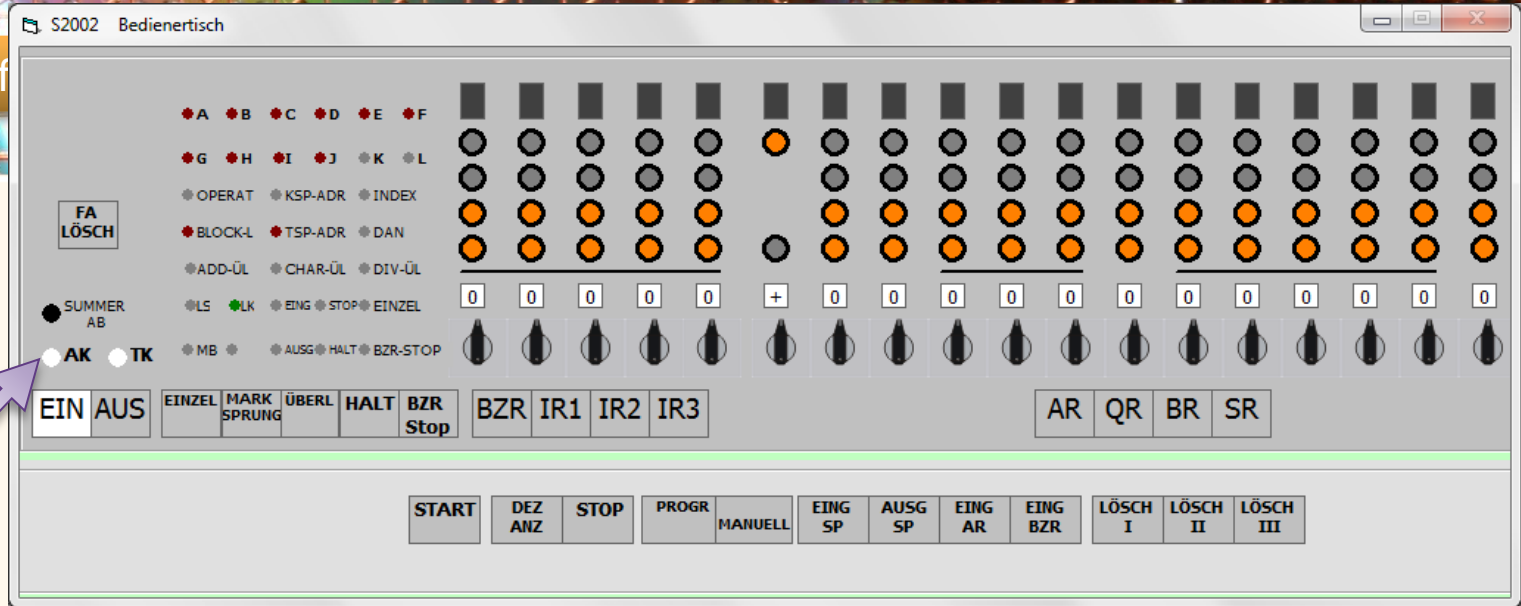
Rechenanlage einschalten (**EIN**), warten bis die Anlage betriebsbereit ist.

Zählschleife



Rechenanlage einschalten (**EIN**), warten bis die Anlage betriebsbereit ist.

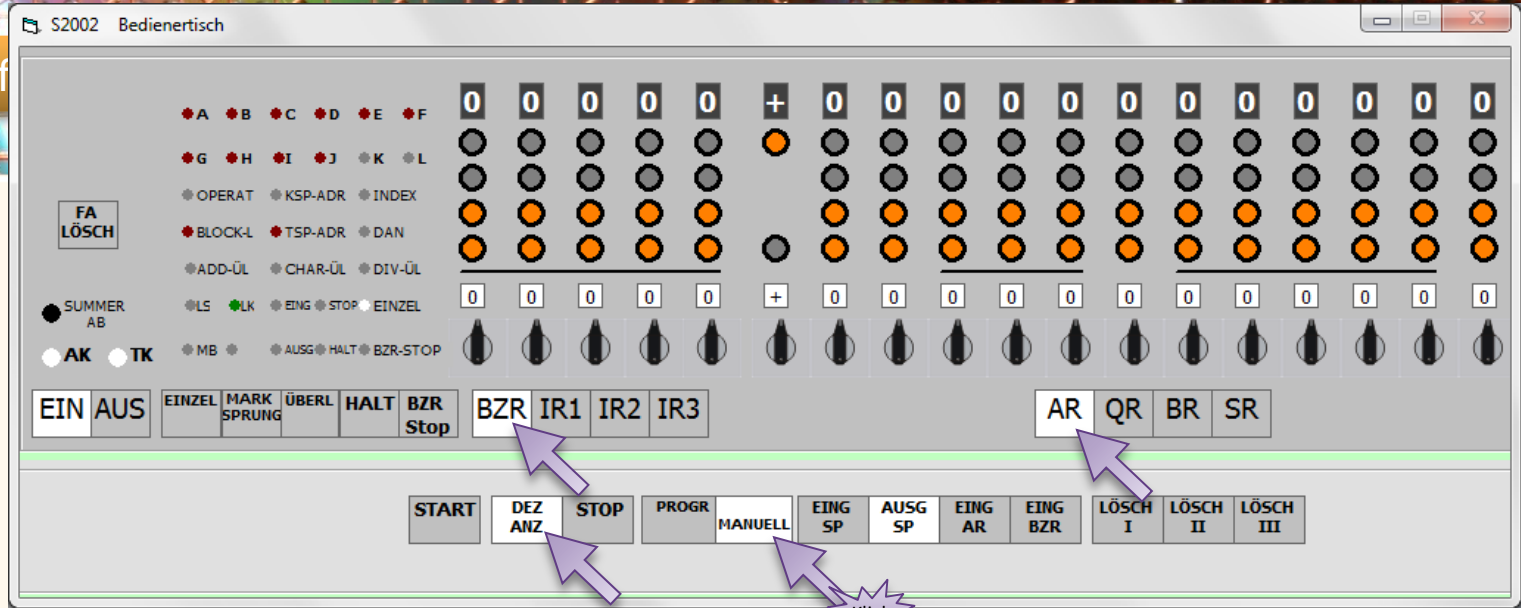
Zählschleife



Manuellen Modus (**MANUELL**) einschalten.

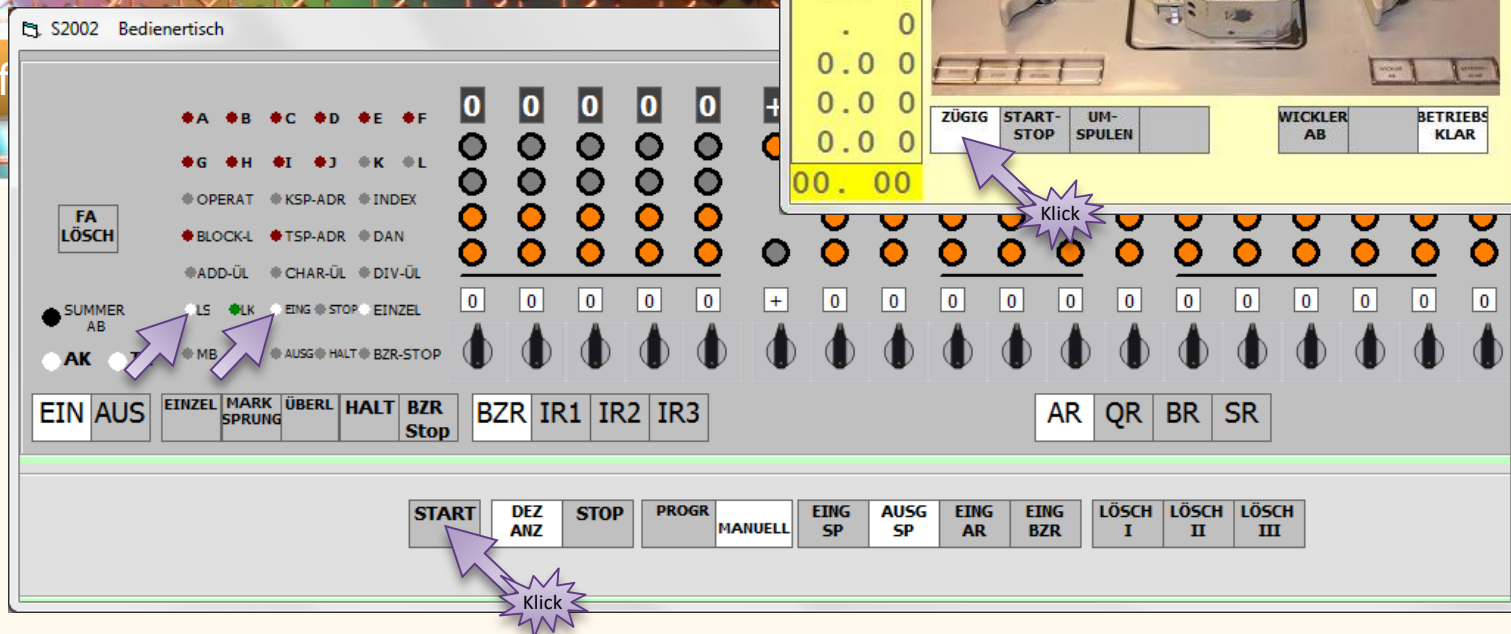
Anzeige von **BZR** und **AR** sind praktisch, dann laufen während des Einlesens die Adressen und Inhalte durch die Anzeige. **DEZ ANZ** hilft beim Ablesen.

Zählschleife



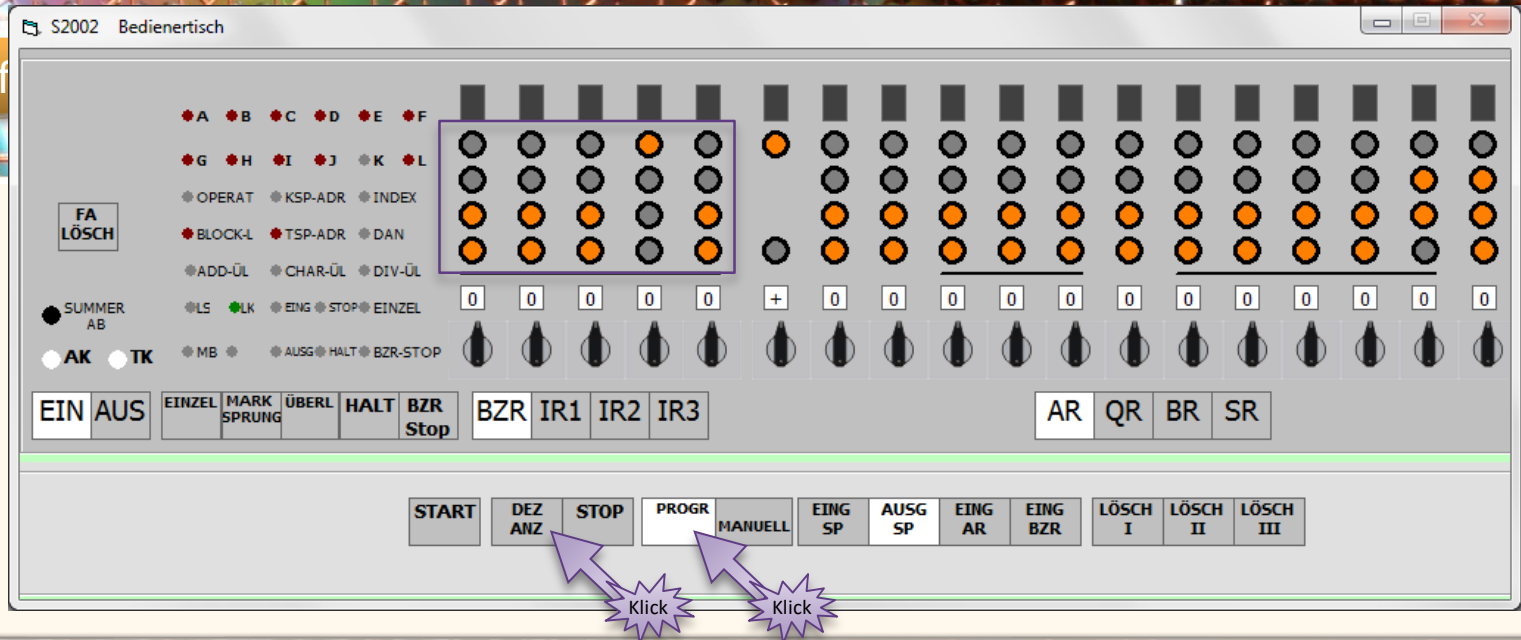
Lochstreifen im **Zügig-Modus** einlesen.

Zählschleife



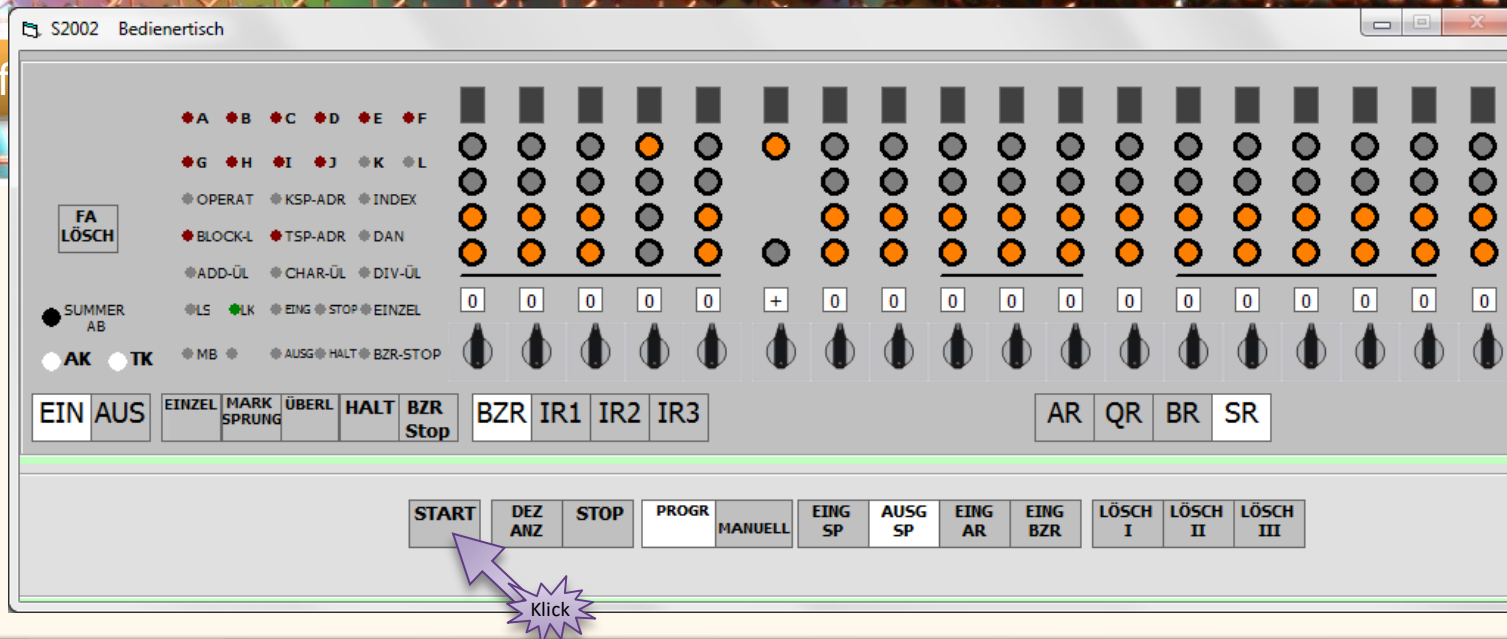
Der Züig-Code hat das **BZR** auf die **Startadresse** eingestellt.
DEZ ANZ ausschalten, weil das Programm sonst nicht in einem Zug abläuft,
sondern im Einzelschrittmodus.
In den Programm-Modus (**PROGR**) umschalten.

Zählschleife



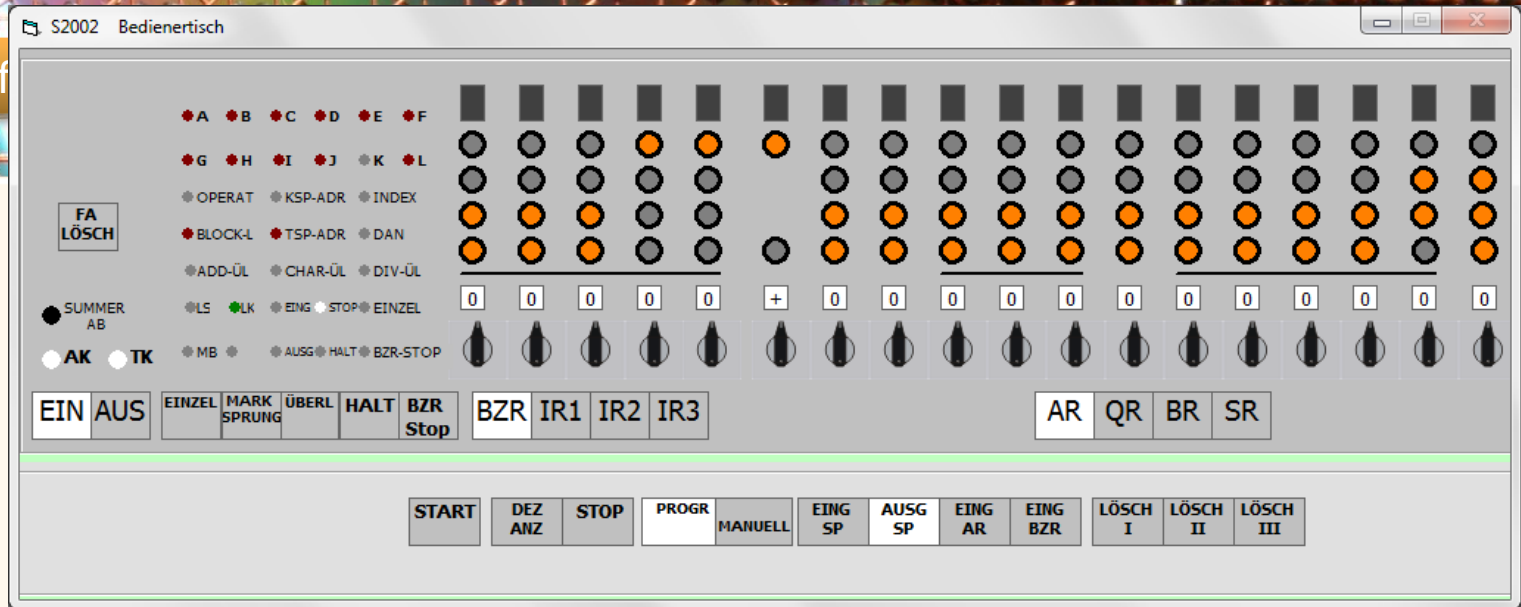
Das Programm starten (**START**).

Zählschleife



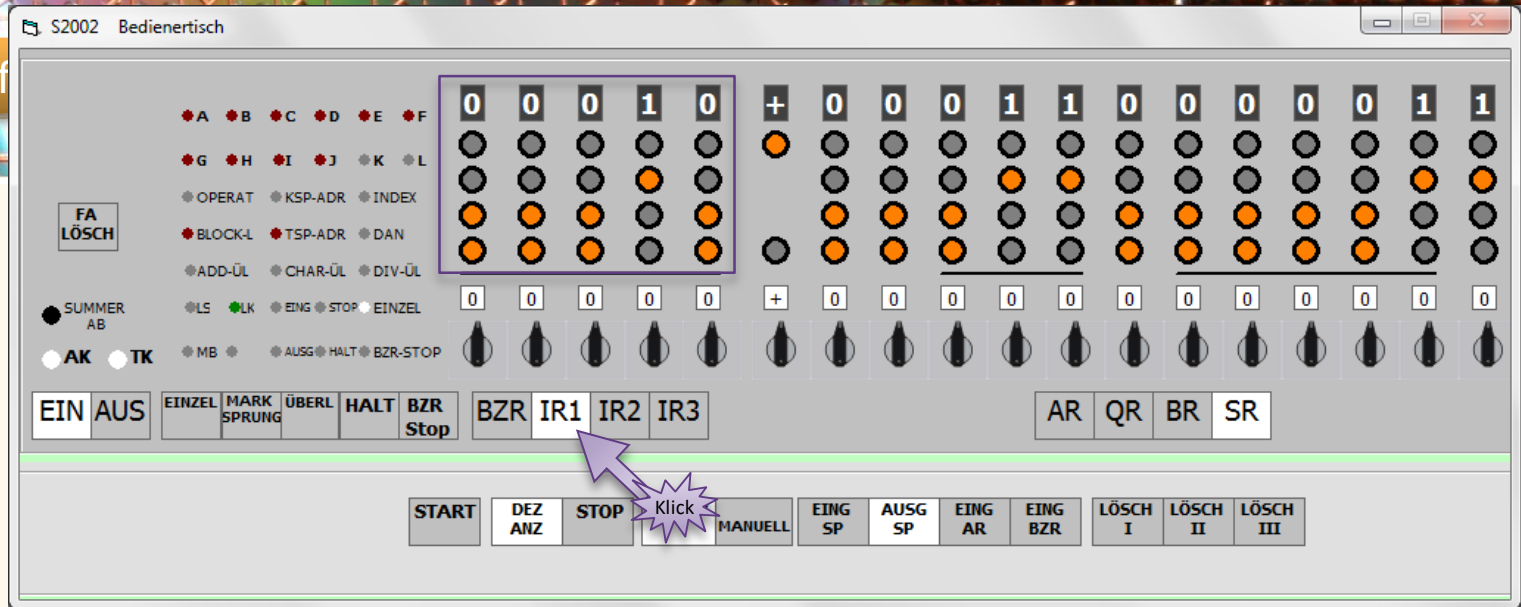
Ist während des Ablaufs die **Anzeige** des **BZR** eingeschaltet, ist die Schleife an der Anzeige deutlich zu erkennen.

Zählschleife



Ist das Programm fertig, kann die Anzeige des **IR1** eingeschaltet werden.
Kontrolle: Richtig, der Inhalt ist jetzt **10**.

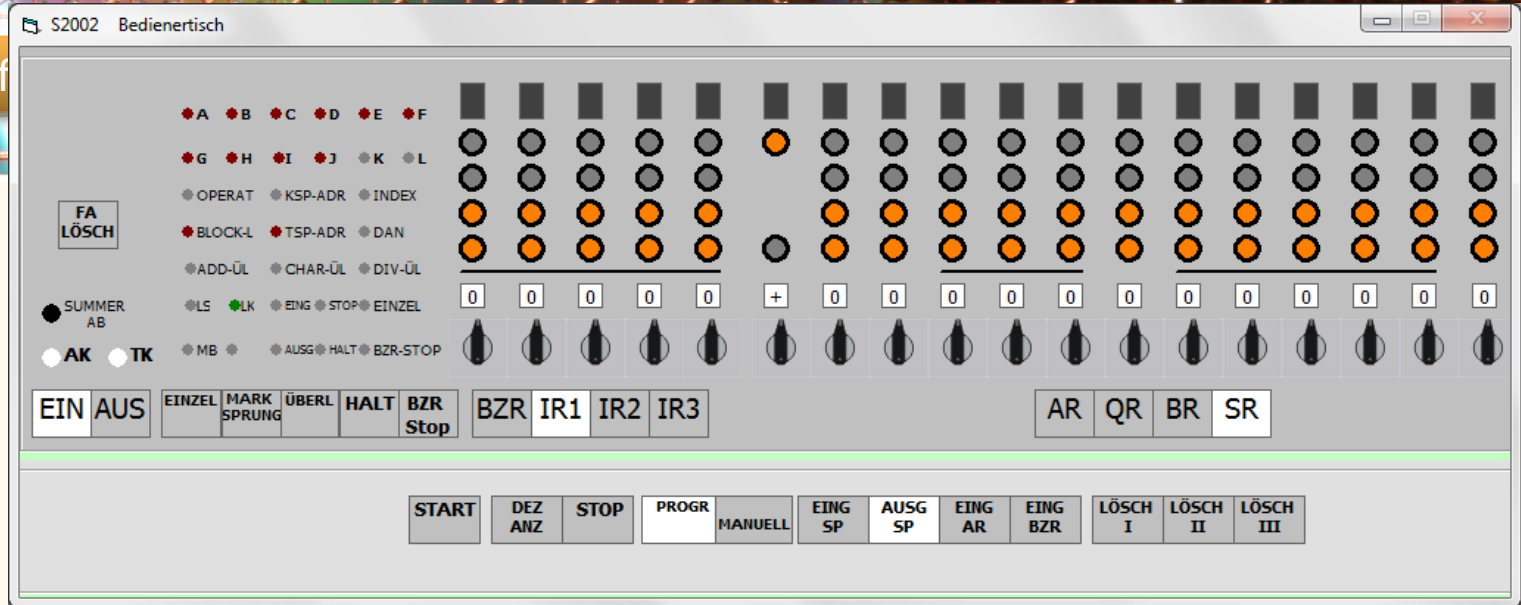
Zählschleife



Ist während des Ablaufs die **Anzeige** des **IR1** eingeschaltet, kann das Hochzählen beobachtet werden.

Zählschleife

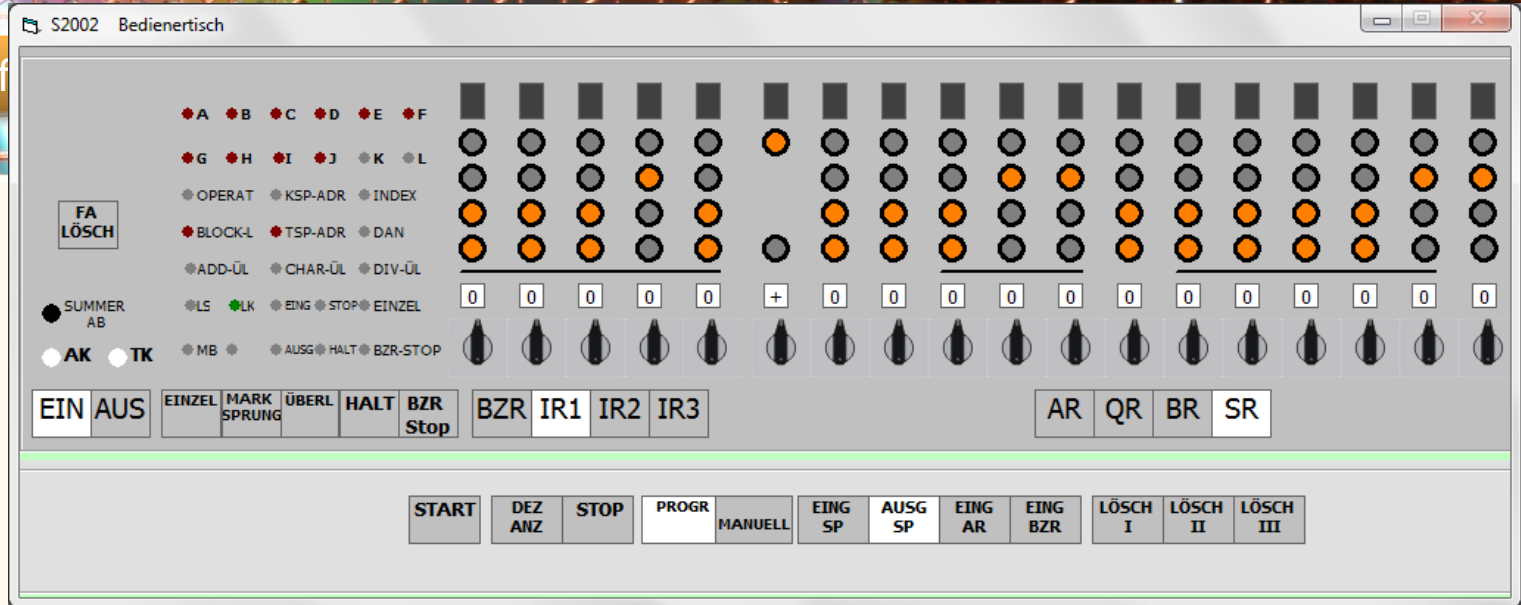
Hier werden nur die Zähl-Schritte (-befehle) angezeigt.



Ist während des Ablaufs die **Anzeige** des **IR1** eingeschaltet, kann das Hochzählen beobachtet werden.

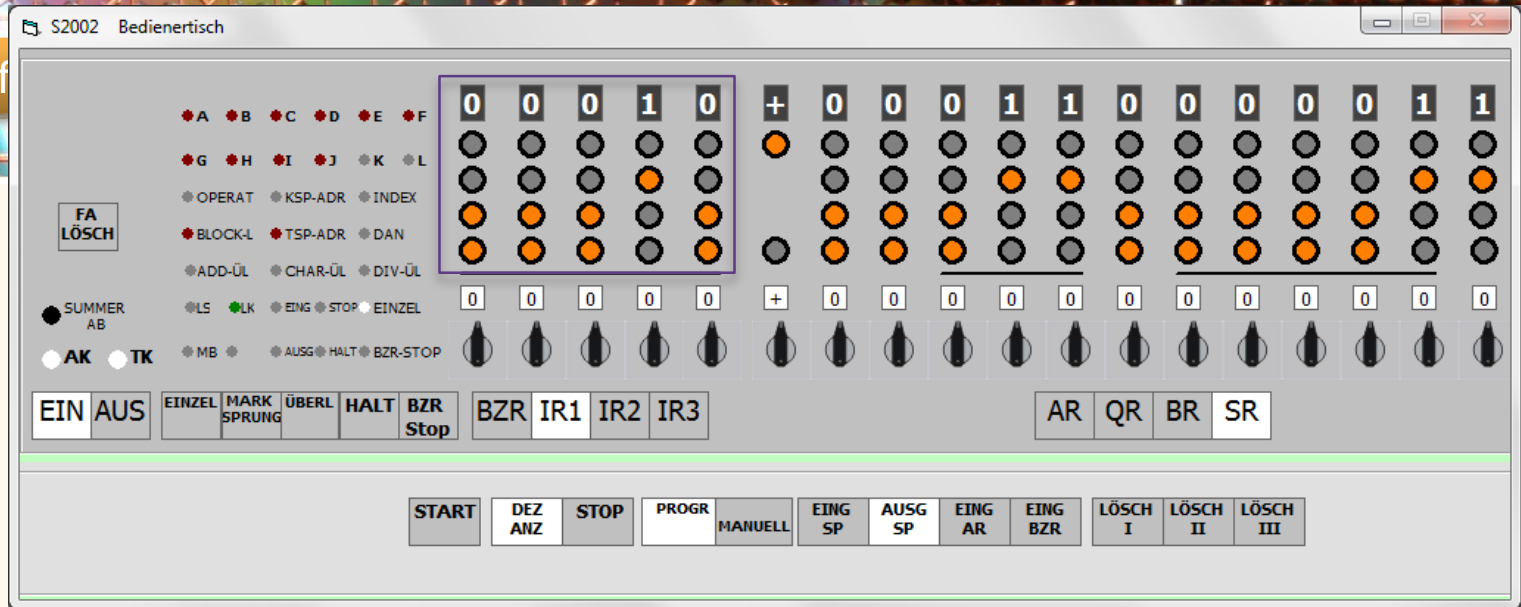
Zählschleife

Hier werden nur die Zähl-Schritte (-befehle) angezeigt.



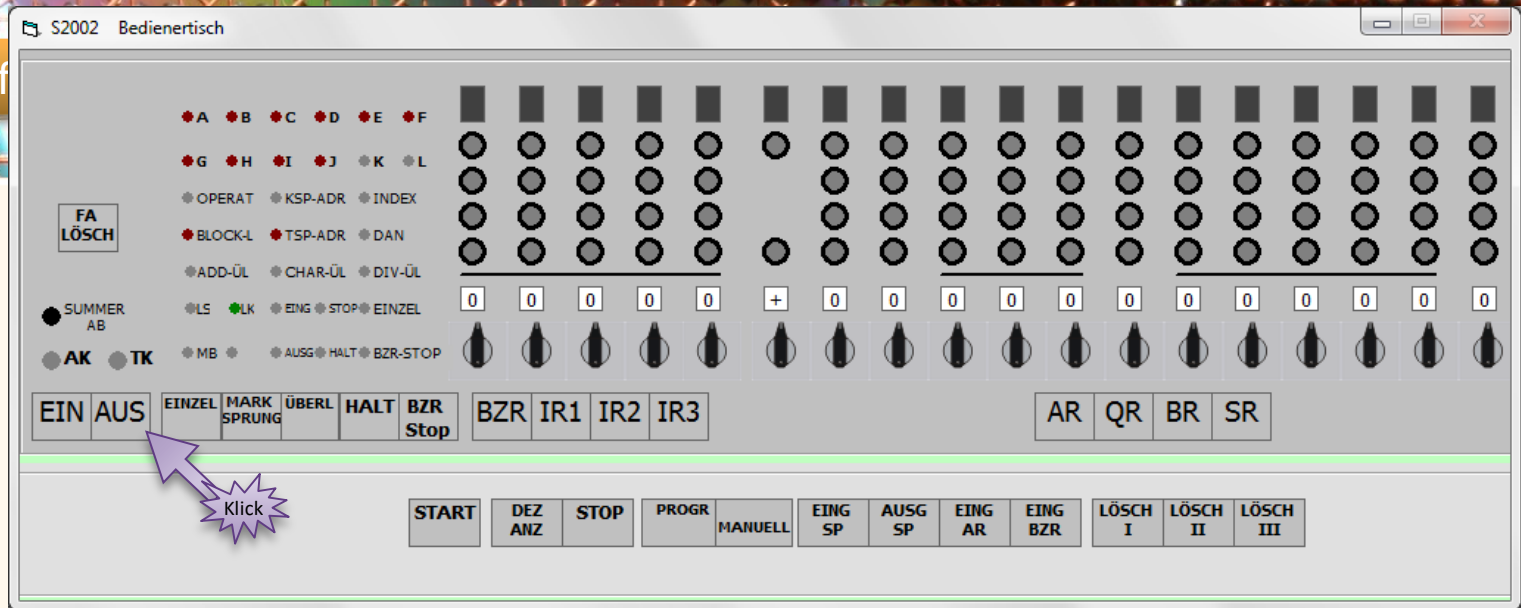
Ist das Programm fertig, kann die Anzeige des **IR1** abgelesen werden:
Kontrolle: Richtig, der Inhalt ist jetzt **10**.

Zählschleife



Die Rechenanlage kann wieder ausgeschaltet werden (**AUS**),
vorher sollten aber immer alle Drehschalter auf 0 gesetzt werden.

Zählschleife





**ENDE DIESER
DOKUMENTATION
VIELEN DANK FÜR IHRE
AUFMERKSAMKEIT**