Der NDR-Klein-Computer

KURZ-

HANDBUCH

zum 6502 Grundprogramm

Version 05/20

2020 Jens Mewes

NKC

Die Legende lebt

Inhalt

1. Einführung 3
1.1. Vorwort 3
1.2. Links 3
2. Inbetriebnahme 4
2.1. Hardware-Voraussetzungen 4
2.2. Voreinstellungen 4
2.3. Software-Voraussetzungen 4
2.4. Empfohlene Erweiterungen 4
2.5. Speicheraufbau 5
2.6. Zeropage 5
2.7. System Flags 6
2.8. System RAM 6
3. Grundprogramm - Menüs 7
3.1. Feste Menüpunkte 7
3.1.1. A = MEM aendern 7
3.1.2. D = MEM anzeigen 7
3.1.3. F = MEM fuellen 7
3.1.4. I = I/O lesen 7
3.1.5. O = I/O setzen 7
3.1.6. K = Krusader starten 7
3.1.7. W = WOZ Monitor 7
3.1.8. R = Programm starten 7
3.2. Dynamische Menüpunkte 7
3.2.1. U = Uhr stellen 7
3.2.2. C = Laufwerk waehlen 7
3.2.1. L = Datei laden 7
3.2.1. S = Datei speichern 7
3.2.1. X = EXE ausführen 7
3.2.1. V = Inhalt 7
3.2.1. E = Stick auswerfen 7
4. Die Unterprogramme 8
4.1. Die Sprungtabelle 8
4.2. Beschreibung der Unterprogramme 10

1. Einführung

1.1. Vorwort

Moin Moin,
seit Rene und ich im letzten Jahr angefangen haben eine 6502 CPU-Karte zu entwickeln, stellte sich auch die Frage nach der Software. Nun ein NKC ist mit 6502-CPU ist ja ein besserer Apple1 Computer, was lag da näher als an dessen Software zu partizipieren. Daher habe ich den WOZ-Monitor (von Steve Wozniak) und den Krusader von Ken Wessen in das Grundprogramm integriert.
Das Grundprogramm stellt darüber hinaus einige grundlegende Funktionen zur Verfügung. Dazu gehören speichern und laden von Programmen/Daten von der CAS-Neo und der IO-USB, Speicher ändern, ansehen oder füllen, IO-Ports lesen und schreiben und die Uhrzeit stellen.
Desweiteren gibt ein eine Menge (hoffentlich) nützlicher Unterprogramme, die über eine Sprungtabelle für eigene Programme erreichbar sind.

In diesem Sinne: **NKC - Die Legende lebt.**

Jens Mewes aka. DerInder Hamburg, Mai 2020

1.2. Links

Infos, Anleitungen, Programme und Foren zum NKC:

http://nkc-schuetz.de/
http://www.ndr-nkc.de
http://www.nkc-wiki.de
https://www.robotrontechnik.de/html/forum/thwb/index.php

Source und Handbuch zum Krusader von Ken Wessen:

https://github.com/st3fan/krusader

Infos und Anleitung zum WOZ Monitor von Steve Wozniak:

https://www.sbprojects.net/projects/apple1/wozmon.php

2. Inbetriebnahme

2.1. Hardware-Voraussetzungen

Für den Betrieb des Grundprogramms werden folgende Hardwarekomponenten benötigt:

- BUS
- CPU6502
- ROA64
- GDP64
- KEY

Alternativ zur KEY und der GDP64 kann auch die GDP64-FPGA verwendet werden.

Mehr Baugruppen werden nicht für den Betrieb des Grundprogramms benötigt, obwohl mit dieser Minimalausstattung natürlich nicht sehr viel gemacht werden kann. Eine Übersicht von Erweiterungen erfolgt in Kapitel 2.4.

2.2. Voreinstellungen

Auf der CPU6502 muss die IO-Adresse auf $C0 eingestellt werden.

2.3. Software-Voraussetzungen

Es ist keiner weitere Software für den Betrieb des Grundprogramms nötig, da das Grundprogramm selbst die Grundlage aller anderen Programme bildet.

2.4. Empfohlene Erweiterungen

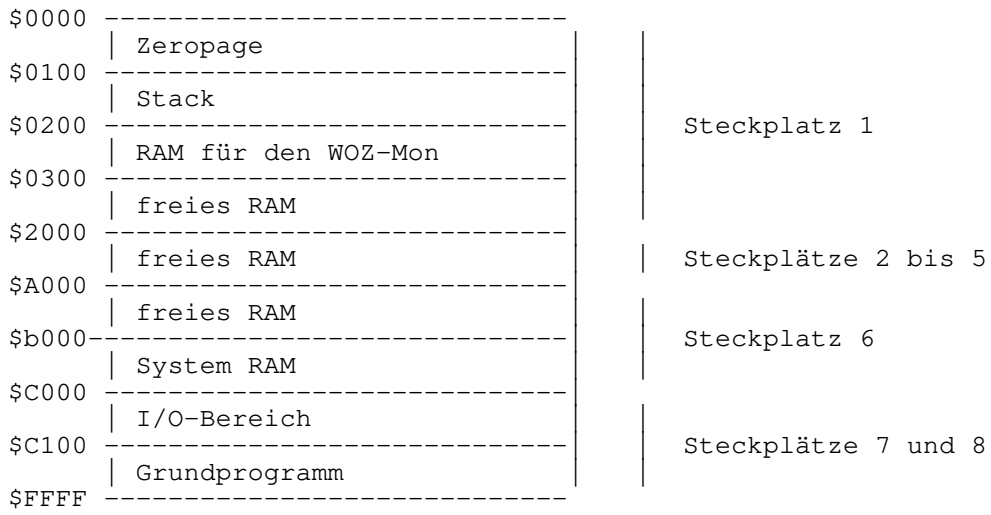
Für den NKC sind natürlich eine Menge von Baugruppen vorhanden. Hier erfolgt nun eine Aufstellung aller Baugruppen die direkt vom Grundprogramm unterstützt werden.

UHR3	-	Echtzeituhr des NKCs
CAS-NEO	-	Wird im FAT-Modus als Massenspeicher genutzt
SOUND(2)	-	Soundgenerator
SER2	-	Serielle Schnittstelle
IO-USB	-	Anschluss für USB-Stick

Die allermeisten NKC-Erweiterungskarte laufen auch mit der CPU6502, eine Ausnahme bilden hier allerdings die SER-Karte, die DRAM128(256) und die COL256. Letztere benötigen ein WAIT-Signal, das der 6502 Prozessor nicht kennt.

2.5. Speicheraufbau

Das GP6502 kommt in 2 (E)EPROMs, diese kommen auf die Steckplätze 7 und 8 der ROA64-Karte. Auf dem 1. und 6. Steckplatz muss RAM sein.



2.6. Zeropage

Die Belegung der Zeropage durch das Grundprogramm sieht folgendermaßen aus:

$00 - $1F	Grundprogramm -register/speicher
$20 - $7f	frei
$80 - $ff	Krusader und WOZ-Mon

Der Bereich $00 bis $1f sieht im Einzelnen so aus:

$00	**$01**	**$02**	**$03**	**$04**	**$05**	**$06**	**$07**
R0	R0H	R1	R1H	R2	R3H	R3	R3H
$08	**$09**	**$0A**	**$0B**	**$0C**	**$0D**	**$0E**	**$0F**
R4	R4H	R5	R5H	R6	R6H	R7	R7H
		$1A	**$1B**	**$1C**	**$1D**	**$1E**	**$1F**
		SYSFLAG0	SYSFLAG1	T_COUNT0	T_COUNT1	T_COUNT2	T_COUNT3

Die nicht beschriebenen Adressen sind interne Speicher.

2.7. System Flags

Die Zeropage enthält zwei System Flag Bytes diese haben folgende Bedeutung:

```
SYSFLAG0
FGDP            = $01           ; GDP Flag
FGDPHS          = $02           ; GDPHS Flag
FGDPFPGA        = $04           ; GDPFPGA Flag
FSER            = $08           ; SER2 Flag
FUHR            = $10           ; UHR3 Flag
FCN             = $20           ; CAS-Neo Flag
FSND            = $40           ; Sound Flag
FBB             = $80           ; BankBoot Flag

SYSFLAG1
FVDIP           = $01           ; VDIP Flag
FSTICK          = $02           ; USB-Stick Flag

FUNMI           = $40           ; NMI-Flag User Program
FNMI            = $80           ; NMI-Flag
```

Diese Flags werden automatisch vom GP verwaltet.

2.8. System RAM

Das System RAM ist zweigeteilt. Die erste Hälfte von $B000 bis $B7FF wird zum Speichern diverser Werte genutzt. Für User interessante Adressen sind folgende:

```
UHR3
$B000           SEK             ; Sekunden
$B001           MINU            ; Minuten
$B002           STD             ; Stunden
$B003           WOTAG           ; Wochentag
$B004           TAG             ; Tag
$B005           MON             ; Monat
$B006           JAHR            ; Jahr
$B007           JAHRH           ; Jahrhundert

NMI Userprogramm
$B00B           NMI_VECL        ; Adresse Low
$B00C           NMI_VECH        ; Adresse High

Eingabepuffer
$B026 - $B076                   ; Puffer für max. 81 Zeichen

Ausgabepuffer
$B077 - $B7FF                   ; Puffer für 1920 Zeichen
```

Der Bereich $B800 - $BFFF enthält vor allen Dingen den Bildschirmspeicher für die CO-Routine.

3. Grundprogramm - Menüs

In diesem Kapitel werden die einzelnen Menüpunkte und Funktionen des Grundprogramms beschrieben.

Eine Rückkehr aus einem Untermenü erreicht man immer durch drücken der ESC (Escape) Taste.

Das Hauptmenü ist dynamisch aufgebaut, das heißt es gibt einige Menüpunkte, die immer vorhanden sind, Andere sind nur bei erkannter Hardware verfügbar.

3.1. Feste Menüpunkte

3.1.1. A = MEM aendern

3.1.2. D = MEM anzeigen

3.1.3. F = MEM fuellen

3.1.4. I = I/O lesen

3.1.5. O = I/O setzen

3.1.6. K = Krusader starten

3.1.7. W = WOZ Monitor

3.1.8. R = Programm starten

3.2. Dynamische Menüpunkte

3.2.1. U = Uhr stellen

3.2.2. C = Laufwerk waehlen

3.2.1. L = Datei laden

3.2.1. S = Datei speichern

3.2.1. X = EXE ausführen

3.2.1. V = Inhalt

3.2.1. E = Stick auswerfen

4. Die Unterprogramme

Die Unterprogramme des Grundprogramms werden über ein Sprungtabelle mit JSR aufgerufen.
Die jeweils benötigten Parameter werden in den internen Registern (A, X, Y) und/oder Registern in der Zeropage (R0, R0H,...) übergeben.

4.1. Die Sprungtabelle

Adr.	Name	Funktion
C100	csts	; Test ob Zeichen an Key
C103	ci	; Zeichen von Key holen
C106	ki	; Zeichen von Key holen und in Großbuchstaben
C109	co	; Zeichen auf Bildschirm ausgeben
C10C	setxy	; Schreibposition setzen (Text)
C10F	wait	; Warten bis GDP bereit
C112	delay	; A*1/10 Sekunden warten
C115	setpen	; Schreibstift setzen
C118	erapen	; Löschstift setzen
C11B	cmd	; Kommando an GDP
C11E	sync	; Sync Status einlesen
C121	reserve	
C124	reserve	
C127	reserve	
C12A	clear	; Bildschirmseite in A löschen
C12D	clrall	; alle Bildschirmseiten löschen
C130	clrinvis	; Aktuelle Schreibseite unsichtbar löschen
C133	clrscrn	; Bildschirm löschen co vorbereiten (CO)
C136	pset	; Punkt auf X/Y-Position schreiben (Grafik)
C139	pget	; Punkt auf X/Y-Position lesen (nur GDP-HS/GDP-FPGA)
C13C	moveto	; Setzt eine X/Y-Position (Grafik)
C13F	drawto	; Zeichnet eine Linie nach X/Y
C142	rect	; Zeichnet ein Rechteck
C145	write	; Schreibt einen Text
C148	center	; Text auf Bildschirm zentriert ausgeben
C14B	read	; Liest einen Text
C14E	load	; Lädt eine Datei
C151	save	; Speichert eine Datei
C154	reserve	; DIR
C157	reserve	; CD
C15A	reserve	; DEL
C15D	reserve	; REN
C160	reserve	
C163	ser2init	; SER2 Kanal A initialisieren
C166	si2st	; Testen ob Zeichen an SER2 Kanal A anliegt
C169	si2	; Zeichen von SER2 einlesen
C16C	so2st	; Prüfen ob SER2 bereit zum senden
C16F	so2	; Zeichen an SER2 ausgeben
C172	getuhr	; Datum und Uhrzeit lesen
C175	setuhr	; Datum und Uhrzeit setzen

Adr.	Name	Funktion
C178	sound	; Soundausgabe
C17B	reserve	
C17E	upper	; Wandelt ein Zeichen in Großbuchstaben
C181	ascii2hex	; Wandelt ein Zeichen von ASCII nach Hex
C184	ascii2bcd	; Wandelt ein Zeichen von ASCII nach BCD
C187	print4x	; Hexadezimale Ausgabe mit 4 Stellen
C18A	print2x	; Hexadezimale Ausgabe mit 2 Stellen
C18D	print8b	; Binäre Ausgabe mit 8 Stellen
C190	print2d	; Dezimale Ausgabe eines Bytes
C193	init_nmi	; Startet und initialisiert den NMI-Aufruf
C196	stop_nmi	; Stoppt den NMI-Aufruf
C199	run_unmi	; Startet und initialisiert ein NMI-Userprogramm
C19C	stop_unmi	; Stoppt ein NMI-Userprogramm
C19F	frame	; Kopf und Fuß ausgeben
C1A2	clrcenter	; Aktuelle Schreibseite Mittelbereich loschen
C1A5	ci_sek	; Zeichen von Tastatur lesen mit Zeitausgabe
C1A8	ki_sek	; Zeichen von Tastatur lesen mit Zeitausgabe und in Großbuchstaben wandeln

4.2. Beschreibung der Unterprogramme

Adresse: $C100
Befehlsname: CSTS
Kurzbeschreibung: Testen ob Zeichen an Tastatur anliegt

Eingaberegister: Keine
Ausgaberegister: A = $00/FALSE kein Zeichen da
$FF/TRUE Zeichen verfügbar
CARRY = 0 kein Zeichen da
1 Zeichen verfügbar
Zerstörte Register: Keine

Adresse: $C103
Befehlsname: CI
Kurzbeschreibung: Zeichen von KEY abholen

Eingaberegister: Keine
Ausgaberegister: A = Zeichen
Zerstörte Register: Keine

Adresse: $C106
Befehlsname: KI
Kurzbeschreibung: Zeichen von KEY abholen und in Großbuchstaben wandeln

Eingaberegister: Keine
Ausgaberegister: A = Zeichen
Zerstörte Register: Keine

Adresse: $C109
Befehlsname: CO
Kurzbeschreibung: Zeichen auf Bildschirm ausgeben

Eingaberegister: A = Zeichen
Ausgaberegister: Keine
Zerstörte Register: Keine

Sonderzeichen:

CTRL-A ($01)	auf 40 Zeichen
CTRL-B ($02)	auf 80 Zeichen
CTRL-E ($05)	Cursor an
CTRL-F ($06)	Cursor aus
CTRL-G ($07)	Bell
BS ($08)	Backspace
CTRL-L ($0C)	Bildschirm löschen, aber Cursor behalten
CR ($0D)	Carriage return

Adresse: $C10C
Befehlsname: SETXY
Kurzbeschreibung: Schreibposition setzen

Eingaberegister: R1 = Spalte 0-79 bzw. 0-39
R2 = Zeile 0-23
Ausgaberegister: Keine
Zerstörte Register: A

Adresse: $C10F
Befehlsname: WAIT
Kurzbeschreibung: Warten bis GDP bereit

Eingaberegister: Keine
Ausgaberegister: Keine
Zerstörte Register: A

Adresse: $C112
Befehlsname: DELAY
Kurzbeschreibung: Wartet A*1/10 Sekunden

Eingaberegister: A = Wartezeit
Ausgaberegister: Keine
Zerstörte Register: A, TEMP

Adresse: $C115
Befehlsname: SETPEN
Kurzbeschreibung: Schreibstift setzen

Eingaberegister: Keine
Ausgaberegister: Keine
Zerstörte Register: A

Adresse: $C118
Befehlsname: ERAPEN
Kurzbeschreibung: Löschstiftstift setzen

Eingaberegister: Keine
Ausgaberegister: Keine
Zerstörte Register: A

Adresse: $C11B
Befehlsname: CMD
Kurzbeschreibung: Kommando an GDP ausgeben

Eingaberegister: A = Kommando
Ausgaberegister: Keine
Zerstörte Register: Keine

Adresse: $C11E
Befehlsname: SYNC
Kurzbeschreibung: Sync Status einlesen

Eingaberegister: Keine
Ausgaberegister: A = $ff/TRUE bei sync,
sonst $00/FALSE
CARRY = 0 kein Sync
1 bei Sync
Zerstörte Register: Keine

Adresse: $C12A
Befehlsname: CLEAR
Kurzbeschreibung: Bildschirmseite löschen

Eingaberegister: A = Seitennummer
Ausgaberegister: Keine
Zerstörte Register: A

Adresse: $C12D
Befehlsname: CLRALL
Kurzbeschreibung: Alle Bildschirmseiten löschen

Eingaberegister: Keine
Ausgaberegister: Keine
Zerstörte Register: A

Adresse: $C130
Befehlsname: CLRINVIS
Kurzbeschreibung: Aktuelle Schreibseite unsichtbar löschen

Eingaberegister: Keine
Ausgaberegister: Keine
Zerstörte Register: A, X, Y, R1, R2

Adresse: $C133
Befehlsname: CLRSCRN
Kurzbeschreibung: Bildschirm löschen und CO vorbereiten

Eingaberegister: Keine
Ausgaberegister: Keine
Zerstörte Register: ...

Adresse: $C136
Befehlsname: PSET
Kurzbeschreibung: Punkt auf Bildschirm schreiben

Eingaberegister: Keine
Ausgaberegister: Keine
Zerstörte Register: A

Adresse: $C139
Befehlsname: PGET
Kurzbeschreibung: Punkt vom Bildschirm lesen (nur GDP-HS und GDP-FPGA)

Eingaberegister: Keine
Ausgaberegister: A
Zerstörte Register: Keine

Adresse: $C13C
Befehlsname: MOVETO
Kurzbeschreibung: X- und Y- Position setzen

Eingaberegister: R1(H) = X-Koordinate (Wort)
R2 = Y-Koordinate (Byte)
Ausgaberegister: Keine
Zerstörte Register: A

Adresse: $C13F
Befehlsname: DRAWTO
Kurzbeschreibung: Linie zu X- und Y- Position zeichnen

Eingaberegister: R1(H) = X-Koordinate (Wort)
R2 = Y-Koordinate (Byte)
Ausgaberegister: Keine
Zerstörte Register: A, R0, R5-R7

Adresse: $C142
Befehlsname: RECT
Kurzbeschreibung: Zeichnet ein Rechteck

Eingaberegister: R1(H) = X-Links (Wort)
R2 = Y-Unten (Byte)
R3(H) = X-Rechts (Wort)
R4 = Y-Oben (Byte)
Ausgaberegister: Keine
Zerstörte Register: A, R0, R5-R7

Adresse: $C145
Befehlsname: WRITE
Kurzbeschreibung: Schreibt einen Text auf den Bildschirm

Eingaberegister: R0 = SIZE (Byte)
R0H = Style (Byte)
R1(H) = X-Koordinate (Wort)
R2 = Y-Koordinate (Byte)
R3(H) = Textadresse (Wort)
Ausgaberegister: Keine
Zerstörte Register: A, Y, R4, R4H

Adresse: $C148
Befehlsname: CENTER
Kurzbeschreibung: Schreibt einen Text zentriert auf den Bildschirm

Eingaberegister:
- R0 = SIZE (Byte)
- R0H = Style (Byte)
- R2 = Y-Koordinate (Byte)
- R3(H) = Textadresse (Wort)

Ausgaberegister: Keine
Zerstörte Register: A, Y, R1(H), R4(H)

Adresse: $C14B
Befehlsname: READ
Kurzbeschreibung: Liest einen Text

Eingaberegister:
- R0 = SIZE (Byte)
- R0H = Style (Byte)
- R1(H) = X-Koordinate (Wort)
- R2 = Y-Koordinate (Byte)
- R2H = TRUE($FF)=>Großbuchstaben
FALSE($00)=>Gross/Kleinbuchstaben
- R3(H) = Textpuffer (Wort)
- R4 = Anzahl Zeichen max. (Byte)
- R4H = Umrandung TRUE($FF)/FALSE($00) (Byte)

Ausgaberegister:
- A = Status

Zerstörte Register: ...

Adresse: $C14E
Befehlsname: LOAD
Kurzbeschreibung: Öffnen, Laden und Schließen einer Datei

Eingaberegister:
- R0(H) = Pointer auf Name
- R3(H) = Pointer auf Speicher
- R4 = aktuelles Laufwerk ($01=CAS-Neo, $02=USB-Stick)

Ausgaberegister:
- A = Status
- X = Fehlernummer ($30-$32)
- R3(H) = nächstes Zeichen (00)

Zerstörte Register: A, Y

Adresse: $C151
Befehlsname: SAVE
Kurzbeschreibung: Öffnen, speichern und Schließen einer Datei

Eingaberegister:
- R0(H) = Pointer auf Name
- R1(H) = Länge der Datei
- R3(H) = Pointer auf Speicher
- R4 = aktuelles Laufwerk ($01=CAS-Neo, $02=USB-Stick)

Ausgaberegister:
- A = Status
- X = Fehlernummer ($38-$3b)

Zerstörte Register: A, X, Y, R3(H)

Adresse: $C163
Befehlsname: SER2INIT
Kurzbeschreibung: SER2 Kanal A initialisieren

Eingaberegister: R0(H) = Zeiger auf Parameter
Ausgaberegister: Keine
Zerstörte Register: Keine

Adresse: $C166
Befehlsname: SI2ST
Kurzbeschreibung: Testen ob Zeichen an SER2 Kanal A anliegt

Eingaberegister: Keine
Ausgaberegister: CARRY = 0 kein Zeichen da
1 Zeichen verfügbar
Zerstörte Register: Keine

Adresse: $C169
Befehlsname: SI2
Kurzbeschreibung: Zeichen von SER2 einlesen

Eingaberegister: Keine
Ausgaberegister: A = Zeichen
Zerstörte Register: Keine

Adresse: $C16C
Befehlsname: SO2ST
Kurzbeschreibung: Prüfen ob SER2 bereit zum senden

Eingaberegister: Keine
Ausgaberegister: CARRY = 0 nicht bereit
1 bereit
Zerstörte Register: Keine

Adresse: $C16F
Befehlsname: SO2
Kurzbeschreibung: Zeichen an SER2 ausgeben

Eingaberegister: A = Zeichen
Ausgaberegister: Keine
Zerstörte Register: Keine

Adresse: $C172
Befehlsname: GETUHR
Kurzbeschreibung: Datum und Uhrzeit lesen

Eingaberegister: Keine
Ausgaberegister: SYSRAM: Adresse $B000 - $B007
Zerstörte Register: A

Adresse: $C172
Befehlsname: SETUHR
Kurzbeschreibung: Datum und Uhrzeit setzen

Eingaberegister: SYSRAM: Adresse $B000 - $B007
Ausgaberegister: Keine
Zerstörte Register: A

Adresse: $C178
Befehlsname: SOUND
Kurzbeschreibung: Soundausgabe

Eingaberegister: R3(H) = Soundtabelle
Ausgaberegister: Keine
Zerstörte Register: A, X, Y

Adresse: $C17E
Befehlsname: UPPER
Kurzbeschreibung: Wandelt ein Buchstabe in Großschrift

Eingaberegister: A
Ausgaberegister: A
Zerstörte Register: Keine

Adresse: $C181
Befehlsname: ASCII2HEX
Kurzbeschreibung: Wandelt ein ASCII-String nach Hexadezimal

Eingaberegister: R3(H) = Zeiger auf ASCII-String
Ausgaberegister: R5(H) = Hexadezimalwert
Zerstörte Register: A, X, Y

Adresse: $C184
Befehlsname: ASCII2BCD
Kurzbeschreibung: Wandelt ein ASCII-String nach BCD

Eingaberegister: R3(H) = Zeiger auf ASCII-String
Ausgaberegister: R5(H) = BCD-Wert
Zerstörte Register: A, Y

Adresse: $C187
Befehlsname: PRINT4X
Kurzbeschreibung: Hexadezimale Ausgabe mit 4 Stellen

Eingaberegister: R0(H) = Zeichen
R3(H) = Textadresse
Ausgaberegister: R3(H) = nächstes Textadresse
Zerstörte Register: A, Y

Adresse: $C18A
Befehlsname: PRINT2X
Kurzbeschreibung: Hexadezimale Ausgabe mit 2 Stellen

Eingaberegister: A = Zeichen
R3(H) = Textadresse
Ausgaberegister: R3(H) = nächstes Textadresse
Zerstörte Register: A, Y

Adresse: $C18D
Befehlsname: PRINT8B
Kurzbeschreibung: Binäre Ausgabe mit 8 Stellen

Eingaberegister: A = Zeichen
R3(H) = Textadresse
Ausgaberegister: R3(H) = nächstes Textadresse
Zerstörte Register: A, X, Y

Adresse: $C190
Befehlsname: PRINT2D
Kurzbeschreibung: Dezimale Ausgabe mit 2 Stellen

Eingaberegister: A = Zeichen
R3(H) = Textadresse
Ausgaberegister: R3(H) = nächstes Textadresse
Zerstörte Register: A, X, Y , R0, R1

Adresse: $C193
Befehlsname: INIT_NMI
Kurzbeschreibung: Startet und initialisiert den NMI-Aufruf

Eingaberegister: Keine
Ausgaberegister: Keine
Zerstörte Register: Keine

Adresse: $C196
Befehlsname: STOP_NMI
Kurzbeschreibung: Stoppt den NMI-Aufruf

Eingaberegister: Keine
Ausgaberegister: Keine
Zerstörte Register: Keine

Adresse: $C199
Befehlsname: RUN_UNMI
Kurzbeschreibung: Startet und initialisiert ein NMI-Userprogramm

Eingaberegister: R3(H) = Programmadresse
Ausgaberegister: Keine
Zerstörte Register: Keine

Adresse: $C19C
Befehlsname: STOP_UNMI
Kurzbeschreibung: Stoppt ein NMI-Userprogramm

Eingaberegister: Keine
Ausgaberegister: Keine
Zerstörte Register: Keine

Adresse: $C19F
Befehlsname: FRAME
Kurzbeschreibung: Gibt Kopf- und Fußzeile aus

Eingaberegister: R3(H) = Textadresse (Wort)
Ausgaberegister: Keine
Zerstörte Register: ...

Adresse: $C1A2
Befehlsname: CLRCENTER
Kurzbeschreibung: Löscht den Bildschirm zwischen Kopf- und Fußzeile

Eingaberegister: Keine
Ausgaberegister: Keine
Zerstörte Register: ...

Adresse: $C1A5
Befehlsname: CI_SEC
Kurzbeschreibung: Zeichen von Tastatur lesen mit Zeitausgabe

Eingaberegister: Keine
Ausgaberegister: A = Zeichen
Zerstörte Register: Y, R0(H) - R4(H)

Adresse: $C1A8
Befehlsname: KI_SEC
Kurzbeschreibung: Zeichen von Tastatur lesen mit Zeitausgabe
und in Großbuchstaben wandeln

Eingaberegister: Keine
Ausgaberegister: A = Zeichen
Zerstörte Register: Y, R0(H) - R4(H)