==========

ADIS51
Disassembler mit One-Line Assembler
fuer die
8051-Microcontroller Familie

==========

Uebersicht

Mit dem Programm ADIS51 koennen absolute HEX-, OBJ- und BIN-Dateien, welche fuer die Microcontroller der 8051 Familie generiert wurden, disassembliert werden. Zudem ist es moeglich, die Daten solcher Dateien als Hex-Wert oder als Befehl ueber den eingebauten One-Line Assembler einfach zu modifizieren und sie als HEX- oder BIN-Datei wieder abzuspeichern. Der Inhalt eines 64K Byte grossen Programmspeichers wird dazu im Assemblerformat oder als Hex-Dump dargestellt. Die Ausgabe des Programmspeicherinhalts in eine LOG-Datei ist in einem List-Format oder Assembler-Source Format moeglich.
Der Disassembler unterstuetzt in Abhaengigkeit des ausgewaehlten Microcontrollertyps dessen spezifischen SFR- und SFB-Symbole. Nach dem Laden einer Datei kann festgestellt werden, in welchen Programmspeicherbereich Code bzw. Daten geladen wurden.

Inhalt

1. Aufruf des Programms

Beim Aufruf des Programms ADIS51.EXE ohne weiteren Parameter wird der 25-Zeilen Text-Modus (mit 80 Spalten) eingestellt. Das Programm unterstuetzt in diesem Modus Monochrom, EGA und VGA oder dazu kompatible Bildschirmadapter. Fuer EGA und VGA Bildschirmadapter kann beim Aufruf des Programms auch der 43- bzw. 50-Zeilenmodus eingestellt werden. Der Aufruf fuer diese Betriebsart lautet :

ADIS51 /43

Nach dem Aufruf von ADIS51 meldet sich das Programm mit dem Titel-Bildschirm Nach Druecken einer beliebigen Taste gelangt man in das Hauptmenü.

2. Hauptmenue

Der Bildschirm des Hauptmenues gliedert sich in 3 Teile :

a) Im oberen Bereich des Bildschirms (3 Zeilen) werden die jeweils ausfuehrbaren Funktionen des Programms mit der jeweils ausloesenden Taste angegeben.
b) Der mittlere Teil des Bildschirms dient zur Anzeige vom Inhalt des 64K Programmspeichers. Er wird zur Ausgabe/Bearbeitung der Disassembler- und der Hex-Dump Funktion sowie zur Anzeige von verschiedenen Informationen in Bildschirmfenstern benutzt. Die Anzahl der Zeilen dieses Bereichs haengt vom dem beim Aufruf von ADIS51 eingestellten Text-Modus ab (16, 34 oder 41 Zeilen).
c) Im unteren Teil des Bildschirms (2 Zeilen) werden Statusinformationen und Fehlermeldungen ausgegeben (siehe Anhang A). Auch die Eingabe von Dateinamen und Adressen erfolgt in diesem Bereich des Bildschirms. Des weiteren wird im rechten Teil dieses Bildschirmbereichs der ausgewaehlte Microcontrollertyp und der Zustand der LOG-Datei (geoeffnet oder geschlossen) angezeigt.

Vom Hauptmeue aus koennen ueber 7 Funktionstasten die Grundfunktionen des Programms aufgerufen werden. Diese Grundfunktionen des Hauptmenues sind wiefolgt den Funktionstasten F1 bis F7 zugeordnet :

<F1> Laden einer absoluten HEX-/OBJ- oder BIN-Datei
<F2> Speichern des Programmspeicherinhalts in einer HEX- oder BIN-Datei
<F3> Auswahl des Microcontrollertyps
<F4> Aufruf des Disassemblers / One-Line Assemblers
<F5> Bearbeitung des Programmspeicherinhalts als HEX-Dump
<F6> Anzeige der Programmspeicherbelegung
<F7> Einstellen von Optionen

Die Betaetigung einer falschen Taste fuehrt optisch (Fehlermeldung) und akustisch zu einer Warnung. Mit <Esc> im Hauptmenue wird ADIS51 beendet und nach DOS zurueckgekehrt. Im folgenden werden die Grundfunktionen des Hauptmenues nacheinander beschrieben.

3. Laden einer absoluten HEX-/OBJ- oder BIN-Datei

Das Programm ADIS51 enthaelt einen 64K grossen Datenpuffer (Programmspeicher), in welchen absoluter Code und Daten, der ueblicherweise von 8051 Assemblern bzw. Compilern erzeugt wird, geladen werden koennen. Die Speicherung von absolutem Code bzw. Daten in Dateien erfolgt bei 8051 Assemblern und Compilern entweder im von der Firma Intel definierten Hexadezimal- oder Object-Format. Als Ausgabeformat vom EPROM/PROM Programmiergeraeten wird haeufig als Dateiformat ein BIN-Format benutzt. Alle 3 Dateiformate kann ADIS51 verarbeiten.

Vom Hauptmenue aus wird ueber <F1> das Laden einer Datei eingeleitet und danach das entsprechende Dateiformat ausgewaehlt (Eingabe von <H>, <O> oder <B> fuer HEX-, OBJ- oder BIN-Datei). Nach der Eingabe des Dateinamens wird die angegebene Datei geoeffnet, gelesen, die fuer ADIS51 relevante Information (absoluter Code/Daten) herausgefiltert und in den 64K-Programmspeicher uebertragen. Waehrend dem Laden des Programmspeichers wird festgehalten, in welchen Speicherbereich des Programmspeichers Code/Daten uebertragen werden. Nach dem Laden einer Datei kann dies ueber die Funktion "Code RAM-Info" (<F6> Hauptmenü) angezeigt werden. Am Ende eines Ladevorgangs wird die Anzahl der geladenen Bytes angegeben (als Hexadezimalwert).
Vor dem Laden einer HEX-, OBJ- oder BIN-Datei kann der 64k Programmspeicher immer komplett geloescht werden (mit 00H beschrieben). Diese Funktion ist auch abschaltbar (mit <F7> Optionen im Hauptmenü).
Beim Laden einer HEX-Datei werden als Code/Daten "Records" mit dem "Recordtyp" 00H als Nutzinformation in den Programmspeicher uebertragen. Bei OBJ-Dateien wird nur Code/Daten von Records mit dem "Recordtyp" 06H ausgewertet und in den Programmspeicher uebertragen. Symbolinformationen aus HEX- oder OBJ-Dateien wird nicht beruecksichtigt. Da BIN-Dateien keinerlei Adressinformationen enthalten, muss vor dem Laden einer BIN-Datei eine Startadresse fuer die zu uebertragenden Daten im Hexadezimalformat angegeben und mit <Return> abgeschlossen werden.

4. Speichern des Programmspeicherinhalts als absolute HEX-/BIN-Datei

Mit dieser Funktion, welche ueber <F2> aus dem Hauptmenue aufgerufen wird, kann der Programmspeicherinhalt als absolute HEX-Datei oder als BIN-Datei abgespeichert werden. Nach der Betaetigung von <F2> muss der Dateiname der HEX- oder BIN-Datei, welche erzeugt werden soll, eingegeben werden. Mit <Return> wird die Eingabe des Dateinamen abgeschlossen. Ist der Dateiname korrekt eingegeben worden, muessen Start- und Endadresse des zu uebertragenden Programmspeicherbereichs angegeben werden. Die beiden 4-stelligen Adressen muessen jeweils im Hexadezimalformat angegeben und mit <Return> abgeschlossen werden.
Wenn Daten aus unterschiedlichen, getrennten Bereichen des Programmspeichers zu einer HEX-Datei zusammengefasst werden sollen, kann dies nur ausserhalb des Programms ueber einen Editor erfolgen. Dazu sind einzelne HEX-Dateien der zusammenhaengenden Programmspeicherereiche zu erzeugen und zusammenzufuegen. Der "End-of-File" Record der HEX-Dateien (bis auf den Record der letzten Datei) muessen dabei entfernt werden.

5. Auswahl des Microcontrollertyps

Mit der Funktion <F3> des Hauptmenues wird ein Microcontrollertyp ausgewaehlt. Nach Betaetigung von <F3> erscheint ein Bildschirmfenster, welches die auswaehlbaren MCU Typen anzeigt. Nach Eingabe einer Ziffer von 1 bis 7 mit anschliessendem <Return> ist der entsprechende Microcontrollertyp selektiert.

Folgende MCU-Typen stehen zur Auswahl :

8051/8031 8052/8032 80512/80532
80(C)515/80(C)535 80C517/80C537 80C515A/80C835A
80C517A/80C537A

Der ausgewaehlte MCU Typ wird immer im unteren rechten Teil der Statuszeilen angezeigt. Nach Aufruf des Programms ist die MCU 8051 ausgewaehlt. Durch die Auswahl eines MCU-Typs werden fuer den Disassembler und Assembler insbesondere die Symbole der "Special Function Register" und der Bits selektiert.

6. Aufruf des Disassemblers/Assemblers

Mit der Funktion <F4> des Hauptmenues wird der Disassembler als Untermenue aufgerufen und zur Eingabe einer Startadresse (hexadezimal mit <Return> abgeschlossen) aufgefordert. Von dieser Startadresse aus beginnt der Disassembler den Inhalt des 64K Programmspeichers zu disassemblieren. Dazu berechnet das Programm ADIS51, ausgehend von der Startadrresse, die Adressen der naechsten 2000 Befehle und gibt die ersten Befehle ab der Startadresse auf dem Bildschirm aus. Der Befehl im hervorgehobenen horizontalen Balken kann ab der Cursorposition editiert bzw. geaendert werden.

Im Untermenue des Disassemblers koennen folgende Funktionen ueber entsprechende Tasten ausgefuehrt werden :

<Cursor Up> Disassembler/Assembler um einen Befehl zurueck scrollen
<Cursor Down> Disassembler/Assembler um einen Befehl vorwaerts scrollen
<Page Up> Disassembler/Assembler um eine Seite zurueck scrollen
<Page Down> Disassembler/Assembler um eine Seite vorwaerts scrollen
<F9> Eingabe/Definition einer neuen Startadresse
<F10> LOG-Datei bearbeiten/ oeffnen/ schliessen
<Esc> Zurueck zum Hauptmenue

Nach Modifikation eines Befehls wird dieser durch Betaetigung der <Return> Taste vom Assembler geprueft und, falls als fehlerfrei erkannt, der entsprechende Code in den Programmspeicher eingetragen. Vorsicht : in diesem Fall wird der dem gerade assemblierten Befehl folgende Inhalt des Programmspeichers neu disassembliert. Dies kann zur Folge haben, dass eine bestehende Befehlsfolge zerstoert wird (z.B. Assemblieren/Eingabe eines 3-Byte Befehls anstelle eines 1-Byte Befehls).
Bei der Dekodierung der Befehle prueft der Disassembler/Assembler abhaengig von ausgewaehlten MCU-Typ, ob die Operanden entsprechender Befehle einem Register- oder Bit-Symbol entsprechen. Ist dies der Fall, erscheint das entsprechende Symbol im Befehl. Diese Funktion kann fuer den Disassembler im Hauptmenue unter <F7> "Optionen" ausgeschaltet werden. Die Ausgabe von Bytes bzw. Worten erfolgt im Hexadezimal-Format. Sprungadressen werden immer als absolute Adresse ausgegeben. Die Eingabe von Bytes und Worten kann dezimal oder hexadezimal erfolgen. Im Anhang B sind weitere Konventionen zum Assembler angegeben.
Mit der Eingabe von <F9> kann fuer den Disassembler eine neue Startadresse (hexadezimal mit <Return> abgeschlossen) eingegeben werden. Die Bearbeitung einer LOG-Datei, was ueber <F10> gesteuert wird, ist in Abschnitt 8 beschrieben. Das Untermenue des Disassemblers wird durch Eingabe von <Esc> wieder verlassen.

7. Anzeige und Modifikation des Programmspeichers als HEX-Dump

Mit der Funktion <F5> des Hauptmenues wird die HEX-Dump Funktion von ADIS51 als Untermenue aufgerufen und zur Eingabe einer Adresse (hexadezimal mit <Return> abgeschlossen) aufgefordert. Von dieser Adresse beginnend werden die folgenden Bytes des 64K Programmspeichers als Hexadezimalwert und in ASCII-Darstellung ausgegeben (16 Bytes pro Zeile). Das in der Bidschirmdarstellung jeweils hervorgehobene Byte kann als Hexadezimalwert oder als ASCII-Wert veraendert werden. Die Adresse des gerade selektierten Bytes wird in den Statuszeilen als Hexadezimalwert angezeigt.

Im Untermenue der HEX-Dump Funktion koennen folgende Funktionen ueber entsprechende Tasten ausgefuehrt werden :

<Cursor Down> Adresse des aktuellen Bytes um 16 erhoehen (optional scrollen)
<Cursor Up> Adresse des aktuellen Bytes um 16 verringern (optional scrollen)
<Cursor Right> Adresse des aktuellen Bytes um 1 erhoehen (optional scrollen)
<Cursor Left> Adresse des aktuellen Bytes um 1 verringern (optional scrollen)

<Page Up> HEX-Dump um eine Bildschirmseite zurueck scrollen
<Page Down> HEX-Dump um eine Bildschirmseite vorwaerts scrollen
<F9> Eingabe einer neuen Adresse
<F10> LOG-Datei bearbeiten/ oeffnen/ schliessen
<Tab> Wechsel zwischen Eingabe in HEX oder ASCII Feld
<Esc> Zurueck zum Hauptmenue

Mit der Eingabe von <F9> kann eine neue Startadresse (hexadezimal mit <Return> abgeschlossen) eingegeben werden. Die Bearbeitung einer LOG-Datei, was ueber <F10> gesteuert wird, ist im naechsten Abschnitt beschrieben. Das Untermenue der HEX-Dump Funktion wird durch Eingabe von <Esc> wieder verlassen.

Hinweis : Die HEX-Dump Funktion eignet sich generell auch zum Analysieren/ Modifizieren einer max. 64k grossen Datei. Dazu wird die Datei einfach als BIN-Datei geladen. Nachdem Start- und Endadresse festgestellt wurden, koennen Modifikationen vorgenommen und mit "Save BIN-File" als Datei wieder abgespeichert werden.

8. Erzeugen einer LOG-Datei

Im Untermenu des Disassemblers/Assemblers und der HEX-Dump Funktion koennen Daten aus diesen Funktionen in eine LOG-Datei geschrieben werden. Die Disassembler Daten koennen gemischt in einem Listing Format oder Assembler-Source Format generiert werden. Die Formatauswahl wird im Hauptmenue ueber <F7> "Options" selektiert. Daten aus dem HEX-Dump werden ebenfalls entweder in Listing Format oder in Assembler-Source Format in die LOG-Datei uebertragen.
Der Status der LOG-Datei (geschlossen oder geoeffnet) wird jeweils im unteren rechten Teil der Statuszeilen angezeigt.
Bei Eingabe von <F10> im Untermenue des Disassemblers/Assemblers oder der HEX-Dump Funktion bei geschlossener LOG-Datei (Statusmeldung "LOG-File : closed") wird zur Eingabe des Dateinamens der LOG-Datei aufgefordert. Die Eingabe des Dateinamens wird mit <Return> abgeschlossen. Der LOG-Dateiname ADIS51.LOG kann als Default uebernommen werden (unmittelbar <Return> nach <F10>).
Wird bei geoeffneter Datei (Statusmeldung "LOG-File : opened") im Untermenu des Disassemblers/ Assemblers oder der HEX-Dump Funktion <F10> betaetigt, kann die LOG-Datei geschlossen (<C> fuer "closed") oder ergaenzt (<A> fuer "add") werden. Es kann immer nur eine LOG-Datei geoeffnet sein. Beim oeffnen einer LOG-Datei wird geprueft, ob schon eine Datei unter dem angegebenen Dateinamen existiert (Auswahlmoeglichkeit : Datei ueberschreiben oder abbrechen). Bei Rueckkehr des Programms nach DOS wird eine geoeffnete LOG-Datei automatisch geschlossen.
Start- und Endadresse des Programmspeicherbereichs, welcher in den LOG-File geschrieben werden soll, muessen jeweils als 4-stellige Zahlen im Hexadezimal format eingegeben und mit <Return> abgeschlossen werden.

9. Anzeige der Programmspeicherbelegung

Diese mit <F6> aus dem Hauptmenue aufrufbare Funktion erzeugt eine Uebersicht ueber die Programmspeicherbelegung. Beim Laden einer HEX- oder OBJ-Datei (siehe 3.) wird diese Information erzeugt. Mit dieser Funktion kann somit uebersichtlich festgestellt werden, in welche Programmspeicherbereiche eine HEX-/OBJ Datei ihren Code/ Daten plaziert hat.
Ein Blockgrafik-Zeichen der Anzeige stellt 1K Byte des Programmspeichers dar. Ein ausgefuelltes Zeichen (gelb) bedeutet, dass im entsprechenden 1K Byte Block Code/Daten geladen wurden. Das Loeschen des Programmspeichers vor dem Laden einer Datei kann ueber <F7> Optionen verhindert werden (siehe naechsten Abschnitt).
Die Anzeige zur Programmspeicherbelegung wird durch Betaetigung einer beliebigen Taste wieder beendet.

10. Einstellung von Optionen

Mit dieser ueber <F7> aus dem Hauptmenue aufrufbaren Funktion koennen folgende 3 Optionen eingestellt werden :

Disassembler-Ausgabe mit SFR- und Bit-Symbolen (mit / ohne)
Loeschen des Programmspeichers vor dem Laden einer Datei (mit / ohne)
Auswahl des Format der Daten in der LOG-Datei (List- oder Assembler-Format)

Die Optionen werden durch Eingabe von <1>, <2> oder <3> entsprechend gesetzt. Mit Betaetigung einer anderen Taste wird diese Funktion wieder ver-

lassen.

Anhang A : Fehler- und Statusmeldungen

Fehlermeldungen	Ursache
Invalid input	Ungueltige Eingabe eines Zeichens
File access error	Eingabe eines ungueltigen Dateinamens bzw. Datei mit dem angegebenen Namen existiert nicht.
Invalid address	Ungueltige Adresse (nur Hexadezimalzeichen erlaubt)
Checksum error	Beim Laden einer HEX- oder OBJ-Datei wurde ein Checksumfehler festgestellt. Das Laden der HEX-/OBJ Datei wurde abgebrochen.
No absolute code/data has been loaded !	Beim Laden einer HEX- oder OBJ-Datei wurden keine HEX-/OBJ-Records mit absolut definierten Daten erkannt. Moeglicherweise besteht die geladene Datei aus einem falschen Dateiformat.
Invalid File Format	Beim Laden einer HEX-/OBJ-Datei wurde ein fehlerhaftes Dateiformat festgestellt; Ladevorgang wurde abgebrochen.
Invalid instruction	Ungueltiger Befehl; Assembler kann den eingegebenen Befehl nicht assemblieren.
Display buffer exceeded	Zum weiteren Disassemblieren ueber die aktuelle Adresse hinaus muss eine neue Startadresse angegeben werden (nur 2000 Befehle ab Startadresse disassemblierbar)..
Code memory limit reached	Beim Disassembler oder in der HEX-Dump Funktion wurden die Grenzen des Programmspeichers erreicht.
End of code memory - input lower address	Mit der eingegebenen Startadresse des Disassemblers koennen die folgenden Befehle bis zur oberen Grenze des Programmspeichers nicht auf einer kompletten Bildschirmseite dargestellt werden. Abhilfe: Eingabe einer kleineren Startadresse.
Disassembler start address reached	Startadresse, von der aus der Disassembler die folgenden 2000 Befehle dekodiert, ist erreicht. Von der Startadresse zu kleineren Adressen hin kann nicht disassembliert werden.
Start address is greater than end address	Beim Beschreiben einer LOG-Datei ist die Startadresse des zu uebertragenden Bereich kleiner als die Endadresse.

Statusmeldungen	Bedeutung
x-File <Name> is loaded	x="HEX","OBJ" oder "BIN"; Datei <Name> wird analysiert und geladen.
File <Name> is created	Es werden Daten in eine HEX- oder BIN-Datei uebertragen.

LOG-File is created	Es werden Daten in die LOG-Datei uebertragen.

Anhang B : One-Line Assembler Konventionen

Der in ADIS51 eingebaute One-Line Assembler verarbeitet die Standard 8051-Befehle nach der folgenden Tabelle :

Opcode	Mnemonic
MOV	A,Rn
	A,direct
	A,@Ri
	A,#data
	Rn.A
	Rn,direct
	Rn,#data
	direct,A
	direct,Rn
	direct,direct
	direct,@Ri
	direct,#data
	@Ri,A
	@Ri,direct
	@Ri,#data
	DPTR,#data16
MOVC	A,@A+DPTR
	A,@A+PC
MOVX	A,@Ri
	A,@DPTR
	@Ri,A
	@DPTR,A
XCH	A,Rn
	A,direct
	A,@Ri
XCHD	A,@Ri
PUSH	direct
POP	
ADD	A,Rn
ADDC	A,direct
SUBB	A,@Ri
	A,#data
ANL	A,Rn
ORL	A,direct
XRL	A,@Ri
	A,#data
	direct,A
	direct,#data
INC	A
DEC	Rn
	direct
	@Ri
INC	DPTR
MUL	AB
DIV	
CLR	A
CPL	
RL	A
RLC	
RR	
RRC	

DA SWAP	
PUSH POP	direct
XCH	A,Rn A,direct A,@Ri
XCHD	A,@Ri
ACALL AJMP	addr11
LCALL LJMP	addr16
JMP	@A+DPTR
SJMP	rel
JZ JNZ JC JNC	
JB	bit,rel
JNB JBC	
CJNE	A,direct,rel A,#data,rel Rn,#data,rel @Ri,#data,rel
DJNZ	Rn,rel direct,rel
CLR CPL SETB C	bit
ANL ORL MOV	C,bit
ANL ORL	C,/bit
MOV	bit,C
RET RETI NOP	-

Rn : R0 - R7
Ri : R0 + R1
direct : SFR-Symbol, Dezimalzahl 0 - 255, Hexadezimalzahl 00H - 0FFH
bit : SFB-Symbol, Dezimalzahl 0 - 255, Hexadezimalzahl 00H - 0FFH
data : Dezimalzahl 0 - 255, Hexadezimalzahl 00H - 0FFH
data16,rel,
addr11,addr16 : Dezimalzahl 0 - 65535, Hexadezimalzahl 00H - 0FFFFH