

ROMライターキット

ROMライターキットは「AKI-80ゴールドキット」と組み合わせてつかいます。



専用コントロールソフトがIBM-PC/互換機用にも対応！PC-9801、IBM-PC、DOS/Vいずれもホストとして使用できます。
必要環境：MS-DOS機、MS-DOS 3.0～(NEC)またはMS-DOS 5.0V～
RS-232Cケーブル他

PC9801 Controlled ROM WRITER

VER 1.2 with AKI80

ROMライターキット

- ★9801シリーズとRS232Cで接続、コントロールするROMライターキットです。
 - ★本体のみでもROMコピー、フランクチェック、コンペアなど可能です。
 - ★対応EPROMは2764、27128、27256の書込電圧12.5V、21Vタイプ及び2864 (EEPROM) です。
 - ★インテル方式高速書込対応。
 - ★98側用フルスクリーン・コントロール・ソフトAKIROM. EXEが付属し、ほとんどのコマンドが矢印キー「↑↓→←」で選択可能です。
 - ★書込ファイルフォーマットはインテルHEXファイルを採用しています。
 - ★インテルHEXユーティリティHEXUTL.COM (付属) にてバイナリtoHEXやHEX分割などが行えます。
 - ★PC9801、PC386、PC286のノーマル機で動作可能。
 - ★要MSDOS、RS232C用ノーマルケーブル(ストレート結線25PIN-25PIN)、AKI80 (256K SRAM付)
 - ★98側用ソフトはROMライターからCOPYコマンドにて98に転送します。(フロッピーいらずの新方式につきメディア指定は必要ありません。)
 - ★MSDOS ver 5.0にも対応。
- ★ROMライターVer1.2のスペシャルモードとして、「BASIC インタプリタ」及び「Z VISION REMORT」のターゲットとしての機能を持っています。いずれも別途に98用ホストプログラムは必要ですが、ROMライター上のメモリにてBASIC言語による開発や、デバッグが可能です。
- 「Z80 BASIC インタプリタ」はシステムロード製のZ80用インタプリタ及び専用ターミナルです。」
- 「Z VISION REMORT」はシステムロード製のZ80用フルスクリーン・ターゲット・デバッグです。

パーツリスト		数		
★半導体 代換品				
27C256-90		1	ROM-Wプログラム済み	
82C55-10	71055-10	1	パラレルI/O	
MAX (ICL) 232	LT1081 NJU6403	1	232ドライバ	
74HC14	各社相等品	1	シュミットインバータ	
NJM317	LM317T	2	可変三端子レギュレータ	
7805	各社相等品	1	三端子レギュレータ	
2SC1815	各社相等品	6	NPN汎用トランジスタ	
LED	各社相等品	5	3mm各色発光ダイオード	
★カーボン抵抗 表 示				
470Ω	黄紫茶金	5	誤差5%茶色の胴体	
4.7KΩ	黄紫赤金	2		
10KΩ	茶黒橙金	9		
★金属被膜抵抗				
150Ω	茶緑黒黒茶	1	誤差1%青の胴体	
240Ω	赤黄黒黒茶	1		
680Ω	青灰黒黒茶	1		
910Ω	白茶黒黒茶	1		
2.4KΩ	赤黄黒茶茶	1		
3KΩ	橙黒黒茶茶	1		
3.3KΩ	橙橙黒茶茶	1		
★積層セラミックコンデンサ				
0.1μF	104	8		バスコン
★電解コンデンサ				
10μF/10V以上		1	バスコン チャージポンプ	
33~220μF/6.3V以上		1	バスコン	
33~220μF/35V以上		1	バスコン	
★その他				
4.7KΩ抵抗アレイ		1	8素子入り	
14.7456MHz XTAL		1	水晶	
タクトSW		8		
ゼロプレッシャーソケット		1	28pゼロプレッシャーソケット	
Dsub 25p		1	RS232C用メス	
ピンヘッダ72p		1	AKI80/ライタボード連結	
ICソケット14p		1		
16p		1		
40p		1		
専用ボード	AE-ROM	1		

★LT1081はリニアテクノロジー社のMAX232同等品です。

★NJU6403はJRC社のMAX232同等品です。

★パーツには不足がないよう細心の注意を払っておりますが、万が一不足の場合、全交換になる場合もありますので、製作前にチェックの上、予め申し出ください。

★ICはCMOSですので、静電破壊の恐れがあります。十分気をつけた取扱いをお願いいたします。

★上記以外に、RS232Cノーマルケーブル(ストレート結線)、電源が必要です。また必要に応じてトグルSW1回路2接点の物か、または中点OFF付き1回路2接点の物を御用意ください。

★このパッケージ以外にAKI80(256K SRAM付)が必要です。

■-1- 製作

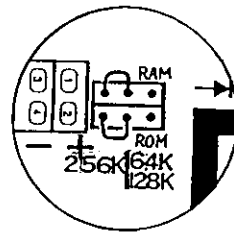
1-1 AKI80の製作。

AKI80の製作はAKI80付属のマニュアルをよく参照してください。このマニュアルでは補足事項のみ述べます。

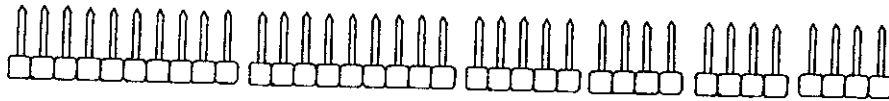
このキットの成功のカギはAKI80の半田付けにあると言っても過言ではありません。トラブルのほとんどがAKI80の半田付けです。フラットパッケージの半田付けには、細心の注意を払い、導通チェッカ又はテストでの導通チェックは怠らないでください。

使用するX'TALはROMライタ付属の14.7456MHzです。これ以外のX'TALでは動作できません。

RAM、ROM共に容量選択は256側です。

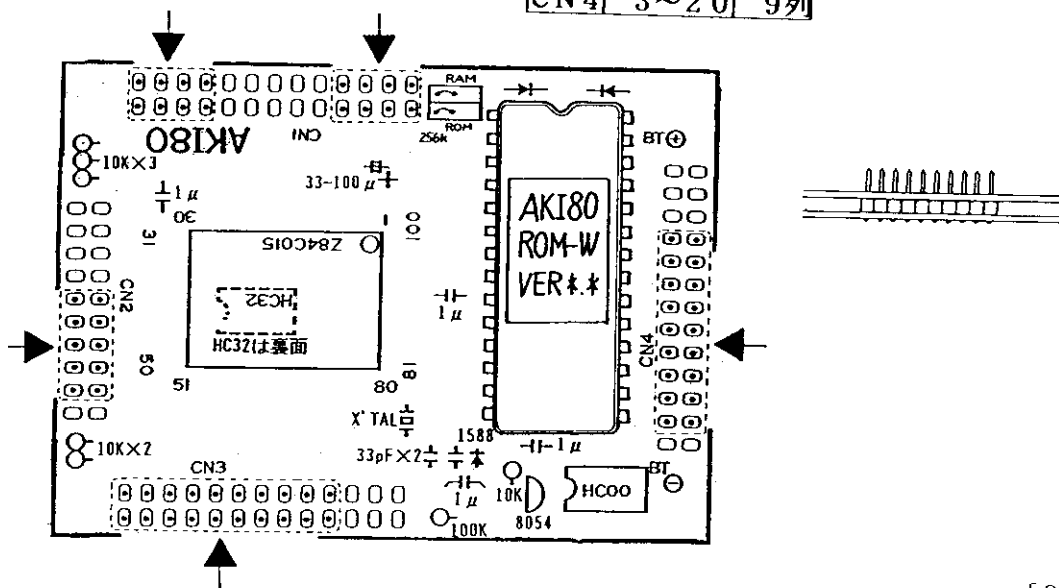


AKI80にピンヘッダを半田付けします。ピンヘッダは分割可能な72(2x36列)ピンタイプが一つ入っています。これをニッパで必要なサイズに割って使用します。分割サイズは、4列が2個。5列、9列、10列が各1個です。



下図のように差しこみ半田付けします。

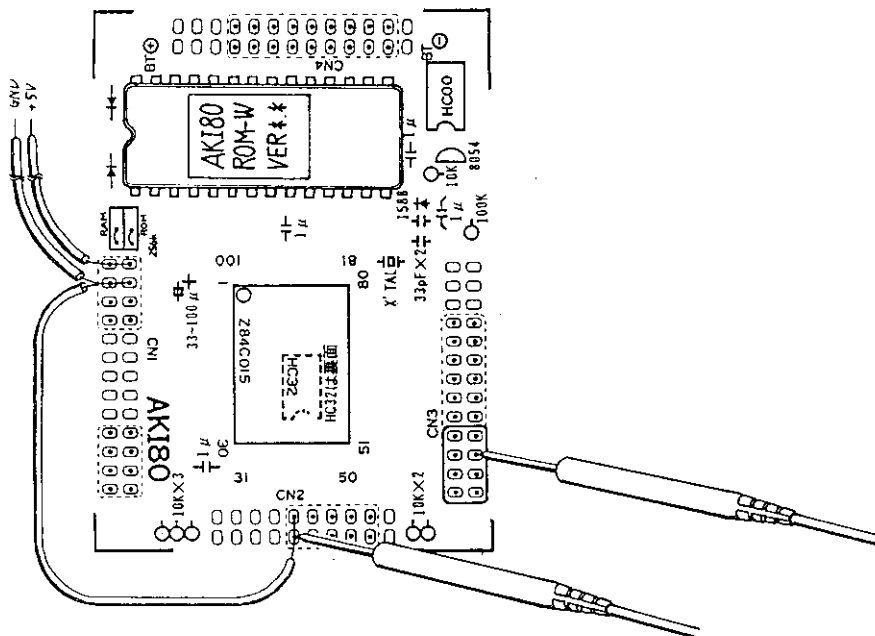
取りつけるピン		
CN1	1~ 8	4列
CN1	19~26	4列
CN2	9~18	5列
CN3	1~20	10列
CN4	3~20	9列



付属のROMには初期段階でAKI80が正常に動くかチェックモードを持っています。ROMライター接続後のAKI80トラブルシューティングは行いにくいので、AKI80組み立て終了後にチェックを行います。

付属のROM (ROM-W) をAKI80に挿入し、SIOAポートのCTSA、DCDA共にGNDと接続し、電源を立ち上げるとチェックモードに入ります。(下図参照)

電源は5V100mA程度の安定化されたものを御用意ください。+/-を間違えないように電源を投入すると、PIOBポート0、2~7が2秒間隔でHi/Loを繰り返します。テストでCN3の2~8ピンの電圧を測定してみてください。0V/5Vの繰り返しを確認できます。



この動作が確認できない場合、AKI80の半田不良をもう一度チェックする必要があります。確認できるまで先に進まないでください。

確認できない方へ。

コンピュータの動作は非常にデリケートです。例えて言えば非常に細い綱を駆け抜けて行く綱渡りのようなものです。ほんのさ細な半田くずでさえ嫌います。十分に注意してください。回路図、ピン配置図より Z84C015のピンから ROMのピン、Z84C015から RAM、Z84C015から HC32などピン間同志で導通及びショートをチェックをしてみてください。必ず不良は発見できるはずです。

1-2 ROMライターボード製作

ジャンパ線を取り付けます。J1～J14の14本あります。必ずメッキ線や抵抗の余りリードなどでジャンパー（基板をジャンプする）し、半田付けしてください。また極力基板と密着するよう取り付けてください。

抵抗を取り付けます。このキットでは2種類の抵抗を使用しています。カーボン抵抗と、金属被膜抵抗です。基板上では特にカーボン、金属の印はつけていませんが、同一値はありませんので、抵抗値より見分けてください。また金属被膜抵抗は、カーボン抵抗より色帯が1本多い5本あります。十分読み違えに注意してください。分かりにくかったらテストで確認しておきましょう。ちなみに向きはありません。

ICソケットを取り付けます。71055、MAX232、HC14のICソケットを取り付けます。切り書きマークを合わせ、取り付けるラインの内側に挿入してください。この際切り書きマークを合わせておき、後で向きを確認できるようにしておきます。

背の低いコンデンサを取り付けます。積層セラミック0.1 μ Fや電解コンデンサ10 μ Fを取り付けます。電解コンデンサには向きがありますので注意してください。基板には+マークがあります。

抵抗アレイ、タクトスイッチ、LED、トランジスタを取り付けます。抵抗アレイはコモンマークを合わせ半田付けしてください。タクトスイッチは基板の図の向きで取り付けます。LEDは基板外側が脚の長いほうです。トランジスタも、基板の図のように取り付けます。

DSUB25ピンを取り付けます。RS232C抜き差しの際、かなりのストレスが半田部に掛かりますので、基板にねじ固定したほうがよいでしょう。

背の高い電解コンデンサを取り付けます。向きに注意してください。

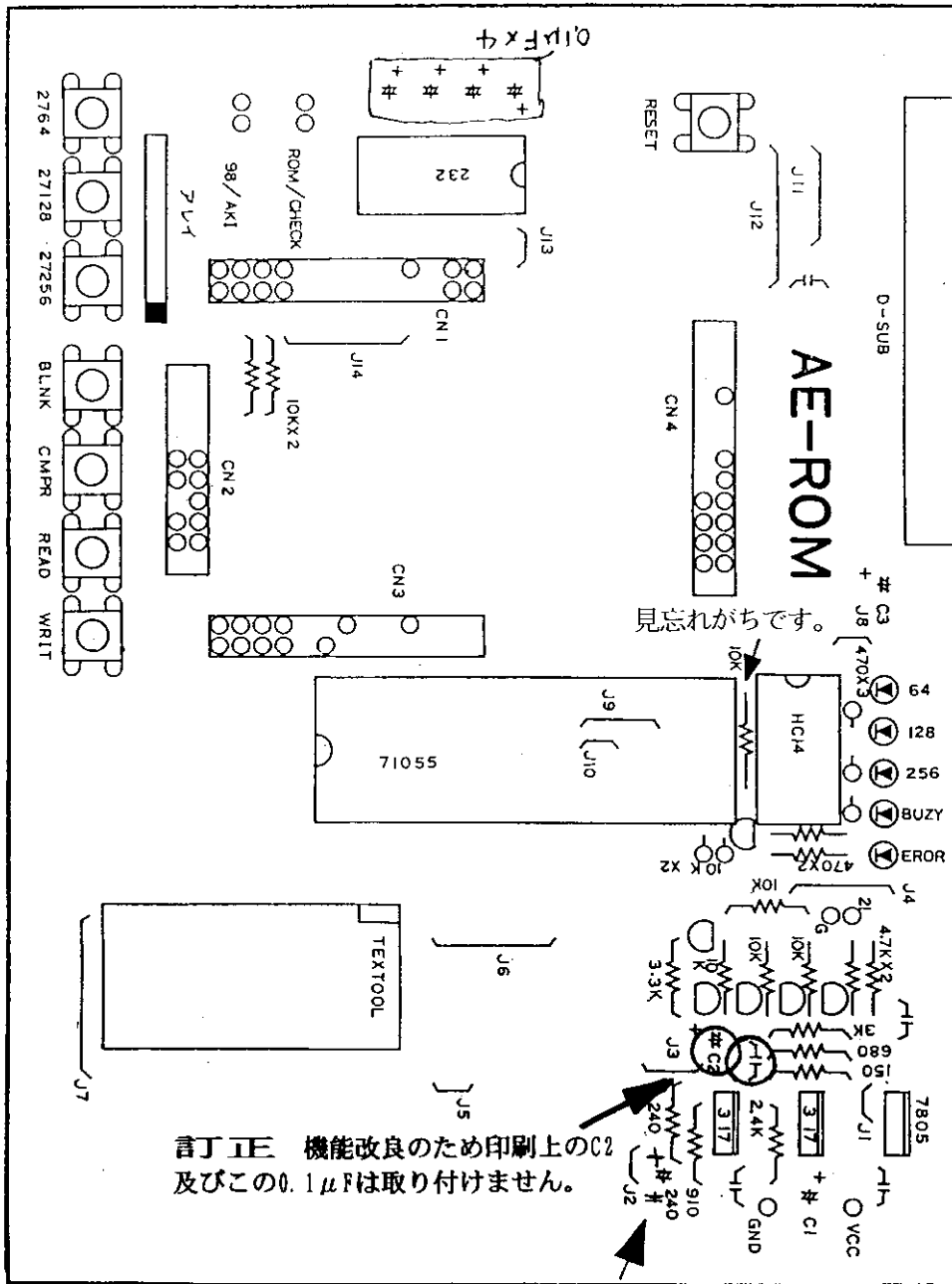
三端子レギュレータ7805、317を取り付けます。基板の図で、2本線側が、放熱部側です。7805はかなり発熱します。放熱器を取り付ける、ケースに取り付け放熱するなどの工夫をしたほうがよいかも知れません。

ここで、ミスがないか十分確認をしてください。AKI80の下にくる抵抗、ジャンパ、71055の下にくるジャンパなど付けわすれ、金属被膜の値ミスなどに気をつけてみてください。

AKI80を取り付けます。各コネクタの半田付けすべき端子は、○印で基板に記入してあります。すでにAKI80にはピンヘッドが取り付けられているはずですので、○印がすべて埋まるよう配置合わせをしたのち、半田付けします。

ICをソケットに挿入します。71055、MAX232、HC14の切り書きマークを合わせて挿入してください。

【部品配置図】



C1 33~220 μ F / 35V以上

C3 33~220 μ F / 6.3V以上
 表記なき \square は 0.1 μ F 積層セラミックです。
 表記なき #+ は 10 μ F / 10V以上の電解コンデンサです。
 表記なき \bigcirc は 2SC1815です。

注意 予告なく訂正済み基板に変更されることがあります。

言訂正 このコンデンサの+マークが基板上で逆についていました。お詫びして訂正させていただきます。

1-3 ゼロプレッシャーソケット取り付け

ゼロプレッシャーソケットは基板部品側にそのまま差し、取り付ければよいのですが、レバーの向きは使いやすい向きにさせていただいてOKです。

ケースなどにROMライタを入れる場合には工夫が必要です。何種類かの例を示しますので、工夫してみてください。

例1) ゼロプレッシャーソケットをげた上げする。

市販のラッピング用ソケットや、丸ピンソケットを何段か組合せ、基板よりげた上げします。

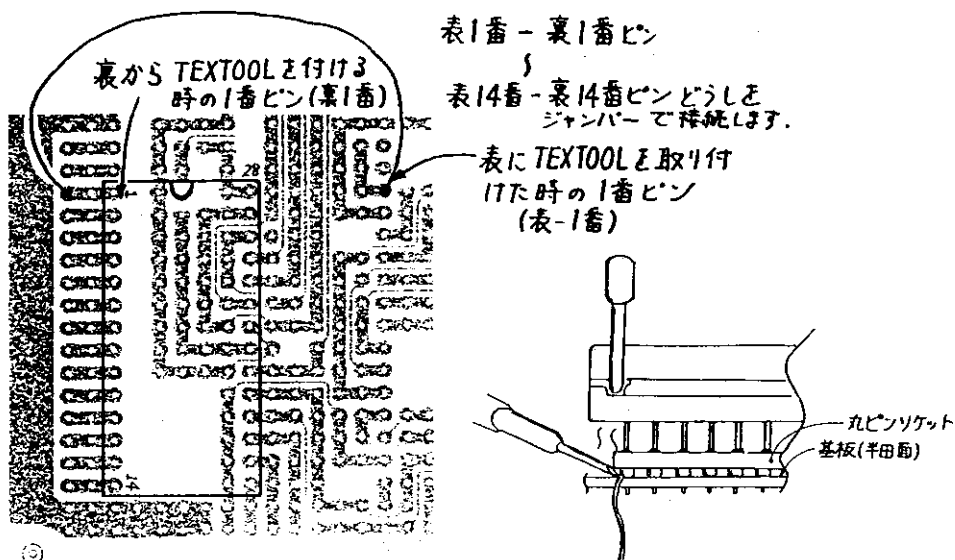
例2) 基板の半田面（裏面）に取り付ける。

基板の半田面になんとか工夫すれば取り付けられます。基板の裏をご覧ください。ゼロプレッシャーソケットがささるランド（半田付けするべきレジストのかかっている穴）と等間隔で離れたところにランドを用意してあります。ゼロプレッシャーソケットの右側（15～28PIN側）を軸に180度回転させ取り付け場所です。（下図参照）

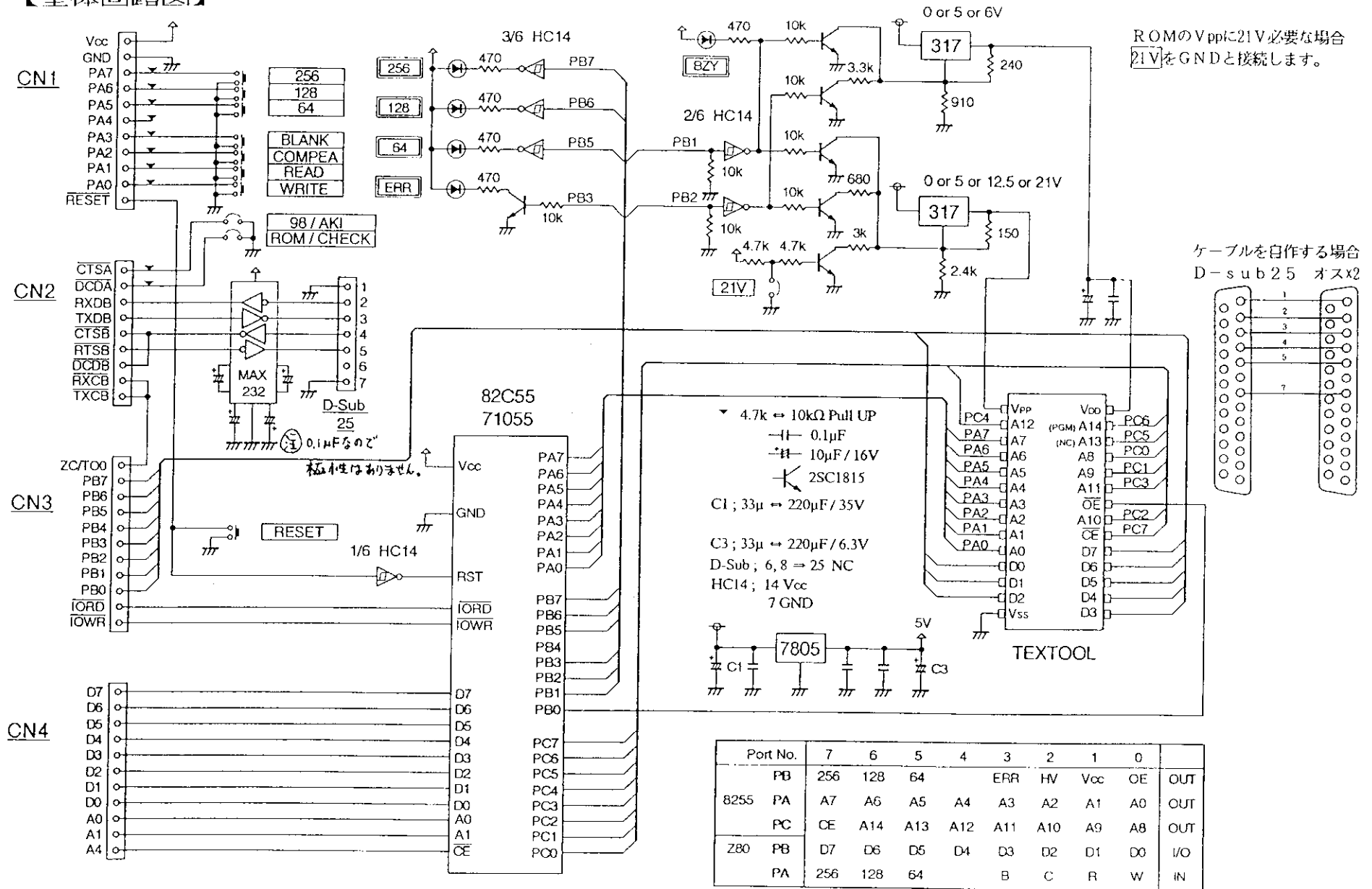
この場所にじかには半田付けできませんので、丸ピンソケットなど裏面からでも半田付け可能なものを中間に介して半田付けします。部品面よりビニール線を使用し、ゼロプレッシャーソケットの左側（1～14PIN側）の未配線部分をジャンパします。この場合タクトスイッチも逆に付けたほうがよいかも。

注意

ゼロプレッシャーソケット取り付けランドには32ピン分のランドが用意してあります。上例2個づつは余りますので注意してください。利用したい方は御自由に。



【全体回路図】



1-4 電源について

電源には1.5V～1.8V/200mAの程度の物が必要です。安定化してある必要はありませんのでACアダプタ等ある場合は利用してください。

ROMの書き込み時にはROMのVcc、Vppとして6V及び12.5Vが必要になります。これは書き込み時のみ317が作り出しROMに供給するため、電源には最低でも1.5V必要な訳です。

オプション扱いですがROMのVppが21Vの物も書き込みたい場合には、別に24V電源も必要です。DC/DCコンバータや、スイッチング電源など利用してもよいと思います。この場合24Vを電源端子に供給すると、7805の放熱が耐えきれなくなりますので、2個の317の入力ピンを基板より浮かし、このピンにのみ24Vを供給してください。そして電源入力端子には、8V以上を加えればOKです。

1-5 チェック及びモードについて

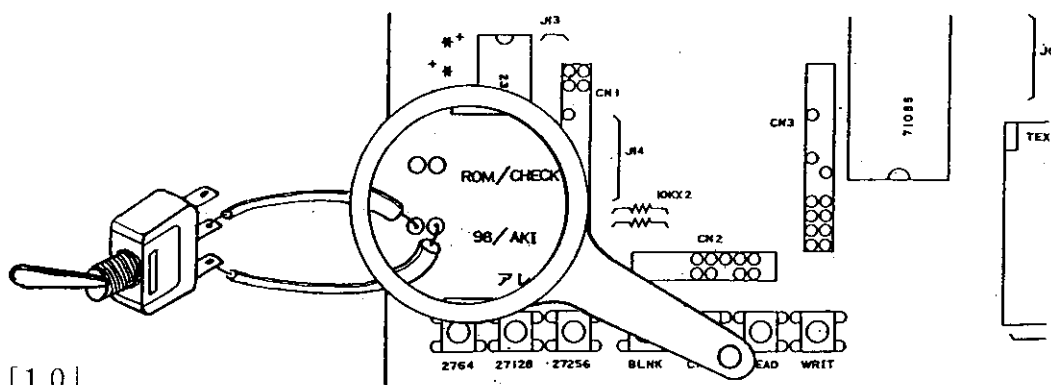
ROMライタをチェック/ファイル転送モードにします。モードSWはMAX232CとタクトSWの間にあります。チェック/ファイル転送モードは初期時にしか使用しませんので、とりあえずジャンプしておいてください。

モード	ROM/CHECK	98/AKI
98制御 ROMライタ	オープン	オープン
AKI制御ROMライタ	オープン	ショート
チェック/ファイル転送	ショート	ショート

上記に示す電源を接続してください。正常に動作していれば、AKI80チェックと同じく、PIOBがHi/Loを2秒間隔でくり返すのですが、71055の全ポートもHi/Loをくり返します。これによりBZYを除くLEDも点滅します。TEXT00Lの2～27ピンもHi/Loをくり返します。テストにて確認してください。確認が取れば完成です。

もし確認が取れない場合で、PIOB、71055共にだめな場合、電源の入力電圧、7805の出力が正しく5Vになっているか、またモードが正しいか、調べてみてください。PIOBはOKで、71055はだめな場合、AKI80のデータバス(CN4)から71055の配線が異常です。

このモードは -2- 98用ソフト転送でも使用しますので、-2-の章が終了後、以下のようにSWに付け替え、モードを切り替えられるようにしてください。98制御、AKI制御のどちらか専用にする場合は、上記のモード表を参照し配線してください。





高速、+5V、0.1 μ F CMOS RS-232ドライバ/レシーバ

ADM222/ADM232A/ADM242

特長

- 200kB/秒の転送レート
- 小容量 (0.1 μ F) 値のチャージ・ポンプ用コンデンサ
- +5V単一電源動作
- EIA-232-EおよびV.28規格に適合
- 2個のドライバと2個のレシーバ
- DC-DCコンバータを内蔵
- +5V電源で±9Vの出力幅幅
- ±10Vのレシーバ入力レベル
- MAX222/MAX232A/MAX242とピン・コンパチブル

応用

- コンピュータ
- 周辺機器
- モデム
- プリンタ
- 計装

概要

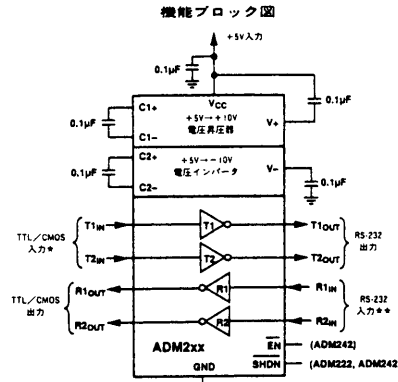
ADM222、ADM232A、ADM242は最高200kB/秒までの転送レートで動作する高速RS-232ライン・ドライバ/レシーバ・ファミリです。+5V単一電源で動作し、小容量 (0.1 μ F) の外部コンデンサを高性能なチャージ・ポンプ用に使用することで、RS-232のバイポーラ・レベルを生成します。各製品は2個のRS-232ドライバと2個のRS-232レシーバを備えています。

これらの製品は、ローパワーCMOSと高速バイポーラ回路を組合わせた最新のBiCMOSプロセス技術で製造されています。これにより、無負荷時電源電流を5mA以下まで抑えながら最高200kB/秒の転送レートを実現しています。

ADM232AはAD232およびADM232Lとピン・コンパチブルの改良製品です。パッケージは16ピンのDIP、スキニー/ワイド幅SOICパッケージが用意されています。

ADM222にはデバイスをディセーブルする時に使用するシャットダウン (SHDN) 機能があります。これにより電源電流を0.1 μ Aに減少できます。シャットダウン中、すべての送信/受信機能はディセーブル状態になります。パッケージは18ピンのDIPとワイド幅SOICパッケージが用意されています。

ADM242には、シャットダウン (SHDN) 機能とイネーブル (EN) 機能があります。このシャットダウン機能で電源電流を0.1mAに抑えることができます。シャットダウン中、トランスミッタはディセーブル状態ですが、レシーバは通常通り動作し続けます。またイネーブル機能により、レシーバ出力をディセーブルできます。つまり共通のバスで動作できる機能があります。パッケージは18ピンのDIPとワイド幅SOICパッケージが用意されています。



※ TTL/CMOS入力に400nFのプルアップ抵抗を内蔵
 ※ RS-232入力に5k Ω のプルダウン抵抗を内蔵

オーダー・ガイド

モデル	温度範囲	パッケージオプション
ADM222AN	-40°C ~ +85°C	N-18
ADM222AR	-40°C ~ +85°C	R-18W
ADM232AN	-40°C ~ +85°C	R-16
ADM232AARN	-40°C ~ +85°C	R-16N
ADM232AARW	-40°C ~ +85°C	R-16W
ADM242AN	-40°C ~ +85°C	R-18
ADM242AR	-40°C ~ +85°C	R-18W

μ PD71055C/G (TMP82C55AP)

パラレル・インタフェース・ユニット

μ PD71055は、マイクロコンピュータ・システム用のプログラマブルなパラレル・インタフェース・ユニットです。3つのI/Oポートを使って、基本的な入出力ポート動作からハンドシェイクによる高度な入出力動作まで行うことができます。

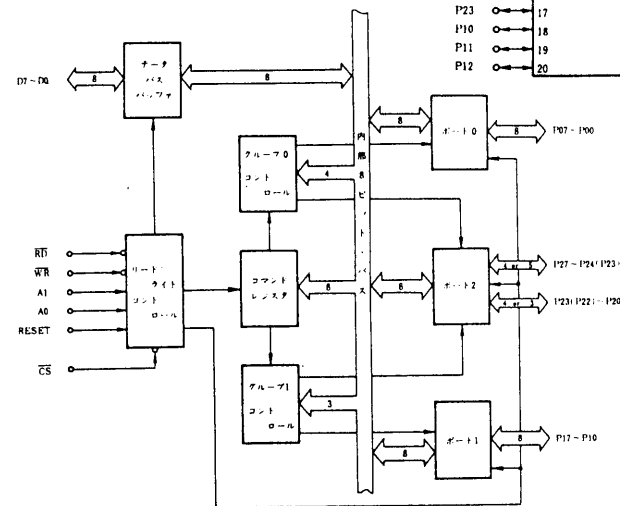
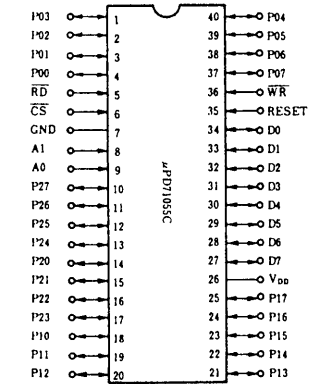
CMOS構造で作られているため低消費電力となっています。

特徴

- 3組の8ビットI/Oポート
- プログラマブルな3つの動作モード
- ビット操作命令
- 標準マイクロコンピュータとコンパチブル
- 8MHz動作
- CMOS
- 単一電源
- 40ピン・プラスチックDIP (600 mil)

μ PD71055C (P40C-100-600A)

端子接続図 (Top View)



■- 2 - 98ソフト転送■

98用のソフトはROMライターよりRS232Cにて転送します。

2-1 9801セットアップ

まず起動ディスクを作成します。

フロッピーベースの方はフォーマット時にシステムを転送して下さい。そしてMSDOSよりRSDRV. SYS及び、SPEED. EXE (COM) をコピーします。
ハードディスクの方は・・・・・・詳しくはMSDOSのマニュアルを読んで下さい。

ソフト転送時のみ、MSDOS標準付属のRSDRV. SYS及びSPEED. EXEが必要です。

CONFIG. SYSに次の行を追加して下さい。(CONFIG. SYSが無い方はエディタで作成しておきます。)

```
DEVICE=RSDRV. SYS
```

デバイスドライバとして登録するため、ここで1度98のリセットスイッチを押して下さい。再びシステムが読み込まれる時に以下のコメントが表示されればOKです。

RS232Cインターフェイスが使用可能です。

SPEED. EXEを実行します。以下の行をタイプし実行して下さい。

```
SPEED R0 4800 B8 PN S1 NONE
```

最近のSPEED. EXEはメニューが付属しています。メニュー起動には
SPEED とタイプし実行します。これにより以下のように設定すればOK。

```
RS232C-0
ボーレート (BPS)    4800
キャラクタ長        8ビット
パリティチェック   パリティ無
ストップビット長   1ビット
Xパラメータ         無
```

ニブレイク

98ソフトはROMライターから、見かけ上テキストファイルとして転送されてきます。実は実行可能なAK. COMと言うバイナリファイルなのですが。

9801のSIOには8251というかなりクラシックなものが使われています。COPYコマンドによる転送では98、ROMライター通信時と同じ9600ボーでは取りこぼしが発生します。そこで98ソフト転送時のみ4800ボーで行います。

2-2 ROMライターセットアップ

ROMライターはチェックモードのまま電源をいれないでください。
9801とRS232Cノーマルケーブルで接続します。98は標準RS232Cポートを使用します。

注意

一般にコンピュータ同志を接続する場合、RS232Cケーブルはクロスタイプを使用しますが、ROMライターではノーマルタイプ、ストレート結線を使用するようにしました。これはクロスタイプには結線により色々な種類があるため、混乱を招く恐れがあり、また一般にクロスタイプはノーマルタイプよりやや値段も高いようです。ノーマルタイプはモデムでも利用されるため安価であり、結線も一種類のみです。

9801セットアップが終了した後、ROMライターの電源をいれて下さい。基板チェック時と同じく、BZYのLED以外が2秒間隔で点灯していますね。

2-3 ソフト転送

98側でコピーコマンドを実行します。以下のようにタイプし実行して下さい。

```
COPY AUX AK.COM
```

続いて、ROMライターからソフトを送信します。

ROMライターの「WRITE」スイッチを押して下さい。

これで、ROMライターのBZYのLEDが点灯します。もし点灯しない場合は98のセットアップがおかしいか、RS232Cケーブル接続不良です。もう一度確認して下さい。

正常に動作していれば、約1分後にBZYのLEDが消灯し98のディスプレイに、

```
1個のファイルをコピーしました。
```

と表示されます。DIRコマンドでファイルの存在及びファイルサイズを確認して下さい。

```
AK.COM      25078      (COPYAを使うと25079)
```

もしファイルサイズが違う場合は、転送ミスです。AK.COMを必ずデリートし、ROMライターのリセットを押し、もう一度コピーコマンドを実行して下さい。

また2分たってもコピーが終了しない場合、98をリセットするしかありません。RSDRV.SYSやSPEED.EXEを確認し、もう一度最初(2-1)からやり直して下さい。

ニブレイク

当社はハード屋であり、ソフトは今まで取り扱ったことが無いため、フロッピーの管理体制がありません。(壊れやすいとか、磁石に弱いとか、5inch、3.5inchなど)そこで考え出されたのがROMにソフトを埋め込む新方式です。始めはとまどうかも知れませんが、

メディアの選択ミスや、破壊など無く、信頼性は高まると思います。またキットの低価格化にもつながりました。

2-4 ファイル解凍

AK.COMはLHAを利用した自己解凍型圧縮ファイルが組みこまれています。これを実行することによりROMライタの98用ソフトがファイルとして誕生します。以下の文をタイプし実行して下さい。

AK

これにより二種類のファイルが解凍されます。DIRコマンドにて確認して下さい。

```
AKIROM . EXE    24942
HEXUTL . COM    10820
```

2-4-1 ファイルの簡単な紹介（詳しい使い方は後述）

1) AKIROM.EXE

このファイルはROMライタ制御のプログラムです。すべての制御はこのプログラムで行います。

2) HEXUTL.COM

インテルHEXファイル用ユーティリティプログラムです。バイナリファイルをインテルHEXファイルに変換する機能、長いインテルHEXファイルを2764や27256用に分割する機能、インテルHEXの絶対番地にオフセットを与える機能などがあります。

ニブレイク

AK.COMはフリーソフトウエアであるLHAを利用させていただいております。LHAは吉崎 栄泰氏のアーカイバソフトで、日本を代表する圧縮解凍プログラムです。大変有用なソフトを公開して頂き、この場にてお礼をさせていただきます。

技術的なこと。

AK.COMはバイナリファイルですが、テキストファイルとして転送するため、ファイル中にターミネイト文字に相当する1AHがありません。本来あるはずの1AHを無くしてしまっています。

AK.COMでは最初のルーチンで、あるべき場所に1AHを書戻してから、LHAの自己解凍型ファイルを中間ファイルとして作り、それを子プロセスとして実行しています。そのためAドライブのルートディレクトリ(A:*)にCOMMAND.COMが必要です。またディスクには最低110Kバイト以上の空き容量が必要です。

■-3- AKIROM. EXE リファレンス

3-1 初期手順

AKIROM. EXEは、フルスクリーンROMライターコントローラーです。ハードディスクなどでサブディレクトリを作っている方は、アセンブラなどの開発ソフトと同じディレクトリにおいて下さい。

ROMの書込みデータには、インテルHEXフォーマットが使えます。バイナリ型式では、前もってHEXUTL. COMにてインテルHEXファイルにしておいて下さい。予め書込みデータは用意する必要があります。

書込めるEPROMは2764、27128、27256のノーマルタイプ、CMOSタイプで、書込み電圧は標準で12.5Vおよびオプションで21Vも可能です。2864の場合は5Vです。書込み電圧はROMライター、ボード上で設定します。ボード上の21V端子を開放のままでは12.5V。GNDと続して21Vです。くれぐれも間違えないように。2864の場合はいずれでも問題ありません。

ROMライター側はRS232Cで98と接続し、モード設定でROMライターモード（98制御）にし、電源を入れ直すか、ROMライターリセットを押し使用して下さい。ROMライターモード（98制御）ではリセットを除く全てのタクトSWは無効になります。

注 意

ROMライター電源投入時にはROMはTEXTTOOLに挿入しないで下さい。不本意に書き込んでしまう恐れがあります。
またBZYのLEDが点灯時の抜き差しも厳禁です。BZYのLEDが点灯している間はROMに電源が供給されているためです。

注 意

98のコマンドラインとしてマニュアル上では全角文字で記述していますが、必ず半角文字でタイプして下さい。
また全て大文字小文字の区別はありません。

ニブレイク

ROMライターモードでは98制御と、AKI制御はSW切り替えにより移行できますが、他のモードへは、リセットしないと移行できません。この時点でチェックモードにする人はいないと思いますが。

3-2 コマンドラインオプション

標準的な設定ではコマンドラインオプションは必要ありません。以下をタイプし実行して下さい。

AKIROM

コマンドラインオプションには以下のものがあります。

AKIROM [-64, -128, -256, -2864] [-MXX] [DATA DIR]

1) [-64, -128, -256, -2864]
ROMの容量を予め設定します。-64又は-128又は-256又は-2864の内一つが有効です。指定しない場合、AKIROM起動後に選択します。

2) [-MXX]
XXには00~FFが入ります。内部メモリの初期化定数です。AKIROM.EXEには内部にROM容量と同じ容量の内部メモリがあります。このメモリにインテルHEXファイルをロードする訳ですが、インテルHEXのデータがない部分の値を何にするか決定するものです。焼きこみの際、ROMのデータ空き部分もこの値が入ります。デフォルト(指定しない場合)はFFで、ROMのデータは消去状態で全部がFFですので、データがFFの場合書き込まなくても同じです。これにより書き込みタイマをキャンセルして書き込みの高速化をはかれます。この値の設定が必要がある場合は、CPU開発時に暴走しては困る時など、NOP命令(Z80では00、8086では90)で埋めておけば防げることもあります。

3) [DATA DIR]
データディレクトリが他のディレクトリにある場合指定します。指定しない場合はカレントディレクトリにインテルHEXファイルがあるものとします。

例) 27256の焼きこみで、AドライブのZ80というディレクトリにインテルHEXファイルがある場合、

```
AKIROM -256 A:¥Z80
```

ROM容量が決定している、データディレクトリが決定しているなど、ROM.BATを上記の内容にすればBEST。

```
A) TYPE ROM.BAT
AKIROM -256 A:¥Z80
```

ニブレイク

一般にROMの書き込みのことを、焼き込む、トーストするなどと言われます。由来は・・・・・・知りません。

3-3 オペレートコマンド

コマンドラインにてROM容量を指定していない場合、まずROM容量を設定します。

[↑] [↓] キーにて選択しリターンキー [RET] で決定します。

2764	↑
27128	
27256	
2864	↓

決定後オペレート コマンド セレクト [OPERATE COMMAND SELECT] 画面に入ります。コマンドラインで設定してあるとROM容量はパスします。サンプル画面を参照してください。

画面の見方

アクティブ状態のセレクトウインドは黄色の反転スクロールバーで表示されます。決定すると緑の反転バーに変わります。

HEXファイル名及び、スタートストップアドレスは、アクティブ状態で、黄色のカーソルブリンク、決定後に白になります。

サンプル画面

AKI-80 ROM WRITER ver 1.2		秋月電子通商
ROM TYPE	OPERATE COMMAND SELECT	DATA DIR = A:¥
2764	Load HEX File	HEX FILE = *.HEX
27128	Write & Transmit	
27256	Write from AKI-RAM	
2864	Transmit to AKI-RAM	
	Read & Receive	
	Read to AKI-RAM	
	Compare HEX File	
SIRT ADS	Compare AKI-RAM	
STOP ADS	Blank Check	
	Rom Type	
0000 H	Start Stop Adress	
7FFF H	Save HEX file	

[ESC] to EXIT

3-4 終了方法

[OPERATE COMMAND SELECT] がアクティブのとき [ESC] キーを押すと終了します。

[ESC] キーは全てのアクティブウインドから一つ手前のウインドに抜け出すため使用されます。

注 意

コマンドラインオプションでROM容量を選択していない場合は、最初に [ROM TYPE] ウインドがアクティブになりますが、ここから終了させるためには一度 [ESC] を押し、[OPERATE COMMAND SELECT] に抜けだし、もう一度 [ESC] を押してください。

3-5 メイン操作 [OPERATE COMMAND SELECT] 説明

基本的なキー操作

[↑] [↓] で選択し、[RET] で決定します。(RET) はリターン、改行の意)

3-5-1 【Load Hex File】

HEXファイルを98内部メモリに取りこみます。(HEX File=98メモリ)

これを選択すると、[HEX FILE = *.HEX] がアクティブになります。

* (ワイルドカード) を消しファイル名を入力するか、さらに [RET] を押せば、ワイルドカードに適合するファイル名が一覧表示されます。

1) ファイル名入力

ラインエディタ型式で入力します。DOSでファイル名として許されていない文字や、全角文字は入力できません。入力後 [RET] で決定で、上記のデータディレクトリよりファイルを検し、内部メモリにロードします。

2) ワイルドカード選択

ファイル名にワイルドカード (*、?) が含まれていると、ファイル選択画面にワイルドカードに適合するファイル名が一覧表示されます。表示されるファイルは日時の新しい順 (ファイルのタイムスタンプの新しい順) に最大30個までです。必ず最新のファイルが先頭にきますので、開発時にはかなり有効な機能です。

[↑] [↓] [←] [→] キーで選択し、[RET] を押すとファイル名が [HEX FILE =] にコピーされ、さらに [RET] で内部メモリにロードします。

ニブレイク

98、ROMライタ間はポーレート9600ボーで転送されます。これは1ビットの転送に1/9600秒を要します。1バイト(8ビット)の転送ではスタートビット及びストップビットが付加されるため10ビットを転送することになります。また27256は32Kバイトの容量があり、つまり転送に要する時間は

$$1/9600 \times 10 \times 32768 = 34.13 \text{ (秒)}$$

の時間が最低でも必要です。

3-5-2 【Write & Transmit】

98内部メモリをROMライターに転送し、ROMに書き込みます。(98メモリ→ROM)

未書込のROMを正しくTEXTTOOLに挿入し、[RET]するとROMのブランクチェック、データ転送、書き込み、コンペアチェック、チェックサムを一気に行います。不良があると処理を中断します。

ROMライター側はROMの種類とBZYのLEDを点灯させ、終了後BZYのLEDは消灯します。不良があるとERRORのLEDが点灯します。ERRORのLEDは次の処理が行われるまで保持します。

通常書き込み処理では、[Load HEX File]と[Write & Transmit]のみで事足ります。

3-5-3 【Write From AKI-RAM】

ROMライター内蔵のRAMからROMに書き込みます。(AKI-RAM→ROM)

AKI80のRAMは書き込みROMのバッファとして、全ての領域が確保されています。このため一度転送してしまえば、2個目以降の書き込みにはRAMからデータを読みだし転送時間がなくなり書き込み時間の高速化がはかれます。

3-5-4 【Transmit to AKI-RAM】

98のメモリからROMライターのメモリに転送します。(98メモリ→AKI-RAM)

書き込みは行わない、データの転送のみを行います。Write From AKI-RAMと組み合わせれば、ブランクチェックを行わないで書き込みができますので、ROMエミュレータなどの書き込みに有効です。(ROMエミュレータの場合、Vppには注意してください。)

3-5-6 【Read & Receive】

すでに書き込み済みのROMを読みだし、98メモリに転送します。(ROM→98メモリ)

TEXTTOOLのROMデータを読みだします。98のメモリに転送すると同時にROMライター内部のメモリに取りこみます。ROMデータの保存などに有効です。

3-5-7 【Read to AKI-RAM】

すでに書き込み済みのROMを読みだし、ROMライターの内部メモリにのみデータを取りこみます。(ROM→AKI-RAM)

Write From AKI-RAMと組み合わせれば単純なROMコピーなど98を経由せずに行えます。

3-5-8 【Compare HEX File】

98メモリにロードされているHEXファイルとROMのデータを比較（コンペア）します。（98メモリ=?ROM）

すでに書き込んだROMとHEXファイルが同じものか、確認する時有効です。

3-5-9 【Compare AKI-RAM】

書き込んだROMとROMライター内蔵のRAMのデータを比較します。（ROM=?AKI-RAM）

ROMコピー時に正しくROMが書き込まれているかチェックする時に有効です。

3-5-10 【Blank Check】

ROMが空ROM（未書込）かチェックします。（ROM=全FF）

ROMが消去されているか確認したい場合に有効です。ROMライター上でのみのコピー時には必ず行ったほうがよいでしょう。

3-5-11 【Rom Type】

ROMの容量を選択します。

前途参照。（3-3）

3-5-12 【Start Stop Address】

ROMの容量範囲内で、処理する範囲を決定します。

書き込み、読みだし、及び全てのチェックの有効範囲を決めます。これにより任意のバイトのみ書き込みたい時や、書き込みデータサイズがわかっていて処理を早く終了させたいときに有効です。

		最大有効範囲		
例) ROMの番地0010Hから001FHまでの16バイトの書き込み	<table border="1"><tr><td>0010 H</td></tr><tr><td>001F H</td></tr></table>	0010 H	001F H	2764 0000H~1FFFH
0010 H				
001F H				
		27128 0000H~3FFFH		
		27256 0000H~7FFFH		
		2864 0000H~1FFFH		

Start Stop Address を選択すると【STAT ADS STOP ADS】ウインドがアクティブになります。0~Fのキーのみが有効です。必ずHEX 4桁で入力してください。

注 意

例えば0000H番地の1バイトのみの書き込みの場合、スタート、ストップアドレス共に0000Hになります。ストップアドレスは何バイト処理するかではなく、終了時のアドレスです。間違えやすいので要注意です。

3-5-13 【Save HEX File】

98メモリのデータをHEXファイルとして書き出します。(98メモリ→HEX FILE)

ROMのデータをバックアップ、解析などに有効です。

選択すると、[HEX FILE = *.HEX] ウィンドが有効になります。使用法は3-5-1 [Load HEX File] と同じです。

3-6 エラーコメント一覧

ROMライター動作時にエラーを起こすとコメント欄にエラーが表示されます。エラーを詳しく説明します。

1) 送信に失敗しました。

主にROMライターが接続されていない、電源が入っていない、RS232Cケーブルが断線しているなどの場合に表示します。実際には232ポートに何も接続してない場合です。

2) 受信に失敗しました。

主にROMライターのモードが違う、RS232Cのケーブル断線、ROMライター以外のものが接続されている場合に表示します。実際には何もROMライターから返答がない場合です。

3) AKI-80ではありません。

主にROMライターではないものが接続されている、受信エラー時、ROMライターが暴走してしまった場合などに表示します。実際にはROMライターから期待される返答と違うものが入力された場合です。

4) アドレス範囲以外です。

スタートストップアドレス設定時に設定できる範囲以上や、ストップアドレスがスタートアドレスより前にある場合に表示します。

5) ファイルが見つかりません。

ファイルをロードする際に指定のファイルがない場合、またロード、セーブ共にワイルドカードに適合するファイルがない場合に表示されます。

6) HEX FILEではありません。

ファイルロード時に、インテルHEXフォーマット以外のファイルをロードしようとする则表示します。

7) ファイルがオープンできません。

ディスクエラー、DOSエラー時に表示されます。

8) BUFF OVER FLOW max 7FFFh

ロード時にHEXファイルが絶対番地7FFFhを超えている場合に表示します。付属のHEXUTLで分割処理するか、又は書き込むHEXファイルを32Kバイト以内にしてください。

9) ROM 容量以上1続けますか?

ROM容量が2764、27128、2864の場合、HEXファイルの絶対番地がROM容量設定以上に達した場合表示し、確認を求めます。

10) 既存ファイル名です。書換えますか?

HEXファイルのセーブ時に同名のファイルがあると表示され、確認を求められます。

11) ドライブ×:は××です。中止(A),もう一度(R)

色々なディスクエラー時に表示し、処理を求めます。

12) その他のエラー

ROM不良や、書き込みエラー、転送エラーなど全てのオペレートコマンドでのエラーも表示します。

ニブレイク

98とROMライタの通信は全て処理データ分だけのバイナリデータを転送しています。また転送終了後には必ずチェックサムを取り信頼性を向上しています。表示されるチェックサムは単純和です。

注 意

ディスクエラーの表示は赤色で表示します。そのためノートパソコンの場合多少見にくいかも知れません。

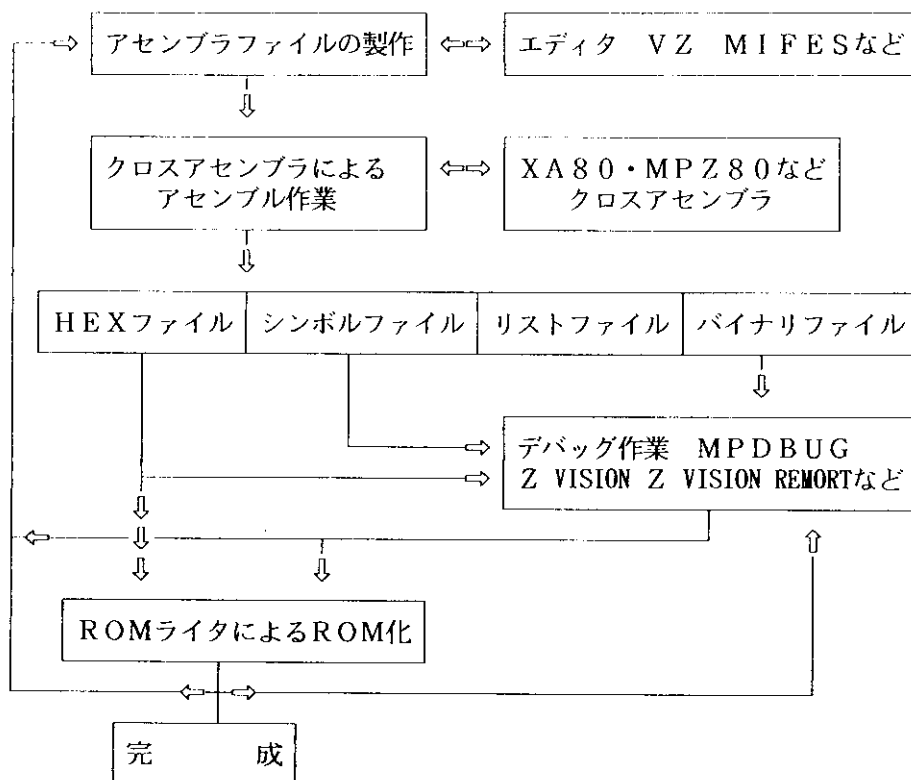
ROMライタのBZYのLED点灯時は[ESC]キーによる処理中断はできません。どうしても止めたい場合は、ROMライタリセットSWによる中断をしてください。特に問題はありません。

ROMライタ、AKIROM. EXE共にできるだけのエラー処理対策がしてありますが、シリアル通信という過程上、絶対ないとは言い切れません。何か動作がおかしいと思ったら、ROMライタのリセットを押し、AKIROM. EXEをいったん終了させもう一度立ち上げてください。

AKIROM. EXEで、ディスクプロテクトされているフロッピーにデータをセーブしようとする、AKIROM. EXEは強制的に終了します。

■ 4- ROMライタの色々について

4-1 98上でのAK180開発の実際



- 1) エディタでアセンブラのソースファイルを作ります。
- 2) クロスアセンブラでアセンブラソースファイルをアセンブル（機械語に直す）します。
- 3) 必要があれば、デバッガでデバッグ（虫取りの意味）をします。
- 4) ROMに焼き込み実際に動かしてみます。
- 5) 一発で動くことはまずないので、1～4を何回か繰り返し完璧なものにします。

アセンブラソースの書き方までは説明し切れるものではありませんので、クロスアセンブラのマニュアルや、参考文献、付属のZ84C015、ASMなど参考にしてください。

参考文献

Z80ファミリ ハンドブック CQ出版

この本はZ80CPUやZ80ファミリのIOについて詳細に述べられています。AK180はZ80ファミリで構成されているので、使い方、動作などこれ1冊で分かり、便利です。

4-2 ROMライター オンボードコントロール

ROMライターのみでも4種類の動作が可能です。
モードはROMライターAKI制御にしてください。

4-2-1 ROMの容量を選択します。

ボード上タクトSWで、2764、27128、27256のうち1つを押してください。
LEDでも容量が示されます。電源投入時は必ず27256が選択されています。

4-2-2 オペレート

1) BLNK

ブランクチェックです。ROMが未書き込みかチェックします。

2) READ

ROMデータを内部のRAMに読み込みます。

3) WRIT

内部のRAMからデータをROMに書き込みます。

4) CMPR

コンペアチェックです。書き込んだROMと内部RAMのデータを比較します。

4-2-3 ROMコピー時の実際

- 1) 書き込み済みのROMと未書き込みのROMを用意します。
- 2) 書き込み済みのROMをTEXTTOOLに正しく挿入し、READスイッチを押します。BZYのLEDが点灯し、終了後に消灯します。これで内部RAMにデータが取りこまれました。
- 3) 未書き込みのROMをTEXTTOOLに挿入します。BLNKスイッチを押し、未書き込みか確認します。BZYのLEDが点灯し、終了後に消灯します。もし書き込み済みのROMだとERRORのLEDが点灯します。
- 4) WRITスイッチを押し、ROMにデータを書き込みます。BZYのLEDが点灯し、終了後に消灯します。もし正しく書き込めないとERRORのLEDが点灯します。
- 5) CMPRスイッチにて書き込まれたデータが正しいかチェックしておきます。
- 6) 2個以上のコピーの場合、3～5のみ行ってください。

4-3 LEDについて

ボード上には5個のLEDが装備されています。98制御、AKI制御共に表示します。

BZY) ビジー 稼働中を示します。点灯時のROMの抜き差しは厳禁です。
ERROR) エラー 処理中にエラーが発生すると表示します。次の命令を受けるまで保持します。

64、128、256) ROM容量を表示します。

4-4 TEXTTOOLについて

ROMライターにはなくてはならない部品、抜き差し自由なゼロプレッシャーソケットにはエイリス製の物を使用しています。このキットの中で一番高価な部品です。(キット価格の1/4強を占める)エイリス製のゼロプレッシャーソケットは非常に丈夫で、また何よりいいところは、一般の丸ピンソケットに挿入できることです。いくら丈夫とはいえ、やはり何千個と抜き差ししていると劣化してくるものです。この場合の交換や、下駄上げなど丸ピンソケットの挿入できる魅力は大です。

4-5 内部RAMについて

内部RAMは全てROMのバッファとして確保しています。AKI80には256KのRAMが乗っているのですが、これは27256のROMと同じデータ量です。これを丸々バッファに割り当てているため、ROMライターのプログラムでは、RAM上にプログラムデータを取りません。またスタックも使用していませんのでROMバッファとしてのRAMを破壊することはありません。

RAMはバックアップが可能です。この機能を使えばバッファは電源を切ってもデータが保持されるため、同じものを書き込む場合はデータを再ロードする必要がありません。

ニブレイク

ROMの消去にはROMレーザーという紫外線発生装置で行います。紫外線発生には殺虫灯(蛍光灯の一種)が使えます。殺虫灯の出す紫外線の直視は目に悪いので、光が漏れない箱にいれ、自作のROMレーザーの出来上がり。20~60分で消去できます。

書き込んだ大切なROMには遮光シールを張り保護しましょう。

訂正

BZYはBUSYの誤りです。

HEXUTL (Ver 2.00) の使用方法
----- Apl. 11. 1992 By (有) 秋月電子通商 -----

1、HELUTL.COMとは、

HEXUTL は、アセンブラ、コンパイラなどの Binary または、Intel HEX ファイルを ROM Writer などを使用する Intel HEX ファイルに変換するファイルコンバータです。

但し、Ver 2.00 で扱えるデータサイズは、65536 バイトまでです。

ファイルコンバータ機能には、

```
mode 0 : Binary file --> HEX file  
mode 1 : Binary file --> Binary file  
mode 2 : HEX file    --> HEX file  
mode 3 : HEX file    --> Binary file
```

以上4つの動作モードがあります。これらの動作モードは、入出力ファイルの拡張子が '.HEX' か、'.BIN' かを見て自動判別していますが、それ以外の拡張子の場合、オプションパラメータを用いて機能を指定することができます。

この他、8 BIT データバスの ROM を使用して、16, 24, 32, バス対応のデータを作成する機能や、ROM 容量よりも大きなデータファイルを分割して、それぞれの ROM に対応したデータファイルを作成する機能があります。

2、基本機能を利用する場合、

【1】 Binary ファイルを Intel HEX ファイルに変換する場合

Binary ファイルが、'A:¥PROG¥LEDON.BIN' の場合、

```
A:¥) HEXUTL A:¥PROG¥LEDON.BIN
```

と入力することにより、Intel HEX ファイル 'A:¥PROG¥LEDON.HEX' が作成されます。

【2】 Intel HEX ファイルを Binary ファイルに変換する場合

Intel HEX ファイルが、'A:¥PROG¥LEDOFF.HEX' の場合、

```
A:¥) HEXUTL A:¥PROG¥LEDOFF.HEX A:¥PROG¥LEDOFF.BIN
```

と入力することにより、'A:¥PROG¥LEDOFF.BIN' が作成されます。

3、入力書式

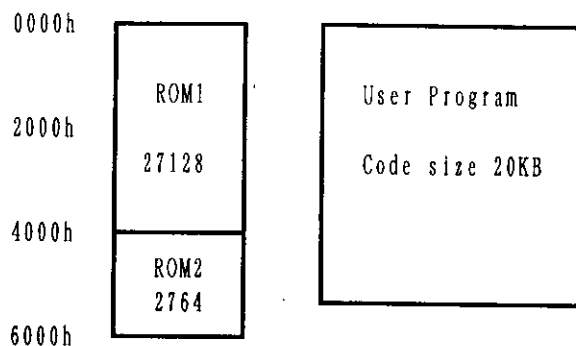
【書式】 HEXUTL [option] file_name_1 [option] [file_name_2] [option]

【パラメータの意味】

- file_name_1 入力ファイル名、省略することはできません。
入力ファイルの拡張子により以下のモードが自動選択されます。
- 拡張子省略時 mode 0 が選択されたものと解釈します。
拡張子 '.BIN' が自動的に付加されます。
- .BIN mode 0 が選択されたものと解釈します。
.HEX mode 2 が選択されたものと解釈します。
- [file_name_2] 出力ファイル名、省略することができます。
省略された場合、入力ファイル名と同じになります。
出力ファイルの拡張子により、入力ファイルの拡張子と組み合わせ
以下のモードが自動的に設定されます。
- 拡張子省略時 入力ファイルの項目で設定されたモードと同じ。
拡張子 '.HEX' が自動的に付加されます。
- .HEX 入力ファイルの項目で設定されたモードと同じ。
.BIN 入力ファイルの項目で設定されたモードに1足されたモードにな
ります。
- (option) オプションは省略することもできますが、
省略時はデフォルトが採用されます。
- B モード0を選択します。(デフォルト)
-D モード2を選択します。
-H モード3を選択します。
設定モードは、拡張子による自動判別が優先します。
入出力ファイルの拡張子が、'.BIN'、'.HEX' のいずれ かの場合、
機能設定オプションは指定しないでください。
- Oxxxx オフセットアドレスを指定します。省略した場合 0h となります。
アドレスは、HEX (0~F) 1~4 桁で記述してください。
入力ファイルが Binary, Intel HEX どちらとも有効です。
アドレス 0h から、オフセットアドレスまでは、フィルデータ・
オプションで指定したデータとなります。
- Fxx フィルデータを指定します。省略した場合 FFh となります。
データは、HEX (0~F) 1~2 桁で記述してください。
データが定義されていない部分を指定したデータで埋めます。

- Rxxxx 使用する ROM の容量を指定してください。
省略した場合、8000h になります。
ROM の容量は、HEX (0~F) 1~4 桁で記述してください。
指定できる容量は、0001h~FFFFh までです。
- Nxx ROM の番号を指定してください。省略した場合、1 になります。
ROM の番号は、HEX (0~F) 1~2 桁で記述してください。

【例】 以下のメモリーマップを持つシステムの場合

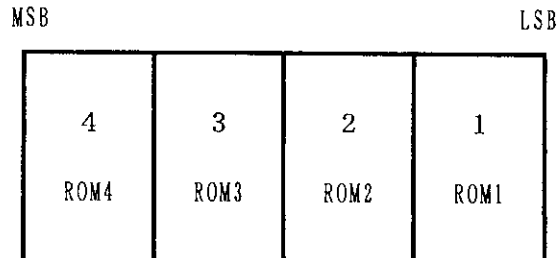


ROM1 のデータを作成する場合、-R4000 -N1 を指定します。
-R2000 を指定した場合以下のように分割される。
-N1 --- 0000h~3FFFh ◎ROM1 のエリアに該当する
-N2 --- 4000h~7FFFh

ROM2 のデータを作成する場合、-R2000 -N3 を指定します。
-R2000 を指定した場合以下のように分割される。
-N1 --- 0000h~1FFFh
-N2 --- 2000h~3FFFh
-N3 --- 4000h~5FFFh ◎ROM2 のエリアに該当する。

- Wn 1ワードのバイト数を設定します。1~9 を指定してください。
省略した場合、1 となります。
- Ln ワード内での、データの位置を指定します。
省略した場合、1 となります。
-Wn オプションが省略されている場合は無効です。
データの位置は 1 から -Wn オプションで指定した値 n までです。

【例】 下図の様な 32 ビットデータバスが必要な場合、



1 ワードが 4 バイトなので、-W4 を指定します。
ROM1 に書き込むデータを作成する場合 -L1 を指定します。
ROM2 に書き込むデータを作成する場合 -L2 を指定します。
ROM3 に書き込むデータを作成する場合 -L3 を指定します。
ROM4 に書き込むデータを作成する場合 -L4 を指定します。

・デフォルトの設定は以下のオプションを設定した場合と同等です。

```
HEXUTL file1.BIN file2.HEX -B -O0 -F0 -R8000 -N1 -W1 -L0
```

4、HEXUTL.COM を用いた応用例

【I】、基本形

```
HEXUTL TEST
```

[動作] 入力ファイル TEST.BIN を読み込み、
出力ファイル TEST.HEX を作成します。

【II】、基本の変形

```
HEXUTL MOVE.BAS -04000
```

[動作] 入力ファイル MOVE.BAS を読み込み、
出力ファイル MOVE.HEX を作成します。
出力ファイルには、4000H のオフセットアドレスが付加されます。
アドレス 0000h~3FFFh まで FFh で埋められます。

【111】、応用編

HEXUTL ROBOT.HEX ROBOT_O -W2 -L1 -O100 -FFF : 16ビットバス用、偶数バイト
HEXUTL ROBOT.HEX ROBOT_E -W2 -L2 -O100 -FFF : 16ビットバス用、奇数バイト

[解説] これは、1ワードが2バイトの16ビットバスコンピュータなどのROMを作成する場合に指定するオプションの例です。

1ワードが2バイトであることを-W2オプションで指定します。
偶数バイトは、-L1オプションとなります。

86系のCOMファイルをROMに書き込む場合にこのようになるでしょう。

- [動作1] 入力ファイル ROBOT.HEX を読み込み、
出力ファイル ROBOT_O.HEX を作成します。
出力ファイル ROBOT_O.HEX には、オフセットアドレス 100Hが足され、
入力ファイルの偶数アドレスのデータが抽出され出力されます。
アドレス 0000h~007Fh まで FFh で埋められます。
- [動作2] 入力ファイル ROBOT.HEX を読み込み、
出力ファイル ROBOT_E.HEX を作成します。
出力ファイル ROBOT_E.HEX には、オフセットアドレス 100Hが足され、
入力ファイルの奇数アドレスのデータが抽出され出力されます。
アドレス 0000h~007Fh まで FFh で埋められます。

5、プログラム終了時に出力されるエラーレベルについて

- 0: 正常に終了した。
- 1: HEX 文字列に無効な文字が含まれていた。
- 2: ファイル名が無効 (存在しない、アクセスできない等)。
- 3: ファイルが見つからない。
- 4: 入力ファイル (Binary) が、65536 BYTE を越えるため扱えない。
- 5: 最終アドレスが FFFFh を越えるため扱えない。
- 6: 読み込みファイルを開くことが出来なかった。
- 7: 書き込みファイルが開くことが出来なかった。
- 8: 入力書式に誤りがあるか、入力ファイル名がない。

が出力されますので、バッチファイルに組み込むことが出来ます。

6、終りに

本プログラムは、コードサイズ削減を優先した構成となっています。基本動作部分のバグは、取ったつもりでいますが、制作者の意図しない使い方をした場合、簡単に暴走することがあります。特に、エラー処理が甘かったりしますので、意味不明のエラーで止まってしまうことが考えられます。

最後に、HEXUTL は、これまでになく高機能なファイルコンバータですので、使いこなすには、試行錯誤が必要かと思いますが、きっとお役に立てると思います。目的にかなった動作モードが見つかるよう心からお祈り致します。

■ - 5 - Z84C015. ASM

これはおまけとして付けました。AKI80開発時に使用してください。

Z84C015の固有のIOの定義、初期化ルーチン、IO割込み(IM2)の割込みベクトルなどのサンプルが記述されたアセンブラファイルです。IO初期化データや、割込みフックなどは記述し直し利用してください。

簡単にリモートデバッグ対応にできるよう記述してあります。この場合4行目

```
DBUG EQU 0000H を
DBUG EQU 8000H に変更してください。
```

またリストの0107番地スタックポインタ設定で、

```
LD SP,0 としてありますが、SRAMが64Kの場合には、
LD SP,0A000H としてください。またリモートデバッグでその領域をシステ
```

ムスタックとして使用してしまいますので、F000程度に変更しておいたほうがよいかも知れません。詳細はデバッグのマニュアルをご覧ください。

```
LD SP,0F000H
```

リストの0116番地の JR \$ はここで無限ループに入りますのでカットしてください。(ターミネイトのつもりで入れてあります。)これ以降にプログラムを書いてください。

Z84C015. ASMのリストファイルを示します。

```

*****
:Z84C015専用セットアップ
:AKI80のスタートアップアセンブラです。
:アセンブラのヘッダファイルとして使用ください。
*****
0000 EQU 0000H ;REMORT DEBUG時には8000Hに変更
0010 EQU ;パラレルポートI/Oアドレスセット
0011 EQU ICH ;Aポートデータ
001E EQU IEH ;Bポートデータ
0018 EQU ;SIOポートI/Oアドレスセット
001A EQU IAH ;ch8送信/受信バッファ
0010 EQU ;カウンタタイマI/Oアドレスセット
0011 EQU I0H ;CTC0のアドレス
0012 EQU I1H ;CTC1のアドレス
0013 EQU I2H ;CTC2のアドレス
0014 EQU I3H ;CTC3のアドレス
00E0 EQU ;Z84C015専用I/Oアドレスセット
00F0 EQU ;WDIER,WDTPR,HALTMR
00F1 EQU ;WDC ;列フ(4EH) ;制御フ(B1H)
00F4 EQU ;DCC ;列フ(4EH) ;制御フ(B1H)
004E EQU ;WDCL ;列フ(4EH) ;制御フ(B1H)
*****
: I/Oセットデータ
: *****
: 代表的な値を記入済み。適宜変更のこと。
*****
0000 ORG DBUG
0001 JP START
0002 PLOACD: DB 00FH ;PI0Aモードワード **001111
0003 DB 00H ;PI0Aデータレクシオンワード
0004 DB 17H ;PI0Aインタラプトマスクワード ***0111
0005 DB 00H ;PI0Aインタラプトマスクワード
0006 DB 0E4H ;PI0Aインタラプトベクトル
0007 EQU $
0008 PAEND: EQU 0CFH ;PI0Bモードワード **001111
0009 PLOBCD: DB 00H ;PI0Bデータレクシオンワード
000A DB 17H ;PI0Bインタラプトマスクワード ***0111
000B DB 00H ;PI0Bインタラプトマスクワード
000C DB 0E6H ;PI0Bインタラプトベクトル
000D EQU $
000E SIOACD: DB 18H ;SIOA W0 チャネルリセット
000F DB 04H ;SIOA W0 ポインタ4
0010 DB 44H ;SIOA W4 ログ/リセット、ストップ、ビート、リセット、リセット
0011 DB 01H ;SIOA W0 チャネルリセット
0012 DB 03H ;SIOA W1 制込制御の1作/ルディ
0013 DB 0E1H ;SIOA W3 制御情報
0014 DB 05H ;SIOA W0 ポインタ5
0015 DB 68H ;SIOA W5 送信バッファ制御情報
0016 EQU $
0017 SAEND: EQU 18H ;SIOB W0 チャネルリセット
0018 SIOBCD: DB 04H ;SIOB W0 ポインタ4
0019 DB 44H ;SIOB W4 ログ/リセット、ストップ、ビート、リセット、リセット
001A DB 01H ;SIOB W0 チャネルリセット
001B DB 02H ;SIOB W1 制込制御の1作/ルディ
001C DB 0F0H ;SIOB W2 ポインタ2
*****

```

```

001D 03          DB      03H      :SIOB WR0 ポインタ3
001E E1          DB      0E1H     :SIOB WR3 受信バッファ制御情報
001F 05          DB      05H      :SIOB WR0 ポインタ5
0020 68          DB      68H      :SIOB WR5 送信バッファ制御情報
0021          SBEND: EQU      $
:
0021 05          CTC0CD: DB      05H      :CTC0 チャンネルコントロールワード *****1
0022 00          DB      00H      :CTC0 タイムコンスタントレジスタ
0023 E8          DB      0E8H     :CTC0 インタラプトベクトル*****000
0024          COEND: EQU      $
0024 05          CTC1CD: DB      05H      :CTC1 チャンネルコントロールワード *****1
0025 00          DB      00H      :CTC1 タイムコンスタントレジスタ
0026          C1END: EQU      $
0026 05          CTC2CD: DB      05H      :CTC2 チャンネルコントロールワード *****1
0027 00          DB      00H      :CTC2 タイムコンスタントレジスタ
0028          C2END: EQU      $
0028 05          CTC3CD: DB      05H      :CTC3 チャンネルコントロールワード *****1
0029 00          DB      00H      :CTC3 タイムコンスタントレジスタ
002A          C3END: EQU      $
:
002A FB          WDMCD: DB      0FBH     :ウォッチドック、リセット設定 *****011
002B 00          DGCCD: DB      00H      :デインジチェーン順位設定 00000***
:
:*****
:      I N T 処理 (IM1)
:*****
0038          ORG      DBUG + 0038H
0038 FB          EI
0039 ED 4D       RETI
:
:*****
:      N M I 処理
:*****
0066          ORG      DBUG + 0066H
0066 ED 45       RETN      : N M I 禁止
:
:*****
:      I/O セットアップ
:*****
0068 21 03 00   IOSET: LD      HL,PIOACD   :PIOAコマンドセットアップ
006B 06 05      LD      B,PAEND-PIOACD
006D 0E 1D      LD      C,PIOA+1
006F ED B3      OTIR
0071 21 08 00   LD      HL,PIOBCD   :PIOBコマンドセットアップ
0074 06 05      LD      B,PBEND-PIOBCD
0076 0E 1F      LD      C,PIOB+1
0078 ED B3      OTIR
007A 21 21 00   LD      HL,CTC0CD   :CTC0コマンドセットアップ
007D 06 03      LD      B,COEND-CTC0CD
007F 0E 10      LD      C,CTC0
0081 ED B3      OTIR
0083 21 24 00   LD      HL,CTC1CD   :CTC1コマンドセットアップ
0086 06 02      LD      B,C1END-CTC1CD
0088 0E 11      LD      C,CTC1
008A ED B3      OTIR
008C 21 26 00   LD      HL,CTC2CD   :CTC2コマンドセットアップ
008F 06 02      LD      B,C2END-CTC2CD
0091 0E 12      LD      C,CTC2
0093 ED B3      OTIR
0095 21 28 00   LD      HL,CTC3CD   :CTC3コマンドセットアップ
0098 06 02      LD      B,C3END-CTC3CD
009A 0E 13      LD      C,CTC3
009C ED B3      OTIR
009E 21 0D 00   LD      HL,SIOACD   :SIOAコマンドセットアップ
00A1 06 09      LD      B,SAEND-SIOACD
00A3 0E 19      LD      C,SIOA+1
00A5 ED B3      OTIR
00A7 21 16 00   LD      HL,SIOBCD   :SIOBコマンドセットアップ
00AA 06 0B      LD      B,SBEND-SIOBCD

```

```

00AC 0E 1B          LD      C,SIOB+1
00AE ED B3          OTIR
00B0 3A 2A 00       LD      A,(WDMCD)      :ウォッチドック、リセットモードセット
00B3 D3 F0          OUT     (WDM).A
00B5 3A 2B 00       LD      A,(DGCCD)      :割込優先順位セット
00B8 D3 F4          OUT     (DGC).A
00BA 3E 4E          LD      A,WDCCL        :ウォッチドッククリア
00BC D3 F1          OUT     (WDC).A
00BE C9             RET
:*****
:      割込アドレス、DWL設定。
:      アドレス値をそのまま記入のこと
:*****
00E4          ORG      DBUG + 00E4H
00E4 00 00         DW      0000H          :PIOA割込
00E6 00 00         DW      0000H          :PIOB割込
00E8 00 00         DW      0000H          :CTC0割込
00EA 00 00         DW      0000H          :CTC1割込
00EC 00 00         DW      0000H          :CTC2割込
00EE 00 00         DW      0000H          :CTC3割込
00F0 00 00         DW      0000H          :SIOBトランスミッタハフェンフティ
00F2 00 00         DW      0000H          :SIOB外部ステータス割込
00F4 00 00         DW      0000H          :SIOBレシーバキャラクタアハイラフル
00F6 00 00         DW      0000H          :SIOB特殊受信状態
00F8 00 00         DW      0000H          :SIOAトランスミッタハフェンフティ
00FA 00 00         DW      0000H          :SIOA外部ステータス割込
00FC 00 00         DW      0000H          :SIOAレシーバキャラクタアハイラフル
00FE 00 00         DW      0000H          :SIOA特殊受信状態
:*****
:      スタート
:*****
0100          ORG      DBUG + 0100H
0100 F3          DI
0101 ED 5E          IM      2          :セットアップ中、割込不可
0103 3E 00         LD      A,DBUG / 100H   :割込モード2
0105 ED 47         LD      I,A            :割込上位ベクトルロード
0107 31 00 00     LD      SP,0          :スタックポインタ FFFFHから
:
010A 21 00 80     LD      HL,8000H      :外部I/O使用時のRESET待ち
010D 2B          WEIT: DEC     HL
010E 7C          LD      A,H
010F B5          OR      L
0110 20 FB       JR      NZ,WEIT
:
0112 CD 68 00     CALL   IOSET
0115 FB          EI
:*****
:      ここから好きにする。
:*****
0116 18 FE       JR      $
0118          END

```


■ 6 - 特別付録

ROMライターには特別付録としてターゲットデバッガのモニタが内蔵されています。これによりROMライターでありながら、ROMライターのRAM上でデバッグが可能です。

ターゲットデバッグとは開発すべきCPUボード（ターゲットと呼ぶ）上にプログラムをロードし、ホストコンピュータ（この場合は98）と通信しながら、実際にプログラムを走らせデバッグ（虫取り）することです。ホスト側はターゲット側の状態を監視し、実機によるテストを行います。

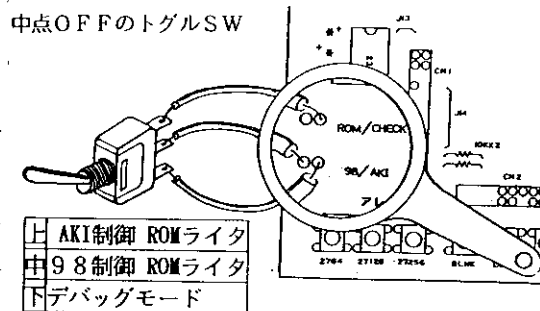
AKI80をより効率よく開発していただくため、システムロード社のご協力により、Z VISION REMORTのターゲットプログラムをROMライターのROM上に実装しています。これによりROMライターのモードを切り替えれば、即デバッガとして動作可能です。

Z VISION REMORTのホストプログラムは別途必要ですが、ミニ版を実費程度で御用意していますので、御利用いただければ幸いです。御案内は最終頁にあります。

ROMライターセットアップ

ROMライターのモード切り替えで、
- 1 - 組み立て の章には記載がありませんが、隠しスイッチとしてもう一つのモードが用意されています。

右記のようにSWを配線して、デバッグモードを選択後、電源をいれ直すか、リセットを押し、立ち上げ直してください。



Z VISION REMORTの使い方については、付属のマニュアルに詳しく記載されていますのでそちらを参照してください。ハードウェアの説明でRS232Cケーブル、SIOのポートなどはROMライターのそれと一部違いがありますが、一切のハードウェア変更は必要ありません。

注 意

ROMライターではRAMにスタックを取りませんが、Z VISION REMORTではRAMにシステムスタックを取りますので、27256の書き込みにはデータの再ロードをしてください。またこの関係で、ROMライターモード、デバッグモードの移行にはAKI80のリセットが必要です。お忘れなく。

Z VISION REMORTはホスト側、ターゲット側共に著作権はシステムロード社が保有します。

■ 6-1 BASICインタプリタ内蔵

「Z VISION REMOTE」同様システムロード社のご好意によりROMライター中にBASICインタプリタが内蔵されています。

AKI80をより簡単に開発していただくため、BASICでプログラムが組めます。「Z80 Basic インタプリタ」のホストプログラムは別途必要ですが、これも実費程度でご用意していますので、ご利用いただければ幸いです。ご案内の項をご覧ください。

専用ターミナルソフトBTERMよりROMライター上でBasic言語によるプログラム開発が行なえます。また完成したBasicプログラムはターミナルソフトBTERM上よりROM化でき、AKI80単体上でも自走可能です。

概要については下記の通りです。詳細は「Z80 Basic インタプリタ」付属のマニュアルをご覧ください。

ROMライターセットアップはデバッグモードにします。(Z VISION REMOTEと同じポジションです。REMOTEとBasicは自動認識になっています。)
BASICの起動はBTERMを起動したのち、ROMライターをリセット。逆にZ VISION REMOTEではリセットしたのちZVRを起動します。

【 REMOTE BASIC命令一覧表 】

LET	変数へ代入します。
PRINT	コンソール表示します。
INPUT	コンソールから変数へ値を入力します。
GOTO	指定した行番号へジャンプします。
GOSUB	サブルーチン呼び出しします。
RETURN	サブルーチンを終了し、元のプログラムへ実行を移します。
IF	論理式の条件判断を行います。
FOR	FORからNEXTまでの間の文を繰り返し実行します。
NEXT	FORからNEXTまでの間の文を繰り返し実行します。
STOP	プログラムを中断します。
REM	注釈文を宣言します。
SYSTEM	REMOTE BASICのリセットを行います。
CALL	機械語サブルーチン呼び出しします。
POKE	メモリにデータを書き込みます。
OUT	IOポートにデータを書き込みます。

【 REMOTE BASIC関数一覧表 】

RND	乱数を返します。
ABS	絶対値を返します。
SIZE	使用可能メモリのバイト数を返します。
PEEK	メモリアドレスのデータを返します。
INP	IOポートアドレスのデータを返します。

【 REMOTE BASICコマンド一覧表 】

RUN	プログラムを実行します。
CONT	途中で中断したプログラムの実行を再開します。
LIST	プログラムの内容を表示します。
NEW	メモリ内のプログラムを消去します。
WRITE	メモリ内のプログラムをROMに書き込みます。
WRITE A	メモリ内のプログラムをROMに書き込みます。ただし、書き込まれたROMはCPUボード単体で起動し、自動的にプログラムを実行します。
BLANK	ROMのブランクチェックを行います。

御 紹 介

AKIROM. EXEおよびHEXUTL. COMはコピーフリーといたします。
フリーソフトウェアです。

上記のソースファイル及び実行形式ファイル、ライター側のプログラムソースを実費にて配布いたします。

ROMライターのAKI80に10MHzバージョンが使われるようになりましたので10MHz用にパッチをあてたシステムROMのHEXファイルも同梱いたします。

どうしても98ソフトが転送できない、AKIROM. EXEを改造してみたいという方、10MHzのAKI80なのになぜ9.8304MHzで動かないんだという方は、ご利用ください。日本語バージョンのAKIROMJ. EXE、矢印バージョンのAKIROMM. EXEなども含まれます。

AKIROM、HEXUTLソースファイル ¥1500-
5 inch 2HD 又は 3.5 inch 2DD を指定してください。

Z80クロスアセンブラ XA80
[システムロード製]

Z80、64180用のアブソリュート（絶対番地）クロスアセンブラです。全てのDOSマシンで動作。

XA80 5 inch 2HDのみ ¥2500-

Z VIZION REMOTE ミニ版
[システムロード製]

前項で紹介した、Z80フルスクリーンターゲットデバッガです。
ターゲットデバッグとは、ターゲットとなるボードの上に実際にプログラムロードしてホストコンピュータ(9801)と通信し、実行させながらデバッグします。これにより視覚的に、効率的にプログラムの開発が行なえます。
ミニ版は本製品版に比べ、多少機能の限定(DBUG領域8000から2Kbyte)がありますが、本製品版の評価、アマチュアの使用には十二分に耐えます。詳細はインターフェース誌やトラ技誌のシステムロード社の広告でぜひ一度ご覧ください。

Z VIZION REMOTE ミニ版 ¥2500- (5 inch 2HDのみ)
Z VIZION REMOTE 本製品版 ¥49000- (両ディスク付属)

Z80 BASIC インタプリタ
[システムロード製]

従来よりある「TINY BASIC」をAKI80その他のZ80CPUボードとパソコン(9801)をRS232Cで接続し、簡単な制御、ハードのテストなど手軽にBASICプログラムを走らせることが出来ます。専用ターミナルソフト、インタプリタ部、サンプルプログラムなどが付属。

Z80 BASIC インタプリタ (5 inch 2HDのみ) ¥2500-

AKI80開発関連紹介

MPDB MPデバッガ

〔無線パーツ製〕

AKI80専用のリモートデバッガキット。

- ◆8MHz, 10MHz自動対応版
- ◆基本的なダンプやムーブなどのほかにトレースやゴーアンドブ레이크ができる本格デバッガソフト。
- ◆RS232Cを通してパソコンから行うリモートでバッグ型式。XMODEMプロトコルを使用してオブジェクトコードを送受信可能。
- ◆パソコン側ターミナルソフト添付(3.5インチ, 5インチ2HDディスク)。
- ◆インターフェイス基板はIC1ヶのシンプル回路で、簡単に拡張可能。
- ◆必要なコネクタ・ケーブル類を含む部品キット。
- ◆PC9801用

MPDB

¥13950-

Z80用マクロアセンブラ MPZ80

〔無線パーツ製〕

究極のZ80用のクロスアセンブラです。

- ◆独自のハッシュ検索法による高速なラベル検索
 - ◆逆ポーランド手法による強力なパラメータ演算機能
 - ◆強力なマクロ機能によるプログラムの簡易化
 - ◆条件つきアセンブル機能
 - ◆16ビットレジスタ同志のロード命令他の拡張命令
 - ◆リストファイル作成・制御機能
- など強力な機能が満載です。マクロ(置き換え)によるプログラム簡易化は一度使ってしまうと手放せません。すべてのDOSマシンで動作。MPDBとの組合せ最適

MPZ80

¥10,000.

送料は数量に関係無く一律¥600-です。当社通販部までお申し込みください。当社では消費税はいただいております。

質問は封書か往復葉書にてお願いいたします。

158 東京都世田谷区瀬田5-35-6 秋月電子通商

ROM WRITER KIT ver1.2 (C) AKIZUKI DENSI by GO! 92-6-6

Special Thanks for KOHKI & SYSTEM LOAD, Ltd

当機運用に当たりホスト側のハード及びソフトに、いかなる障害が発生しても、当社では一切その責を追いかねます。悪しからず御了承ください。

AKIROMライター バージョンアップのお知らせ Ver1.2

大変ご好評いただいております、ROMライターキットのプログラムのマイナーバージョンアップを行ないました。

■主なバージョンアップ内容

§ ROMライター機能

- ★2864 (EEPROM) の書込サポート
- ★超高速書込アルゴリズムROM及び一部未対応だったROMの書込サポート

§ プログラム開発機能

- ★Basicのインタプリタ内蔵
(別売のZ80 Basic インタプリタ [SYSTEM LOAD] を使用すれば、専用ターミナルソフト BTERMよりROMライター上でBasic言語によるプログラム開発が行なえます。また完成したBasicプログラムはターミナルソフトBTERM上よりROM化でき、AKI80上で自走可能です。)

■マニュアル変更内容

★AKI80のX'TALには、付属の14,7456MHzをご使用ください。ポーレートの関係で、これ以外のX'TALでは動作しません。

★98側ソフトは本文同様、ROMライターから転送を行ないますが、転送されるAK.COMのファイルサイズは本文とは異なります。

AK.COM	25078	(25079の場合あり)
--------	-------	--------------

★ファイル解凍内容

AK.COMを実行すると以下の2つのファイルが解凍されます。

AKIROM.EXE	24942
HEXUTL.COM	10820

Basic インタプリタを内蔵したため、ROMの空きスペースがなくなり、Z84C015.ASMはAK.COMには入っていません。本文にZ84C015.ASMの内容はすべて掲載していますのでそちらを参照してください。

★AKIROM.EXEについて

2864サポートのため、[ROM TYPE] ウィンドウでは2864が選択できるようになりました。またコマンドラインオプションにおいても、-2864が有効となります。それ以外はまったく同様です。(本体のみ、AKIコントロールでは2864の書込は行なえません。)

★HEXUTL.COMについて

Ver1.1からVer2に変更されています。使用法は大幅に異なりますので、HEXUTL.DOCを参照してください。

★BASIC インタプリタについて

本文30ページ「特別付録」同様、システムロードのご協力により、Basic インタプリタ及び、Z VISION REMOTEのターゲットプログラムが内蔵されています。

AK180をより簡単に開発していただくため、BASICでプログラムが組めます。「Z80 Basic インタプリタ」のホストプログラムは別途必要ですが、これも実費程度でご用意していますので、ご利用いただければ幸いです。(¥2,500-)

ROMライターセットアップはデバッグモードにします。(Z VISION REMOTEと同じポジションです。REMOTEとBasicは自動認識になっています。) BTERMを起動したのち、デバッグモード選択し電源を入れなおすか、リセットスイッチを押してください。

概要については以下のとおりです。詳細は「Z80 Basic インタプリタ」付属のマニュアルをご覧ください。

【ディスク内容】

- ・BASIC専用TERMINALソフト
- ・REMOTE BASIC本体(HEXファイル)
- ・BASICサンプルプログラム

【 REMOTE BASIC命令一覧表 】

LET	変数へ代入します。
PRINT	コンソール表示します。
INPUT	コンソールから変数へ値を入力します。
GOTO	指定した行番号へジャンプします。
GOSUB	サブルーチン呼び出します。
RETURN	サブルーチンを終了し、元のプログラムへ実行を移します。
IF	論理式の条件判断を行います。
FOR	FORからNEXTまでの間の文を繰り返し実行します。
NEXT	FORからNEXTまでの間の文を繰り返し実行します。
STOP	プログラムを中断します。
REM	注釈文を宣言します。
SYSTEM	REMOTE BASICのリセットを行います。
CALL	機械語サブルーチン呼び出します。
POKE	メモリにデータを書き込みます。
OUT	IOポートにデータを書き込みます。

【 REMOTE BASIC関数一覧表 】

RND	乱数を返します。
ABS	絶対値を返します。
SIZE	使用可能メモリのバイト数を返します。
PEEK	メモリアドレスのデータを返します。
INP	IOポートアドレスのデータを返します。

【 REMOTE BASICコマンド一覧表 】

RUN	プログラムを実行します。
CONT	途中で中断したプログラムの実行を再開します。
LIST	プログラムの内容を表示します。
NEW	メモリ内のプログラムを消去します。
WRITE	メモリ内のプログラムをROMに書き込みます。
WRITE A	メモリ内のプログラムをROMに書き込みます。ただし、書き込まれたROMはCPUボード単体で起動し、自動的にプログラムを実行します。
BLANK	ROMのブランクチェックを行います。

★ゼロプレッシャーソケットについて

3M製TEXT TOOLからARIES (I/I)製パーインユニバーサルテストソケットに変更いたしました。これは耐久性及び使い易さからの選択です。3M製より丈夫で、何よりゼロプレッシャーソケットが、丸ピンソケットに入るのが魅力です。高価な専用下駄あげソケットを使用しなくて済み、交換対策や下駄あげも安価な丸ピンソケットが使用できます。

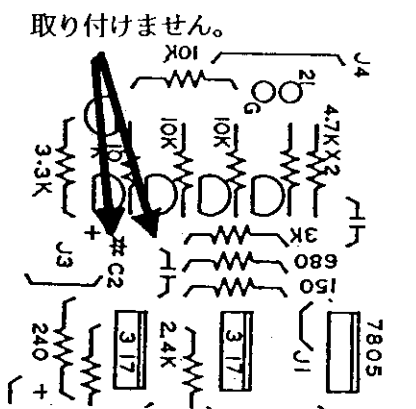
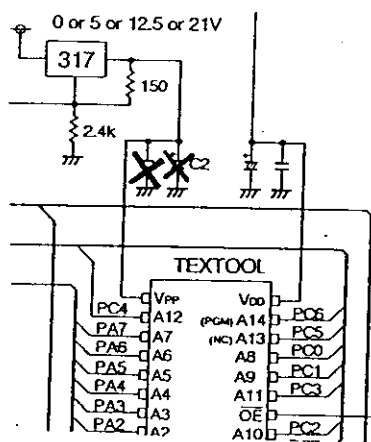
★回路変更について

回路を下図のように変更いたします。

具体的にはVpp用の317のパスコンである0.1μFと10μF (C2)は取り付けません。これによりインテル方式高速書込未対応のROMも書込可能となります。

★回路変更にもない、パーツリストから以下の2点の部品が削除されます。

- 10μF 25V以上の電解コンデンサ 1個
- 0.1μF 積層セラミックコンデンサ 1個



★ご注意

AK180をROMライターキットにご使用の場合、クリスタル発振器は14.7456MHzをお使いください。

HEXUTL (Ver 2.00) の使用方法
----- Apl. 11, 1992 By (有) 秋月電子通商 -----

1、HELUTL.COMとは、

HEXUTL は、アセンブラ、コンパイラなどの Binary または、Intel HEX ファイルを ROM Writer などを使用する Intel HEX ファイルに変換するファイルコンバータです。但し、Ver 2.00 で扱えるデータサイズは、65536 バイトまでです。

ファイルコンバータ機能には、

```
mode 0 : Binary file --> HEX file  
mode 1 : Binary file --> Binary file  
mode 2 : HEX file    --> HEX file  
mode 3 : HEX file    --> Binary file
```

以上4つの動作モードがあります。これらの動作モードは、入出力ファイルの拡張子が '.HEX' か、'.BIN' かを見て自動判別していますが、それ以外の拡張子の場合、オプションパラメータを用いて機能を指定することができます。

この他、8 BIT データバスの ROM を使用して、16, 24, 32, バス対応のデータを作成する機能や、ROM 容量よりも大きなデータファイルを分割して、それぞれの ROM に対応したデータファイルを作成する機能があります。

2、基本機能を利用する場合、

【1】 Binary ファイルを Intel HEX ファイルに変換する場合

Binary ファイルが、'A:¥PROG¥LEDON.BIN' の場合、

```
A:¥) HEXUTL A:¥PROG¥LEDON.BIN
```

と入力することにより、Intel HEX ファイル 'A:¥PROG¥LEDON.HEX' が作成されます。

【2】 Intel HEX ファイルを Binary ファイルに変換する場合

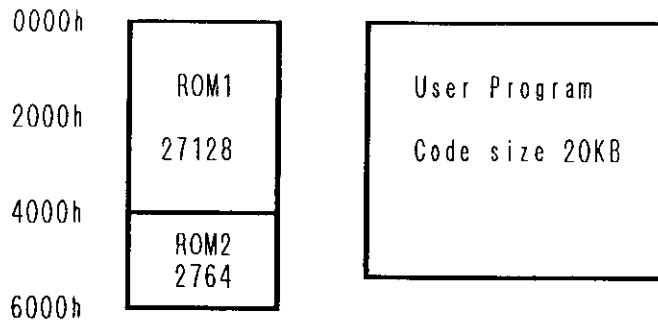
Intel HEX ファイルが、'A:¥PROG¥LEDOFF.HEX' の場合、

```
A:¥) HEXUTL A:¥PROG¥LEDOFF.HEX A:¥PROG¥LEDOFF.BIN
```

と入力することにより、'A:¥PROG¥LEDOFF.BIN' が作成されます。

ROM の容量は、HEX (0~F) 1~4 桁で記述してください。
 指定できる容量は、0001h~FFFFhまでです。
 ROM の番号を指定してください。省略した場合、1 になります。
 ROM の番号は、HEX (0~F) 1~2 桁で記述してください。

【例】 以下のメモリーマップを持つシステムの場合

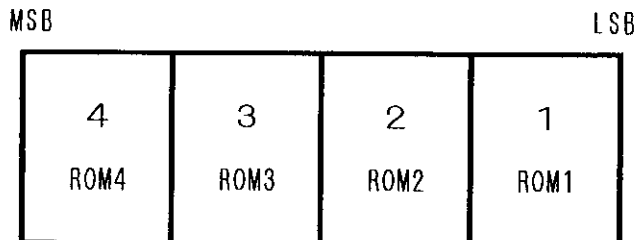


ROM1 のデータを作成する場合、-R4000 -N1 を指定します。
 -R2000 を指定した場合以下のように分割される。
 -N1 --- 0000h~3FFFh ◎ROM1 のエリアに該当する
 -N2 --- 4000h~7FFFh

ROM2 のデータを作成する場合、-R2000 -N3 を指定します。
 -R2000 を指定した場合以下のように分割される。
 -N1 --- 0000h~1FFFh
 -N2 --- 2000h~3FFFh
 -N3 --- 4000h~5FFFh ◎ROM2 のエリアに該当する。

-Wn 1ワードのバイト数を設定します。1~9 を指定してください。
 省略した場合、1 となります。
 -Ln ワード内での、データの位置を指定します。
 省略した場合、1 となります。
 -Wn オプションが省略されている場合は無効です。
 データの位置は 1 から -Wn オプションで指定した値 n までです。

【例】 下図の様な 32 ビットデータバスが必要な場合、



1 ワードが 4 バイトなので、-W4 を指定します。
 ROM1 に書き込むデータを作成する場合 -L1 を指定します。
 ROM2 に書き込むデータを作成する場合 -L2 を指定します。

3、入力書式

【書式】 HEXUTL [option] file_name_1 [option] [file_name_2] [option]

【パラメータの意味】

- file_name_1 入力ファイル名、省略することはできません。
入力ファイルの拡張子により以下のモードが自動選択されます。
- 拡張子省略時 mode 0 が選択されたものと解釈します。
拡張子 '.BIN' が自動的に付加されます。
- .BIN mode 0 が選択されたものと解釈します。
.HEX mode 2 が選択されたものと解釈します。
- [file_name_2] 出力ファイル名、省略することができます。
省略された場合、入力ファイル名と同じになります。
出力ファイルの拡張子により、入力ファイルの拡張子と組み合わせ
以下のモードが自動的に設定されます。
- 拡張子省略時 入力ファイルの項目で設定されたモードと同じ。
拡張子 '.HEX' が自動的に付加されます。
- .HEX 入力ファイルの項目で設定されたモードと同じ。
.BIN 入力ファイルの項目で設定されたモードに1足されたモードにな
ります。
- (option) オプションは省略することもできますが、
省略時はデフォルトが採用されます。
- B モード0を選択します。(デフォルト)
-D モード2を選択します。
-H モード3を選択します。
設定モードは、拡張子による自動判別が優先します。
入出力ファイルの拡張子が、'.BIN', '.HEX' のい
ずれかの場合、
機能設定オプションは指定しないでください。
- Oxxxx オフセットアドレスを指定します。省略した場合 0h となります。
アドレスは、HEX (0~F) 1~4 桁で記述してください。
入力ファイルが Binary, Intel HEX どちらとも有効です。
アドレス 0h から、オフセットアドレスまでは、フィルデータ・
オプションで指定したデータとなります。
- Fxx フィルデータを指定します。省略した場合 FFh となります。
データは、HEX (0~F) 1~2 桁で記述してください。
データが定義されていない部分を指定したデータで埋めます。
- Rxxxx 使用する ROM の容量を指定してください。
省略した場合、8000h となります。

ROM3 に書き込むデータを作成する場合 -L3 を指定します。
ROM4 に書き込むデータを作成する場合 -L4 を指定します。

・デフォルトの設定は以下のオプションを設定した場合と同等です。

```
HEXUTL file1, BIN file2, HEX -B -00 -F0 -R8000 -N1 -W1 -L0
```

4、HEXUTL.COM を用いた応用例

【I】、基本形

```
HEXUTL TEST
```

[動作] 入力ファイル TEST. BIN を読み込み、
出力ファイル TEST. HEX を作成します。

【II】、基本の変形

```
HEXUTL MOVE, BAS -04000
```

[動作] 入力ファイル MOVE. BAS を読み込み、
出力ファイル MOVE. HEX を作成します。
出力ファイルには、4000H のオフセットアドレスが付加されます。
アドレス 0000h~3FFFh まで FFh で埋められます。

【III】、応用編

```
HEXUTL ROBOT, HEX ROBOT_0 -W2 -L1 -0100 -FFF : 16ビットバス用、偶数バイト  
HEXUTL ROBOT, HEX ROBOT_E -W2 -L2 -0100 -FFF : 16ビットバス用、奇数バイト
```

[解説] これは、1 ワードが 2 バイトの 16 ビットバスコンピュータなどの ROM を作成する場合に指定するオプションの例です。
1 ワードが 2 バイトであることを -W2 オプションで指定します。
偶数バイトは、-L1 オプションとなります。

86系の COM ファイルをROMに書き込む場合にこのようになるでしょう。

[動作1] 入力ファイル ROBOT. HEX を読み込み、
出力ファイル ROBOT_0. HEX を作成します。
出力ファイル ROBOT_0. HEX には、オフセットアドレス 100Hが足され、
入力ファイルの偶数アドレスのデータが抽出され出力されます。
アドレス 0000h~007Fh まで FFh で埋められます。

[動作2] 入力ファイル ROBOT. HEX を読み込み、
出力ファイル ROBOT_E. HEX を作成します。
出力ファイル ROBOT_E. HEX には、オフセットアドレス 100Hが足され、
入力ファイルの奇数アドレスのデータが抽出され出力されます。
アドレス 0000h~007Fh まで FFh で埋められます。

5、プログラム終了時に出力されるエラーレベルについて

- 0: 正常に終了した。
- 1: HEX 文字列に無効な文字が含まれていた。
- 2: ファイル名が無効(存在しない、アクセスできない等)。
- 3: ファイルが見つからない。
- 4: 入力ファイル(Binary)が、65536 BYTE を越えるため扱えない。
- 5: 最終アドレスが FFFFh を越えるため扱えない。
- 6: 読み込みファイルを開くことが出来なかった。
- 7: 書き込みファイルが開くことが出来なかった。
- 8: 入力書式に誤りがあるか、入力ファイル名がない。

が出力されますので、バッチファイルに組み込むことが出来ます。

6、終りに

本プログラムは、コードサイズ削減を優先した構成となっています。基本動作部分のバグは、取ったつもりでいますが、制作者の意図しない使い方をした場合、簡単に暴走することがあります。特に、エラー処理が甘かったりしますので、意味不明のエラーで止まってしまうことが考えられます。

最後に、HEXUTL は、これまでになく高機能なファイルコンバータですので、使いこなすには、試行錯誤が必要かと思いますが、ぎっとお約に立てると思います。目的にかなった動作モードが見つかるよう心からお祈り致します。

1992年4月 BY T.K

AKI ROM WRITER KIT Ver1.2 追記 秋月電子通商 1992-5-27 by GO!
