8ビット・マイクロプロセッサ：Z80の プログラム開発（I）

福井　稔・岡田俊治・及川浩和

1．は　じ　め　に

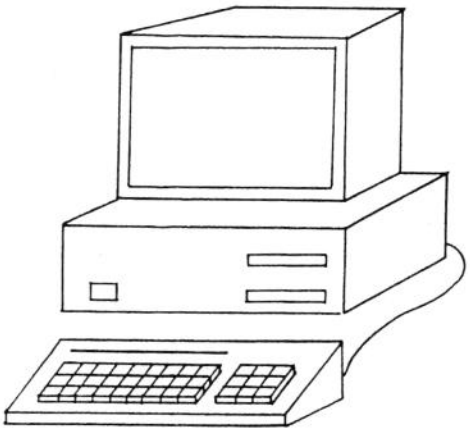
我国では1976年にTK—80というマイコン・キットがNECから発売になり，爆発的な売れ行きをみて，コンピュータは大型計算機の世界からアマチュアでもいじれる電子部品となってきた。その後半導体の製造技術の発展に支えられパーソナル・コンピュータとなって我々の周りに，日常的に存在するようになった。もう一方では，電子制御部品となって，家庭電化製品はもとより，産業界の多種多様な機器の頭脳となって，これを動かしている。この間コンピュータのCPU(マイクロプロセッサ）は4ビットから8ビット，16ビット，32ビットと大きくなっている。その中にあって，現在でも制御用マイクロプロセッサは，4ビットCUPが，家電製品や産業用機器の制御に最も数多く使われている。また8ビットCPUであるZ80には，現在までにCP/M上で動作する膨大な量のソフトウェアを資産として持ち，依然として大きなシェアを保っている。また16ビット（8086，80286），32ビット（80386）CPUなどは，8ビットCPUを基にして開発されたといえる。そのためマイクロプロセッサを理解するとき，8ビットCPU特にZ80を理解することが肝要であり，効率的でもある。ここでは，Z80マイコンの制御プログラムの開発の実際を，いくつかの実例を示しながら説明する。ただし，Z80CPUのプログラム言語であるニモニック・コードそれ自身は，多くの教科書，参考書で説明があるので詳しく書くことはせず，プログラム開発の流れを示すことにする。

2．プログラム開発の手順

プログラム開発では，一般的にパソコンでプログラムを書き，ROMライターでROMにプログラムを書き込み，そのROMをマイコンに実装し，マイコンを動かし，その動作をチェックする。必要ならプログラムを修正し，完成させる手順をとる。詳しくは，図2－1に示すような経過をとる。

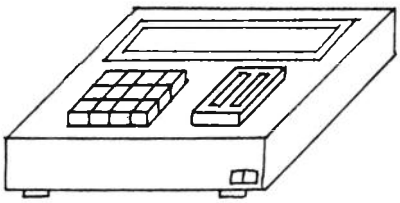
① パソコンで,エディタを使用して,アセンブリ言語によるソースプログラムを作成する。

② パソコンで，クロスアセンブラを使用して，ソースプログラムを機械語プログラムとインテル HEX ファイルに翻訳する。



パソコン

③ HEX ファイルを ROM ライターへ転送する。



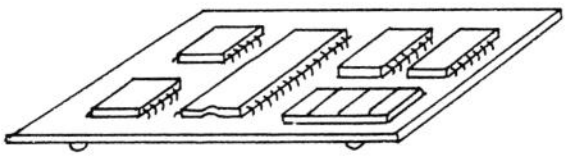
ROMライタ

④ ROM ライターを使用して ROM に書込む。

ROM

⑤ ROM をターゲットのマイコンに実装して動作させる。

⑥ 動作をチェックし，①へ戻りプログラムを修正し，⑥までの行程を繰り返し，プログラムを完成させる。



マイコンボード

図２－１　プログラム開発手順

マイコンのプログラム開発では，まず①パソコンにエディタとよぶソフトウェアを使って，アセンブリ言語で，ソースプログラムを書く。エディタは，我々はMIFES（メガソフト社）を使ったが，他にVZとか，手持ちのワープロソフトで書いてもよい。ただし書き上げた文書ファイルは，テキストファイルの形式（半角英数字）で保存することが必要である。使用する言語は，16ビット，32ビットのCPUでは，FORTRAN，Cなどの高級言語使用することも多いが，８ビットのCPUでは，たいてい機械語と直接対応したニモニックというアセンブリ言語を用いる。つぎに②そのソースプログラムをアセンブラというソフトウェアによって，機械語（バイナリ形式）のファイルに変換する。さらにそのファイルをROMライタに転送するために，インテル社形式のHEXファイルへ変換する。③このHEXファイルは，普通パソコンに装備されているRS232Cインターフェイスを通してROMライタへ転送される。転送のためのソフトウェアは，ROMライタに添付されている場合（ベッカー11）とアセンブラに付属している場合（PZ—80）がある。またアセンブルとは，狭い意味では②の変換作業をさすが，多くのアセンブラでは②③の操作を含んでいる。その後⑤ROMをターゲットのマイコンに実装して実行させる。⑥動作をチェックし，誤りがあるときは①へ戻りプログラムを修正し完成させる。この手順が，一般的なプログラム開発の手順であるが，ときには内容のわからない未知のROMから，ROMライターを使ってプログラムを読み出し，そのHEXファイルをソース・プログラムに変換し，解読する作業もある。このように逆に変換するソフトウェアをディスアセンブラ（逆アセンブラ）という。

表１に我々が使用したプログラム開発のためのハード・ウェアのシステム構成とソフト・ウェアを示した。

表１　プログラム開発環境

ハード・ウェア	パソコンシステム	ＰＣ９８０１	（ＮＥＣ）
	ＲＯＭライター	ＰＺ—Ｗ２	（太平洋工業）
		ベッカー１１	（アバール）
		ＡＫＩ—８０	（秋月電商）
	マイコン	ＰＺ—ＫＩ	（太平洋工業）
ソフト・ウェア	エディタ	MIFES	（メガソフト）
	アセンブラ	ＰＺ—８０	（太平洋工業）
		ＸＡ８０	（システム・ロード）
		ＺＡＳＭ	（マイクロ・オーグ）
	逆アセンブラ	ＺＺ	（マイクロ・オーグ）

3．アセンブル

前述のようにアセンブルとは，ニモニックで書かれたソースプログラムを，機械語プログラムに翻訳することであるが，その方法には，図２－２，図２－３に示すように，アブソリュート(絶対)・アセンブルとリロケータブル（再配置可能）・アセンブルと呼ばれる二つの方法がある。

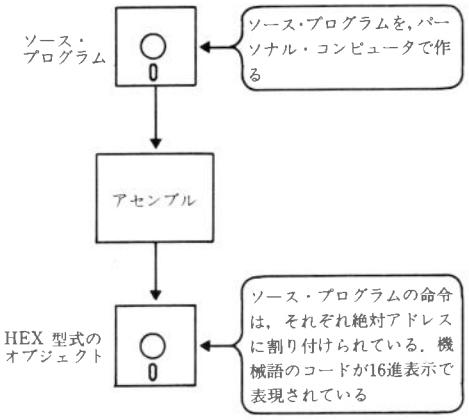


図２－２　アブソリュート・アセンブラによるプログラムの作成

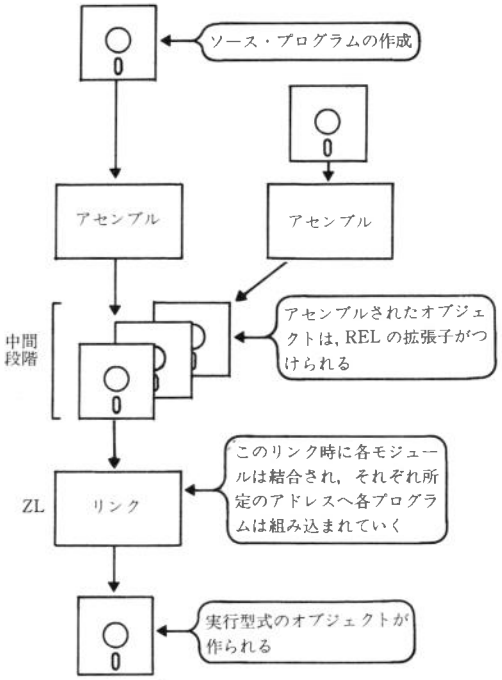


図２－３　リロケータブル・アセンブラによるプログラムの作成

アブソリュート・アセンブルという方法は，比較的短いプログラムで，それだけで一つの独立したプログラムに対して行なわれる方法で，アセンブラがソース･プログラムの情報に基づいて，機械語プログラムが実行されるアドレス（絶対番地）を直接割り付ける方法である。そのプログラムがどのメモリエリアで，実行されるか，直接ソース･プログラムで指定できる。したがって，ソース・プログラムの記述の段階で，どのように実行され，どのような結果になっているかということが，見通せる。初心者でも，Z80CPU のニモニック・コードがわかれば，ソース・プログラムから実行の様子が，容易に理解できる。

リロケータブル・アセンブルでは，プログラムを分割して，あるまとまった機能ごとにプログラムを記述し，あらかじめ機械語プログラムのモジュールにアセンブルしておく。そして全体のプログラムは，それらのモジュールを組合せて構成し，大きなプログラムを完成する手順を経る。それぞれの機能が独立して完成されていると，そのモジュール化されたプログラムは，他のプログラムから共通の処理ルーチンとして利用することができる。これによって，過去に作った処

理プログラムを毎回コーディングすることなく，ただ単に，それらのモジュールを組み合せて，プログラムを作ることが出来る。このとき，それぞれのモジュールは，最終のプログラムのなかで，どのメモリ・アドレスに割り付けられるかは，あらかじめ決めることはできない。そのため，各々のモジュールでは，機械語プログラムは，プログラムの先頭番地を基準に相対的にアドレス（相対番地）が割り付けられている。これら個々のモジュールで使われている変数やラベルの受け渡しの調整と，全体のプログラムのアドレスを決めることなどを行ない，多数のモジュールを連結し，ひとつの機械語プログラムにする。この役割を担うソフトウェアをリンカという。リロケータブル（再配置可能）アセンブルでは，アセンブラ，リンカとライブラリアン（ライブラリ作成ソフトウェア，ライブラリ・マネジャともいう。）を使用する。

現在，入手できる市販の Z80 用のアセンブラを，表２に示す。

表２　市販されている Z80 アセンブラの例

Ｍ８０（マイクロソフト）	（ＣＰ／Ｍ）	＊ＲＡ
ＰＺ－８０（太平洋工業）		＊ＡＡ
ＸＡ８０（システム・ロード）		＊ＡＡ
ＺＡＳＭ（マイクロ・オーグ）		＊ＲＡ
ＰＲＯＡＳＭII（ＩＡ８０，ＩＲ８０／岩崎技研，京都マイクロコンピュータ）		＊ＡＡ，＊ＲＡ

＊ＡＡ：アブソリュート・アセンブラ
＊ＲＡ：リロケータブル・アセンブラ

これらは，M80 を除いて他は，16ビット，32ビットパーソナルコンピュータ（MS—DOS）で動く，Z80 用クロスアセンブラである。

つぎに，アセンブラの実行例を示す。

３．１　アブソリュート・アセンブラ

３．１．１　PZ—80 アセンブラ

PZ—80 アセンブラは，太平洋工業で開発されたアセンブラソフトで，同社の Z80 のマイコン教育モジュール PZ—80K1 または PZ—80H1 とP—ROM ライター（PZ—W2）を利用したマイコン教育に使うことを前提にしている。そのため，狭い意味でのアセンブラ・ソフトウェアではなく，フログラムの開発に必要なさまざまな操作を包括した総合的なソフトウェアである。すなわち，PZ—80アセンブラは，スクリーン・エディタータイプのアセンブラソフトで，入力したニモニック・ソースプログラムを保存する，機械語に変換する，ROM に書き込む，などの一連の作業を，ファンクション・キーの操作により行うことができる。機械語に変換したプログラムを

ROMに書き込むには，別売りのP—ROMライタ（PZ—W2）をRS—232Cケーブルで接続する。書き込み対象ROMには，EP—ROM2716，27128，27128A，27128F，27128N，27256，27256F，2732，2764，2764F，2764N，EEP—ROM2816A，2864Aがある。PZ—80アセンブラのソフトウェア構成と必要なシステムファイルを図3－1，3－2に示す。

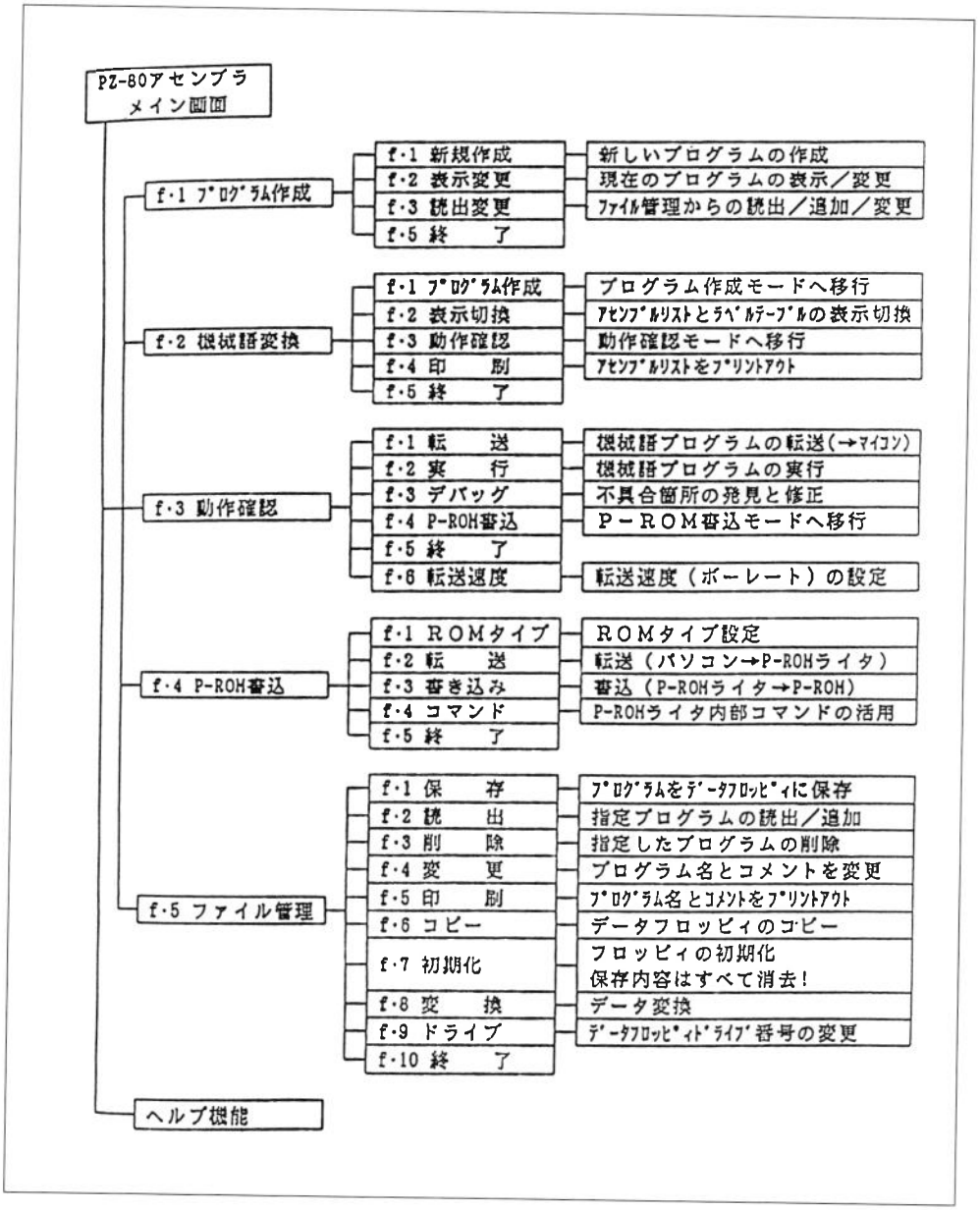


図3－1　PZ—80アセンブラソフトウェア構成

```
ドライブ A: のディスクのボリュームラベルはありません
ディレクトリは A:¥

COMMAND  COM     24161  87-10-23   0:00
Q_PZS    EXE     17198  91-06-03  20:54
Q_PZ0    EXE     35302  91-01-23  19:59
Q_PZ1    EXE    101462  91-06-03  20:46
Q_PZ2    EXE    107060  91-06-03  20:34
Q_PZ3    EXE     49408  91-01-24   8:46
Q_PZ4    EXE     51300  91-01-24   8:47
Q_PZ5    EXE     98822  91-01-24  19:29
Q_DATA           6657  90-06-01
INT64    COM     63750  91-01-24  17:02
INTWORK  COM       752  91-01-24  17:02
INST2    BAT       939  91-05-06   9:56
CONFIG3  DAT        27  90-06-01
CONFIG1  DAT        69  90-06-01
CONFIG2  DAT        48  90-06-01
INSTH    BAT       116  90-06-01
BRUN45A  EXE     92329  89-11-15   4:50
Q3       HLP      3055  90-06-01
INTDEL   COM       180  91-01-24   9:22
INST     BAT       458  91-05-06  10:07
PZ80     BAT        55  90-07-01
AUTOEXEC BAT        55  91-01-24   9:15
PRINT    SYS      5607  87-10-23   0:00
RSDRV    SYS      7998  87-10-23   0:00
CONFIG   SYS        69  90-06-01
SPEED    EXE     27458  87-10-23   0:00
      26 個のファイルがあります.
  461824 バイトが使用可能です.
```

図３－２　PZ—80 アセンブラシステムファイル

次に例としてマイコンモジュール PZ80—K1 上の LED を点滅するプログラムを作り，EP—ROM2716 に書き込み，マイコンに装着して実行する手順を示す。

(1)パソコンと ROM ライタ（PZ—W2）を RS—232C ケーブルで接続する。

(2)パソコン(PC9801)に PZ—80 のプログラムの入ったフロッピディスクを挿入して，コンピュータを立ち上げる。メイン画面が表示される。(図３－３)

(3)メイン画面で[f・1]キーを押して，［プログラム作成］を選択する。

(4)プログラム作成画面に変わるので，もう一度[f・1]キーを押して，[新規作成]を選択する。(図３－４）すると図３－５の格子状の表が表示される。

(5)この表の中にニモニック・ソースプログラムを入力する。(図３－５)

(6)[f・5]キーを押して[終了]する。次に保存するファイル名を入力する。ファイル名は FTEST1.MAC とした。

(7)メイン画面で[f・2]キーを押して，[機械語に変換] する。(図３－６) このときニモニック・ソースプログラムにエラーがあればエラーメッセージが表示される。

(8)[f・5]キーを押して [終了] する。

(9)メイン画面で[f・4]キーを押して [P—ROM 書き込み] を選択する。

(10)[f・1]キーを押して，書き込む [P—ROM のタイプ] を選択する。

(11)[f・2]キーを押して，プログラムをパソコンから P—ROM ライタに [転送] する。

(12) EP-ROM2716 を R-POM ライタに装着する。

(13)[f・3]キーを押して，プログラムを ER—ROM2716 に [書き込む]。

(14)EP—ROM2716 をターゲットに装着して実行する。

(15)動作ミスがあれば(1)からやり直す。

以上の操作により CN1，CN1. LAB，CN1. TEX の３つのファイルが作成される。CN1 は文書データで，データ長とアドレスが保存される。CN1. LAB は，ラベル情報，CN1. TEXはソース・プログラムと変換された機械語が圧縮されて保存される。CN の次にある数字は，作成された順に16進数で自動的に付けられる。

```
                ＰＺ－８０アセンブラ    Ver 2.3      ファイル：
プログラム作成    機械語変換      動作確認      ＰＲＯＭ書込    ファイル管理

      ●希望する ファンクションキー を押して下さい。
       f･1 作　成    Ｚ８０ プログラムを 作成します。
       f･2 機 械 語  Ｚ８０ プログラムを 機械語に変換します。
       f･3 動作確認  プログラムを転送し，動作を確認します。
       f･4 ＰＲＯＭ  ＰＲＯＭライタにプログラムを転送します。
       f･5 管　理    ファイルの保存と読出をします。

                                                  HELP 説明
 f･1 作 成    f･2 機 械 語   f･3 動作確認    f･4 ＰＲＯＭ   f･5 管 理
```

図３－３　メイン画面

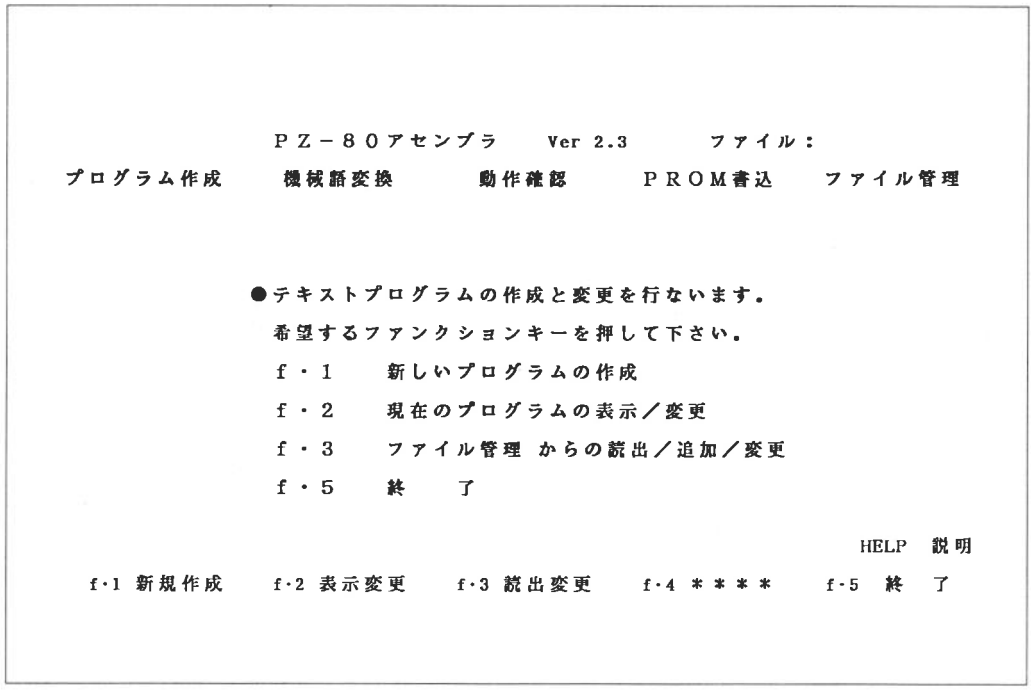


図３－４　プログラム作成画面

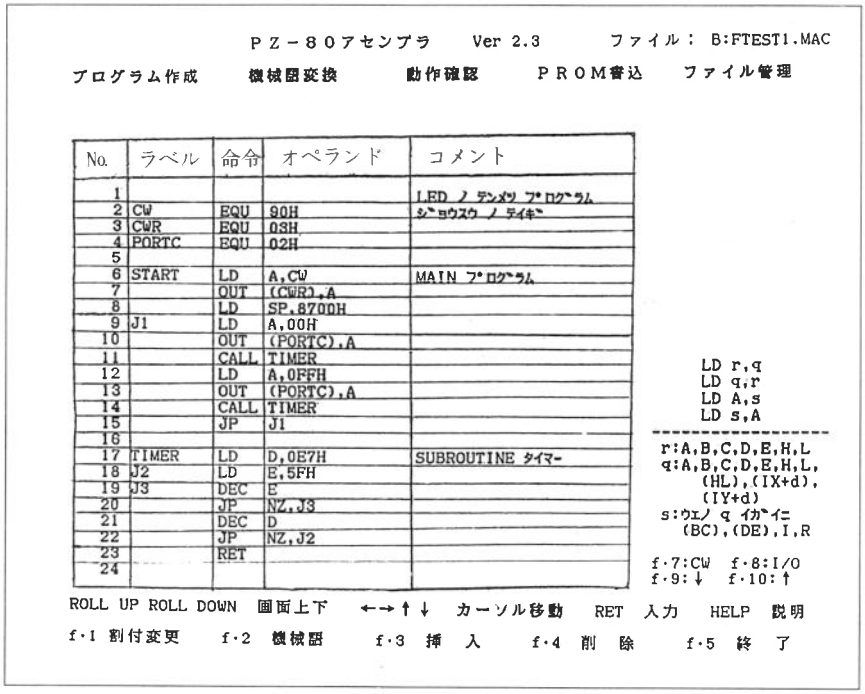


図３－５　プログラム入力画面

```
PACIFIC PZ-80 ASSEMBLER Ver.2.01
PROGRAM NAME :  B:FTEST1.MAC
DATE :          1992-09-17/15:06:43
                                                NAME _____________________

| No. | ラベル | メイレイ | オペランド | コメント | ハンチ | キカイゴ |
|-----|--------|----------|------------|----------|--------|----------|
|   1 |        |          |            | LED ノ テンメツ プログラム |        |          |
|   2 | CW     | EQU      | 90H        | ジョウスウ ノ テイギ |  0090  |          |
|   3 | CWR    | EQU      | 03H        |          |  0003  |          |
|   4 | PORTC  | EQU      | 02H        |          |  0002  |          |
|   5 |        |          |            |          |        |          |
|   6 | START  | LD       | A,CW       | MAIN プログラム |  0000  | 3E 90    |
|   7 |        | OUT      | (CWR),A    |          |  0002  | D3 03    |
|   8 |        | LD       | SP,8700H   |          |  0004  | 31 00 87 |
|   9 | J1     | LD       | A,00H      |          |  0007  | 3E 00    |
|  10 |        | OUT      | (PORTC),A  |          |  0009  | D3 02    |
|  11 |        | CALL     | TIMER      |          |  000B  | CD 18 00 |
|  12 |        | LD       | A,0FFH     |          |  000E  | 3E FF    |
|  13 |        | OUT      | (PORTC),A  |          |  0010  | D3 02    |
|  14 |        | CALL     | TIMER      |          |  0012  | CD 18 00 |
|  15 |        | JP       | J1         |          |  0015  | C3 07 00 |
|  16 |        |          |            |          |        |          |
|  17 | TIMER  | LD       | D,0E7H     | SUBROUTINE タイマー |  0018  | 16 E7    |
|  18 | J2     | LD       | E,5FH      |          |  001A  | 1E 5F    |
|  19 | J3     | DEC      | E          |          |  001C  | 1D       |
|  20 |        | JP       | NZ,J3      |          |  001D  | C2 1C 00 |
|  21 |        | DEC      | D          |          |  0020  | 15       |
|  22 |        | JP       | NZ,J2      |          |  0021  | C2 1A 00 |
|  23 |        | RET      |            |          |  0024  | C9       |
```

図３－６　機械語変換画面

３．１．２　XA80アセンブラ

XA80 は，システム・ロード社で開発された汎用アブソリュート・アセンブラで，アセンブルを行うのに必要な最小限の機能を備えていてとても使いやすい。対象 CPU はZ80 のみである。使用したシステムディスクのディレクトリを図３－７に示す。XA80 本体は XA80. EXE である。

```
 ドライブ A: のディスクにはボリュームラベルがありません
 ディレクトリは A:¥XA80

              <DIR>      92-06-17   17:14
..            <DIR>      92-06-17   17:14
XA80      EXE     33957  91-06-28   22:24
XA80      DOC      4422  91-07-16   16:11
TEST          <DIR>      92-06-17   17:15
        5 個のファイルがあります
   566272 バイトが使用可能です
```

図３－７　XA80 システムディスク

必要最小限の機能といえども，制御や計測などのプログラミングには十分に実用できるソフトである。

XA80の使用環境は，OS は MS—DOS Ver. 2.1 以上，メモリは384 Kbyte 以上必要である。この XA80 を使ったアセンブルの手順について，以下に示す。

例１　全体として一つにまとまった（サブルーチンを含む）プログラム

①　エディタ（MIFES等）でアセンブルの対象とするソース・プログラムを書く。この場合拡張子は ".ASM" としなければならない。XA80 でアセンブルするソース・ファイルは XA80 と同じディレクトリに存在する必要がある。

今回例として用いたプログラムの内容は，PZ—80 アセンブラの実行で用いたプログラムと全く同じで，LED を点滅させるというプログラムである。そのソース・プログラム "TSET1. ASM" を図３－８に示す。

```
TYPE TEST1.ASM
;*************************************
;*                                   *
;*   Sample program for Z80-CPU      *
;*   File name:       TEST1.ASM      *
;*                                   *
;*************************************
;
;     LED ﾃﾝﾒﾂ ﾌﾟﾛｸﾞﾗﾑ
;
CW     EQU       90H
CWR    EQU       03H
PORTC  EQU       02H
           ORG   0100H
;
START:     LD    A,CW
           OUT   (CWR),A
           LD    SP,8700H
J1:        LD    A,00H
           OUT   (PORTC),A
           CALL  TIMER
           LD    A,0FFH
           OUT   (PORTC),A
           CALL  TIMER
           JP    J1
;
;     ﾀｲﾏｰ ﾌﾟﾛｸﾞﾗﾑ
;
TIMER:
;     LOCAL      J2,J3
           LD    D,0E7H
J2:        LD    E,5FH
J3:        DEC   E
           JP    NZ,J3
           DEC   D
           JP    NZ,J2
           RET
           END
```

図３－８　ソース・プログラム（test1. ASM）

② アセンブルする。キーボードから，次の下線部を入力する。

XA80 ↙

パソコンの画面には，以下のように表示される。

```
A>XA80
Z-80/64180 Absolute Cross-Assembler (MS-DOS) Version 2.01
Copyright (C) System Load CO.,LTD. 1989-1991 All rights reserved.

Souce File[.ASM]    : TEST1
Object File[.HEX]   : TEST1
Symbol File[.SYM]   : TEST1   } 省略可
Listing File[.LST]  : TEST1
Pass1....
Pass2....

No Errors
```

上記のように，Souce File の下線部のファイル名を入力しリターンすると，順に Object File, Symbol File, Listing File とファイル名の入力を求めてくるので，その都度同じように入力し，Listing File のファイル名を入力しリターンすると変換が始まる。それぞれのファイルを作成し，ソース・プログラムにエラーがなければ No Errors と表示される。なお，Object File 以下のファイル名は，同じファイル名を使うときは省略することができ，その場合はリターンのみでよい。

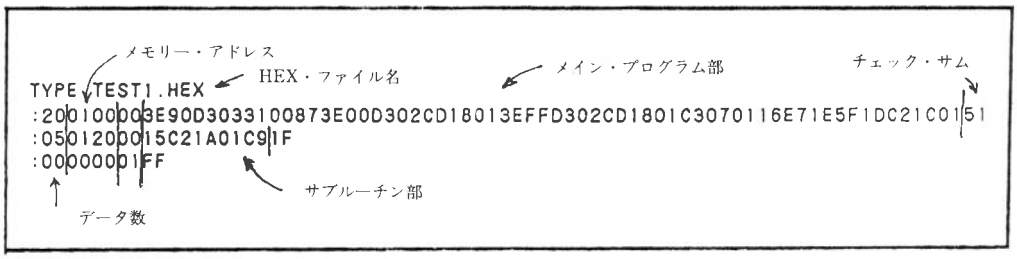


図3－9　オブジェクト・ファイル

```
TYPE TEST1.SYM          アドレス  ラベル
                           ↓       ↓
0090 CW                 0003 CWR          0107 J1
011A J2                 011C J3           0002 PORTC
0100 START              0118 TIMER
```

図3－10　シンボル・ファイル

```
TYPE TEST1.LST

                        ;*************************************
                        ;*                                   *
                        ;*    Sample program for Z80-CPU     *
                        ;*    File name:       TEST1.ASM     *
                        ;*                                   *
                        ;*************************************
                        ;
                        ;     LED ﾃﾝﾒﾂ ﾌﾟﾛｸﾞﾗﾑ
                        ;
0090                    CW      EQU       90H
0003                    CWR     EQU       03H
0002                    PORTC   EQU       02H
0100                            ORG       0100H
                        ;
0100    3E 90           START:  LD        A,CW
0102    D3 03                   OUT       (CWR),A
0104    31 00 87                LD        SP,8700H
0107    3E 00           J1:     LD        A,00H
0109    D3 02                   OUT       (PORTC),A
010B    CD 18 01                CALL      TIMER
010E    3E FF                   LD        A,0FFH
0110    D3 02                   OUT       (PORTC),A
0112    CD 18 01                CALL      TIMER
0115    C3 07 01                JP        J1
                        ;
                        ;     ﾀｲﾏｰ ﾌﾟﾛｸﾞﾗﾑ
                        ;
0118                    TIMER:
                        ;     LOCAL     J2,J3
0118    16 E7                   LD        D,0E7H
011A    1E 5F           J2:     LD        E,5FH
011C    1D              J3:     DEC       E
011D    C2 1C 01                JP        NZ,J3
0120    15                      DEC       D
0121    C2 1A 01                JP        NZ,J2
0124    C9                      RET
0125                            END
```

図３－11　リスト・ファイル

変換された各ファイルは Object File 図３－９，Symbol File 図３－10，Listing File 図３－11 に示す。ヘックス（HEX）ファイル（Object File）は変換された機械語プログラムを16進数のアスキー・コードで表示したファイルで，ROMライタへ，この HEX ファイルが転送される。転送された HEX ファイルは，ROMライタによってマイコンに書き込まれる。このインテル社形式の HEX ファイルは，左端にレコード・マーク（：）があり，次の２桁はデータ部のバイト数を表し（最大20H＝32バイト，１バイトを２文字で表示），次の４桁はデータの配置されるメモリのアドレス（16ビット表示），次にレコード・タイプ００があり，さらにデータがさきに表したバイト数だけ並び、最後に２桁のチェック・サムがくる。ここでいうデータとは、機械語プログラムその

ものであり，リスト・ファイルの中に見られる命令コードがここに並んでいる。

シンボル・ファイルは，プログラム中で使われたラベルとそのアドレスの一覧表である。

リスト・ファイルには，ソース・ファイルとその変換された機械語命令コードが表示されている。

XA80 では，ソース・ファイルを記述するとき，ラベルは各行の先頭から書くことが必要で，後で示す ZASM の場合のように，先頭に空白を設けるとエラーとなる。そのプログラムを図3—12，エラー表示された画面を図3—13に示す。

```
TYPE TEST1.ASM
;*************************************
;*                                   *
;*     Sampie program for Z80-CPU.   *
;*     File name:      TEST1.ASM     *
;*                                   *
;*************************************
;
;       LED ﾃﾝﾒﾂ ﾌﾟﾛｸﾞﾗﾑ
;
ラベルの前に   →CW       EQU       90H
空白を設けない →CWR      EQU       03H
             →PORTC    EQU       02H
               ORG      1000H
;
      →START:      LD          A,CW
                   OUT         (CWR),A
                   LD          SP,8700H
           →J1:    LD          A,00H
                   OUT         (PORTC),A
                   CALL        TIMER
                   LD          A,0FFH
                   OUT         (PORTC),A
                   CALL        TIMER
                   JP          J1
;
;       ﾀｲﾏｰ ﾌﾟﾛｸﾞﾗﾑ
;
      →TIMER:
;                   LOCAL      J2,J3
                   LD          D,0E7H
           →J2:    LD          E,5FH
           →J3:    DEC         E
                   JP          NZ,J3
                   DEC         D
                   JP          NZ,J2
                   RET
                   END
```

図3—12　エラーが発生するソース・プログラム

```
A>XA80
Z-80/64180 Absolute Cross-Assembler (MS-DOS) Version 2.01
Copyright (C) System Load CO.,LTD. 1989-1991 All rights reserved.

Souce File[.ASM]    : TEST1
Object File[.HEX]   :
Symbol File[.SYM]   :
Listing File[.LST]  :
Pass1....
Pass2....
TEST1.ASM 10: O             CW
TEST1.ASM 11: O             CWR
TEST1.ASM 12: O             PORTC
TEST1.ASM 15: O         START:
TEST1.ASM 16: U                     OUT         (CWR)
TEST1.ASM 18: O             J1:
TEST1.ASM 19: U                     OUT         (PORTC)
TEST1.ASM 20: U                     CALL        TIMER

TEST1.ASM 22: U                     OUT         (PORTC)
TEST1.ASM 23: U                     CALL        TIMER

TEST1.ASM 24: U                     JP          J1

TEST1.ASM 28: O         TIMER:

TEST1.ASM 31: O             J2:
TEST1.ASM 32: O             J3:
TEST1.ASM 33: U                     JP          NZ,J3

TEST1.ASM 35: U                     JP          NZ,J2

16 Error(s)     ↑エラーコード　O：オペランドがおかしい；U：シンボル名が見あたらない
```

図３—13　エラーが発生した変換

例２　主プログラム，定数定義部，サブルーチン・プログラムが，それぞれ独立のファイルになっている場合

①　それぞれのソース・ファイルをエディタで作る。主プログラムには，定数定義とサブルーチンの　INCLUDE　文を書いておく（注：INCLUDE　文の前には空白を設ける）。ファイル名は“TEST6. ASM”。図３—14～３—16にそれぞれのファイルを示す。

②　アセンブルする。変換結果を以下に示す。オブジェクト・ファイルを図３—17，シンボリック・ファイルを図３—18に示す。定数定義部，サブルーチンの独立したファイルが，INCLUDE 文によって，主プログラムに組み込まれて変換されている List を図３—19に示す。

```
A>XA80
Z-80/64180 Absolute Cross-Assembler (MS-DOS) Version 2.01
Copyright (C) System Load CO.,LTD. 1989-1991 All rights reserved.

Souce File[.ASM]    : TEST6
Object File[.HEX]   :
Symbol File[.SYM]   :
Listing File[.LST]  :
Pass1....
Pass2....

No Errors
```

```
TYPE TEST6.ASM
;************************************
;*     Sample program for Z80-CPU   *
;*     File name:      ¥TEST6.ASM   *
;************************************

 INCLUDE "Z80PIO.ASM"

;     LED ﾃﾝﾒﾂ ﾌﾟﾛｸﾞﾗﾑ
          ORG     100H
START:    LD      A,CW
          OUT     (CWR),A
          LD      SP,8700H
J1:       LD      A,00H
          OUT     (PORTC),A
          CALL    TIMER
          LD      A,0FFH
          OUT     (PORTC),A
          CALL    TIMER
          JP      J1

 INCLUDE "TIMERSUB.ASM"

          END
;************************************
```

図3—14　ソース・ファイル (test6. ASM)

```
TYPE Z80PIO.ASM
;**************************************
;*     File name:       Z80PIO.ASM    *
;*          PIO ｺﾝﾄﾛｰﾙ ｼﾞｮｳｽｳ          *
;**************************************
Z80PIO:
CW      EQU     90H
CWR     EQU     03H
PORTC   EQU     02H
;**************************************
```

図３—15　ソース・ファイル（z80pio. ASM）

```
TYPE TIMERSUB.ASM
;**************************************
;*      File name:     TIMERSUB.ASM   *
;*        ｻﾌﾞﾙｰﾁﾝ ﾌﾟﾛｸﾞﾗﾑ               *
;**************************************
;       ﾀｲﾏｰ ﾌﾟﾛｸﾞﾗﾑ
                ORG     200H
TIMER:
                LD      D,0E7H
J2:             LD      E,5FH
J3:             DEC     E
                JP      NZ,J3
                DEC     D
                JP      NZ,J2
                RET
;**************************************
```

図３—16　ソース・ファイル（timersub. ASM）

```
TYPE TEST6.HEX
:180100003E90D3033100873E00D302CD00023EFFD302CD0002C30701FD
:0D02000016E71E5F1DC2040215C20202C9EE
:00000001FF
```

図３—17　オブジェクト・ファイル（test6. HEX）

```
TYPE TEST6.SYM
0090 CW                 0003 CWR                0107 J1
0202 J2                 0204 J3                 0002 PORTC
0100 START              0200 TIMER              0000 Z80PIO
```

図３—18　シンボリック・ファイル（test6. SYM）

```
TYPE TEST6.LST
                        ;************************************
                        ;*     Sample program for Z80-CPU   *
                        ;*     File name:      ¥TEST6.ASM   *
                        ;************************************

                         INCLUDE "Z80PIO.ASM"
                        ;************************************
                        ;*     File name:      Z80PIO.ASM   *
                        ;*          PIO ｺﾝﾄﾛｰﾙ ｼﾞｮｳｽｳ        *
                        ;************************************
0000                    Z80PIO:
0090                    CW      EQU     90H
0003                    CWR     EQU     03H
0002                    PORTC   EQU     02H
                        ;************************************


                        ;     LED ﾃﾝﾒﾂ ﾌﾟﾛｸﾞﾗﾑ
0100                            ORG     100H
0100    3E 90           START:  LD      A,CW
0102    D3 03                   OUT     (CWR),A
0104    31 00 87                LD      SP,8700H
0107    3E 00           J1:     LD      A,00H
0109    D3 02                   OUT     (PORTC),A
010B    CD 00 02                CALL    TIMER
010E    3E FF                   LD      A,0FFH
0110    D3 02                   OUT     (PORTC),A
0112    CD 00 02                CALL    TIMER
0115    C3 07 01                JP      J1

                         INCLUDE "TIMERSUB.ASM"
                        ;************************************
                        ;*      File name:     TIMERSUB.ASM *
                        ;*        ｻﾌﾞﾙｰﾁﾝ ﾌﾟﾛｸﾞﾗﾑ             *
                        ;************************************
                        ;     ﾀｲﾏｰ ﾌﾟﾛｸﾞﾗﾑ
0200                            ORG     200H
0200                    TIMER:
0200    16 E7                   LD      D,0E7H
0202    1E 5F           J2:     LD      E,5FH
0204    1D              J3:     DEC     E
0205    C2 04 02                JP      NZ,J3
0208    15                      DEC     D
0209    C2 02 02                JP      NZ,J2
020C    C9                      RET
                        ;************************************


020D                            END
```

図3－19　サブルーチンを主プログラムに組み込んだリスト・ファイル

３．２　リロケータブル・アセンブラ

ZASMは，MS—DOS上で動作するリロケータブル・クロス・アセンブラである。アセンブラZA，リンカーZL，ライブラリアンZLIBなどプログラムを開発するために必要なプログラムから構成されている。アセンブラM80準拠となっていて，強力なマクロ機能をもったアセンブラです。アセンブル対象のCPUにはZ80のほかにHD64180（日立），8080，8085も，対象にしている。最近よく使われるワンチップマイコンTMPZ84C015（東芝）も，CPUはZ80であるのでZASMが，利用できる。ここでは，Z80マイコンの簡単な制御プログラムを材料にして，ZASMによるプログラム開発の具体例を示す。

最初にパソコンのフロッピ・ディスクのZASMのシステム・ディスクには，図３—20に示すように，エディタ（MIFES）とZASMのディレクトリをつくり，ZASMのディレクトリには，最低図３—21に示すファイルを収める。AUTOEXEC. BATとCONFIG. SYSファイルの内容を，それぞれ図３—22，３—23に例示する。つぎに，同じ内容で，構成の異なるプログラムの実行例を二三示す。プログラムは，ニモニック・コードで書かれているが，その内容の解説は省略する。（文中の小文字は任意に変更してよく，下線部はそのままキー・ボードから入力することを示す。）

```
COMMAND  COM     24931   89-11-27
ZASM         <DIR>       92-07-28
PRINT    SYS      5855   89-11-27
AUTOEXEC BAT        48   92-08-20
MIFES        <DIR>       92-08-21
CONFIG   SYS        95   92-08-20
```

図３—20　ZASMシステムデスクの内容

```
ディレクトリは A:¥ZASM

.            <DIR>      92-07-28
..           <DIR>      92-07-28
ZASM     BAT       145   90-08-18  アセンブル・リンクを行なうバッチファイル
ZLIB     EXE     32908   91-03-01  ライブラリアン
ZL       EXE     47332   91-03-01  リンカー
ZA       EXE     70402   91-03-01  アセンブラ
Z80INST  BDF      2264   90-11-09  Z80 トークン定義ファイル
Z80INST  INB     17778   90-11-09  Z80 ニーモニック定義ファイル
ZERRMSG1 JXT      3298   90-10-21  アセンブルエラーメッセージ
ZERRMSG2 JXT      2231   91-02-28  リンクエラメッセージ
ZERRMSG3 JXT      1721   90-07-16  ライブラリアンエラメッセージ
```

図３—21　ディレクトリZASMのファイル

```
A>TYPE AUTOEXEC.BAT
set path=A:¥;A:¥MSDOS;A:¥MIFES;A:¥ZASM
mifes
```

図3－22　AUTOEXEC. BAT

```
A>TYPE CONFIG.SYS
files=20
buffers=20
shell=A:¥COMMAND.COM A:¥ /e:512 /p
device=PRINT.SYS
device=RSDRV.SYS
```

図3－23　CONFIG. SYS

例1　全体として一つにまとまった（サブルーチンを含む）プログラム。

①　MIFES で，アセンブルの対象とするソース・プログラム（ファイル名：ftest1. MAC 図3－24）を書く。

プログラムの内容は，マイコンに接続されている発光ダイオードを点滅させるというプログラムで，アブソリュート・アセンブラの説明で用いたプログラムと同じ内容のプログラムである。

始めの部分で，定数を定義し，ラベル START 以下が主プログラムで，TIMER より以後は，サブルーチンである。

②　MIFES で，プロジェクト・ファイル（ファイル名：ftest1. PRJ 図3－25）を作る。

プロジェクト・ファイルは，ZASM に特有なファイルであり，このファイルに，アセンブルするファイル，リンクするファイル名を書き込み，ターゲット CPU，パスの設定などアセンブルやリンクの条件を決め，ライブラリ・ファイルの指定などを行ないます。アセンブルするファイル，リンクするファイルが，複数あるとは，このプロジェクト・ファイルに，列記する。今の例では，ftest1 とかく。

③　アセンブルする。キー・ボードから，次の下線部を入力する。

ZA ftest1. PRJ↓

パソコンの画面には，図3－26のように表示され，アセンブルが実行される。

```
        FTEST1.MAC

; ************************************
; *   Sample program for Z80-CPU     *
; *  File name:      FTEST1.MAC      *
; *     ＬＥＤ 点滅プログラム        *
; ************************************
; 定数定義
        CW      EQU     90H
        CWR     EQU     03H
        PORTC   EQU     02H
        ORG     1000H       ;先頭アドレス
;
    START:  LD      A       , CW
            OUT     (CWR)   , A
            LD      SP      , 8700H
        J1: LD      A       , 00H
            OUT     (PORTC),A
            CALL    TIMER     ; ｻﾌﾞﾙｰﾁﾝ･ｺｰﾙ
            LD      A       , 0FFH
            OUT     (PORTC),A
            CALL    TIMER     ; ｻﾌﾞﾙｰﾁﾝ･ｺｰﾙ
            JP      J1

; タイマー　サブルーチン
    TIMER:
            LD      D , 0E7H
        J2: LD      E , 5FH
        J3: DEC     E
            JP      NZ , J3
            DEC     D
            JP      NZ , J2
            RET
           END
```

図３―24　ソースプログラム（ftest1）

```
        FTEST1.PRJ
/z80             CPUは Z80
/out_hex         HEX ファイルを出力せよ
ftest1           ソースファイル名
```

図３―25　プロジェクト・ファイル（ftest1）

```
B>
B>ZA ftest1.PRJ
 ZA.EXE ver 3.09 Cross-Assembler for Z80,64180,8085,8080 ←対応CPU
        (c)Micro-org Inc 1989 1990.  updated Mar 01 1991
A:¥ZASM¥Z80INST.BDF
A:¥ZASM¥Z80INST.INB
        Heep memory [ 420144 ]
ZA [ FTEST1.MAC ]     ←ソース・ファイル名
Pass - 1          0 Error(s)       4133 bytes program.     FTEST1.MAC(33)
Pass - 2          0 Error(s)       4133 bytes program.     FTEST1.MAC(33)

        Heep memory [ 407792 ] -- [ 411936 ] bytes

        Heep memory [ 420144 ]
End of ZASM.  time(  0:06 )
```

図3－26　アセンブラ ZA. EXE の実行

ここで，入力の間違いをおかすと，それに対応したエラー・メッセージが表示される。(図3－27)

```
B>
B>ZA ftest1.PRK      ←入力エラー
 ZA.EXE ver 3.09 Cross-Assembler for Z80,64180,8085,8080
        (c)Micro-org Inc 1989 1990.  updated Mar 01 1991
A:¥ZASM¥Z80INST.BDF
A:¥ZASM¥Z80INST.INB
        Heep memory [ 420160 ]
ZA [ FTEST1.PRK ]

Error(1):        ソースファイルが見つかりません。
[ FTEST1.PRK ]

[FTEST1]

        Heep memory [ 411952 ] -- [ 411952 ] bytes

        Heep memory [ 420160 ]
End of ZASM.  time(  0:03 )
```

図3－27　ZA. EXE 実行時のエラー例

アセンブルが成功すると，二つのファイルが作られる。

ftest1. LST（リスト・ファイル）(図3－28)

ftest1. REL（リロケータブル・ファイル）

リスト・ファイルを見ると，ソース・ファイルとその変換された機械語命令コードが表示されている。リロケータブル・ファイルは，その名のとおり再配置可能なファイルで，リンクされるまで，そのプログラムがどのアドレスに置かれるかはわからない。テキスト・ファイルでないので，表示させても（図3－29），その内容を理解することはできない。

```
D>TYPE FTEST1.LST

 ZA ver 3.09 (c)Micro-org Inc.1990 [1992/ 9/20/ 0:58] [¥ZASMWORK¥FTEST1.MAC]page
[  1]
    1 0000
    2 0000                          ; ************************************
    3 0000                          ; *   Sample program for Z80-CPU     *
    4 0000                          ; *  File name:      FTEST1.MAC      *
    5 0000                          ; *     ＬＥＤ 点滅プログラム        *
    6 0000                          ; ************************************
    7 0000                          ; 定数定義
    8 0000  0090                            CW      EQU     90H
    9 0000  0003                            CWR     EQU     03H
   10 0000  0002                            PORTC   EQU     02H
   11 0000                                  ORG     1000H       ;先頭アドレス
   12 1000                          ;
   13 1000  3E 90                   START:  LD      A,CW
   14 1002  D3 03                           OUT     (CWR),A
   15 1004  31 00 87                        LD      SP,8700H
   16 1007  3E 00                   J1:     LD      A,00H
   17 1009  D3 02                           OUT     (PORTC),A
   18 100B  CD p1018                        CALL    TIMER           ; ｻﾌﾞﾙｰﾁﾝ･ｺｰﾙ
   19 100E  3E FF                           LD      A,0FFH
   20 1010  D3 02                           OUT     (PORTC),A
   21 1012  CD p1018                        CALL    TIMER           ; ｻﾌﾞﾙｰﾁﾝ･ｺｰﾙ
   22 1015  C3 p1007                        JP      J1
   23 1018
   24 1018                          ; タイマー　サブルーチン
   25 1018                          TIMER:
   26 1018  16 E7                           LD      D,0E7H
   27 101A  1E 5F                   J2:     LD      E,5FH
   28 101C  1D                      J3:     DEC     E
   29 101D  C2 p101C                        JP      NZ,J3
   30 1020  15                              DEC     D
   31 1021  C2 p101A                        JP      NZ,J2
   32 1024  C9                              RET
   33 1025                                      END

        Offset  Ext No  Line-No SEG     Name    == LABEL list ==

        1007'               16  CSEG    J1
        101A'               27  CSEG    J2
        101C'               28  CSEG    J3
        1000'               13  CSEG    START
        1018'               25  CSEG    TIMER
```

図３－28　リスト・ファイル（ftest1）

```
TYPE FTEST1.REL
,FTEST1am.        ¥ZASMWORK¥FTEST1.MAC(h6 j6 k[       6 [       6 [       6 [       6
["
```

図3—29　リロケータブル・ファイル（ftest1）

④　リンクする。（図3—30）キー・ボードから，次のように入力する。

ZL ftest1. PRJ↓

リンクの実行中に，リンクするリロケータブル・ファイル，出力するファイル（ヘックス　または，バイナリ・ファイル）名が，表示される。その他，リンクによって，マップ・ファイル，シンボル・ファイル，クロスリファレンス・ファイルがつくられる。

fteat1. HEX　（ヘックス・ファイル）

fteat1. BIN　（バイナリ・ファイル）

fteat1. MAP　（マップ・ファイル）

fteat1. SYM　（シンボル・ファイル）

fteat1. XRF　（クロスリファレンス・ファイル）

```
B>
B>ZL ftest1.PRJ /A
 ZL.EXE Ver3.09  Linker for Z80 HD64180 8080 8085
                 (c)Micro-org Inc 1989 1990.  updated Mar 01 1991.
        Heep memory [ 490224 ]
Linking. [ FTEST1.HEX ]

Reading [ FTEST1.REL ]

Link Pass 1.      0 Global(s). Heep memory 486080 byte(s).
Link Pass 2.      0 Global(s). Heep memory 486080 byte(s).
0 Global Symbol(s)
[ FTEST1.HEX ]  Symbol file. Cross reference. writting...
End of link.  time(  0:08 )
```

図3—30

バイナリー（BIN）ファイルは，ソース・ファイルを機械語に変換した，Z80が実行できるプログラムである。当然その内容をそのまま見ることは出来ない。無理に表示すれば，図3—31のようになる。

```
TYPE FTEST1.BIN
>責 1ε_
```

図3—31　バイナリ・ファイル（ftest. BIN）

ヘックス（HEX）ファイルは，BIN ファイルの機械語を16進数のアスキー・コードで表示したファイルで，普通パソコンで作ったプログラムを ROM ライタへ転送するときは，HEX ファイルの形で転送する。転送された HEX ファイルは，ROM ライタでマイコンに書き込むとき，再び BIN ファイルになおされる。HEX ファイルは，テキスト・ファイルであるので，図３－32に示す。アドレスの先頭は1000Hと，ソース・ファイルのなかで指定されているので(ORG 1000H)，0000H-00FFH までは，すべて00が並んでいる。マップ・ファイルを図３－33に示す。

例１の使いかたは，リンクする別のファイルがないので，本質的にはアブソリュート・アセンブラの場合と同じ方法と考えてよい。

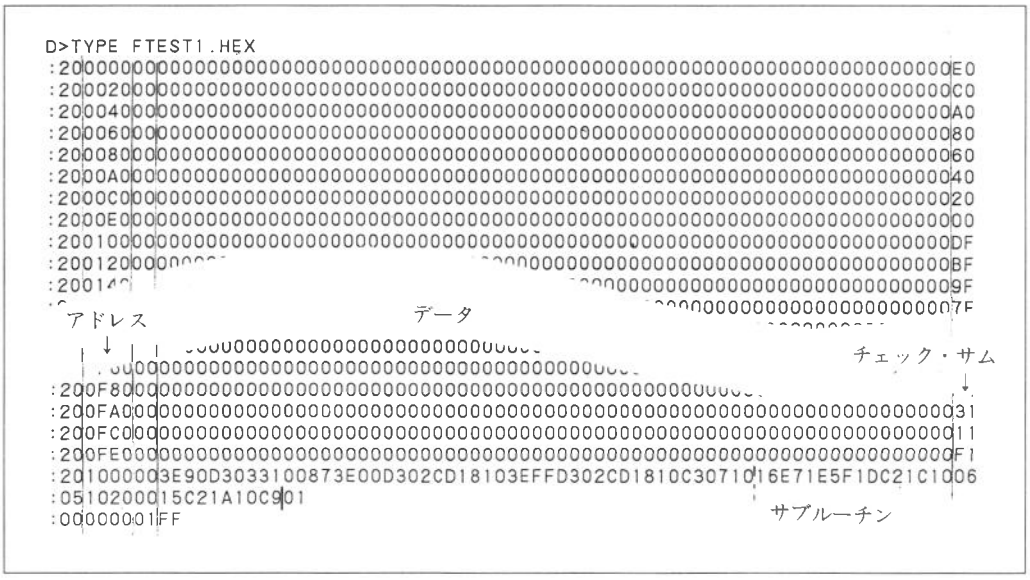


図３－32　ヘックス・ファイル（ftestl. HEX）

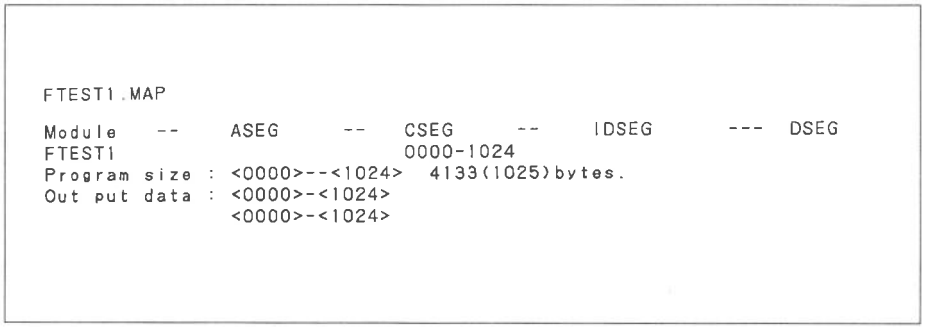


図３－33　マップ・ファイル（ftest1. MAP）

例2　ソース・プログラム中にマクロ命令を含むプログラム

アセンブルの手順は，例1と同じである。

①　ソース・プログラム（ファイル名：ftest2. MAC 図3—34）を書く。ソース・プログラムに

```
TYPE FTEST2.MAC
; ************************************
; *   Sample program for Z80-CPU     *
; *   File name:  FTEST2.MAC         *
; *   LED ﾃﾝﾒﾂ ﾌﾟﾛｸﾞﾗﾑ                 *
; *   (ｿｰｽﾌﾟﾛｸﾞﾗﾑﾆ ﾏｸﾛﾌﾟﾛｸﾞﾗﾑ ｦ ﾌｸﾑ)  *
; ************************************
;    ﾀｲﾏｰ ﾏｸﾛ ﾌﾟﾛｸﾞﾗﾑ                  : *
     TIMER   macro   dstep , estep   ; *
             LOCAL   J2 , J3         ; *
             LD       D , dstep      ; *
        J2:  LD       E , estep      ; *
        J3:  DEC      E              ; *
             JP       NZ, J3         ; *
             DEC      D              ; *
             JP       NZ, J2         ; *
             endm                    ; *
;.....................................*
;.....................................*
;       ｼﾞｮｳｽｳ                        . *
        CW      EQU     90H           ; *
        CWR     EQU     03H           ; *
        PORTC   EQU     02H           ; *
;.....................................*
;                                     *
       ORG     1000H                  ;
     START:    LD      A    , CW      ;
               OUT     (CWR) , A      ;
               LD      SP   , 8700H   ;
        J1:    LD      A    , 00H     ;
               OUT     (PORTC),A      ;
               TIMER   0e7h , 5fh     ;
               LD      A    , 0FFH    ;
               OUT     (PORTC),A      ;
               TIMER   0e7h , 5fh     ;
               JP      J1             ;
              END                     ;
;*************************************
```

図3—34　ソース・ファイル（ftest2. MAC)

マクロ命令，定数定義部，主プログラムを分けて書いてある。マクロ命令部で定義されたマクロ命令 TIMER は，主プログラム中では，一つのニモニック・コードとして使用できる。

② プロジェクト・ファイル（ファイル名：ftest2. PRJ 図３―35）を作る。

```
D>TYPE FTEST2.PRJ
/Z80
/OUT_BIN
ftest2
```

図３―35　プロジェクト・ファイル（ftest2. PRJ）

③ アセンブルする。ZA ftest2. PRJ↓

ftest2. LST（図３―36）をみると，アセンブルされた結果，マクロ命令部は主プログラムの該当マクロ命令の部分へ，繰込まれていることが解る。

④ リンクする。ZL ftest2. PRJ↓

作られたマップ・ファイル，ヘックス・ファイルを，図３―37，３―38に示す。

```
TYPE FTEST2.MAP
Module   --     ASEG      --    CSEG      --     IDSEG       ---   DSEG
FTEST2                         0000-1029
Program size : <0000>--<1029>  4138(102A)bytes.
Out put data : <0000>-<1029>
               <0000>-<1029>
```

図３―37　マップ・ファイル（ftest2.MAP）

```
D>TYPE FTEST2.HEX
:20000000000000000000000000000000000000000000000000000000000000000000000000000000E0
:20002000000000000000000000000000000000000000000000000000000000000000000000000000C0
:20004000000000000000000000000000000000000000000000000000000000000000000000000000A0
                (
:200FC000000000000000000000000000000000000000000000000000000000000000000000000011
:200FE000000000000000000000000000000000000000000000000000000000000000000000000000F1
:201000003E90D3033100873E00D30216E71E5F1DC20F1015C20D103EFFD30216E71E5F1D4C
:0A1020000C21F1015C21D10C30710F7
:00000001FF
```

図３―38　ヘックス・ファイル（ftest2. HEX）

```
TYPE FTEST2.LST

ZA ver 3.09 (c)Micro-org Inc.1990 [1992/ 9/20/ 1:31] [FTEST2.MAC]page[  1]
   1 0000                        ; ************************************
   2 0000                        ; *   Sample program for Z80-CPU     *
   3 0000                        ; *   File name:  FTEST2.MAC         *
   4 0000                        ; *   LED ﾃﾝﾒﾂ ﾌﾟﾛｸﾞﾗﾑ                *
   5 0000                        ; *   (ｿｰｽﾌﾟﾛｸﾞﾗﾑﾆ ﾏｸﾛﾌﾟﾛｸﾞﾗﾑ ﾖ ﾌｸﾑ)  *
   6 0000                        ; ************************************
   7 0000                        ;    ﾀｲﾏｰ ﾏｸﾛ ﾌﾟﾛｸﾞﾗﾑ                 : *
   8 0000                             TIMER   macro   dstep , estep  ; *
   9 0000                                     LOCAL   J2 , J3        ; *
  10 0000                                     LD       D , dstep     ; *
  11 0000                                J2:  LD       E , estep     ; *
  12 0000                                J3:  DEC      E             ; *
  13 0000                                     JP       NZ, J3        ; *
  14 0000                                     DEC      D             ; *
  15 0000                                     JP       NZ, J2        ; *
  16 0000                                     endm                   ; *
  17 0000                        ;.....................................*
  18 0000                        ;.....................................*
  19 0000                        ;        ｼﾞｮｳｽｳ                      .*
  20 0000  0090                           CW      EQU     90H       ; *
  21 0000  0003                           CWR     EQU     03H       ; *
  22 0000  0002                           PORTC   EQU     02H       ; *
  23 0000                        ;.....................................*
  24 0000                        ;                                     *
  25 0000                              ORG     1000H                  ;
  26 1000  3E 90                 START:  LD      A,CW                   ;
  27 1002  D3 03                         OUT     (CWR),A                ;
  28 1004  31 00 87                      LD      SP,8700H               ;
  29 1007  3E 00                 J1:     LD      A,00H                  ;
  30 1009  D3 02                         OUT     (PORTC),A              ;
  31 100B                                      TIMER  0e7h , 5fh        ;
  31 100B                        a             LOCAL   J2 , J3        ;  *
  31 100B  16 E7                 a             LD      D,0e7h         ;  *
  31 100D  1E 5F                 a J2:         LD      E,5fh          ;  *
  31 100F  1D                    a J3:         DEC     E              ;  *
  31 1010  C2 lp100F             a             JP      NZ,J3          ;  *
  31 1013  15                    a             DEC     D              ;  *
  31 1014  C2 lp100D             a             JP      NZ,J2          ;  *
  31 1017                        a             endm                   ;  *
  32 1017  3E FF                               LD      A,0FFH         ;
  33 1019  D3 02                               OUT     (PORTC),A      ;
  34 101B                                      TIMER  0e7h , 5fh        ;
  34 101B                        a             LOCAL   J2 , J3        ;  *
  34 101B  16 E7                 a             LD      D,0e7h         ;  *
  34 101D  1E 5F                 a J2:         LD      E,5fh          ;  *
  34 101F  1D                    a J3:         DEC     E              ;  *
  34 1020  C2 lp101F             a             JP      NZ,J3          ;  *
  34 1023  15                    a             DEC     D              ;  *
  34 1024  C2 lp101D             a             JP      NZ,J2          ;  *
  34 1027                        a             endm                   ;  *
  35 1027  C3 p1007                            JP      J1             ;
  36 102A                                      END                    ;

      Offset  Ext No  Line-No SEG     Name     == LABEL list ==

      1007'               29  CSEG    J1
      1000'               26  CSEG    START
```

図3—36　リスト・ファイル (ftest2. LST)

例３　マクロ命令部，定数定義部をそれぞれ独立したソース・プログラムとし，別途に主プログラムがある場合。

① MIFES で，主プログラム（ファイル名：ftest3. MAC），マクロ命令部（ファイル名：timer. MAC），定数定義部（ファイル名：z80pio. MAC）図３－39(a)，(b)，(c)を書く。
主プログラムの始めの部分に，INCLUDE 文
　INCLUDE timer. MAC
　INCLUDE z80pio. MAC　　と書いておく。
② プロジェクト・ファイル（ファイル名：ftest3. PRJ）を作る。ソースファイル名は，ftest3 一つのみで良い。

```
TYPE FTEST3.MAC
; ************************************
; *    Sample program for Z80-CPU    *
; *    File name:       FTEST3.MAC   *
; *    LED ﾃﾝﾒﾂ ﾌﾟﾛｸﾞﾗ                 *
; *          (ｲﾝｸﾙｰﾄﾌﾞﾝ ﾖ ﾌｸﾑ)        *
; ************************************
                                      ;
     INCLUDE TIMER.MAC                ;
                                      ;
     INCLUDE Z80PIO.MAC               ;
;.....................................;
     START:   LD      A    , CW      ;
              OUT     (CWR) , A      ;
              LD      SP   , 8700H   ;
         J1:  LD      A    , 00H     ;
              OUT     (PORTC),A      ;
              TIMER   0E7H , 5FH     ;
              LD      A    , 0FFH    ;
              OUT     (PORTC),A      ;
              TIMER   0E7H , 5FH     ;
              JP      J1             ;
             END                     ;
;************************************
```

図３－39(a)　プログラムファイル（ftest3. MAC）

```
TYPE TIMER.MAC
; ************************************
; *  File name:    TIMER.MAC          *
; *          ﾏｸﾛﾌﾟﾛｸﾞﾗﾑ              *
;.....................................
;     ﾀｲﾏｰ ﾏｸﾛ ﾌﾟﾛｸﾞﾗﾑ                ;
      CSEG                           ;
                                     ;
      TIMER   macro   dstep , estep  ;
              local   j2 , j3        ;
              ld       d , dstep     ;
         j2:  ld       e , estep     ;
         j3:  dec      e             ;
              jp       nz , j3       ;
              dec      d             ;
              jp       nz , j2       ;
              endm                   ;
;************************************
```

図3－39(b)　マクロ命令ファイル（timer. MAC)

```
D>TYPE Z80PIO.MAC
; ************************************
; *  File name:    Z80PIO.MAC         *
; *     PIOｺﾝﾄﾛｰﾙｼﾞｮｳｽｳ ﾌﾟﾛｸﾞﾗﾑ      *
; .....................................
  CSEG                               ;
  GLOBAL     CW,CWR,PORTC            ;
  Z80PIO::   CW      EQU     90H     ;
             CWR     EQU     03H     ;
             PORTC   EQU     02H     ;
;************************************
```

図3－39(c)　定数定義ファイル（Z80pio. MAC)

③　アセンブルする。　ZA ftest3. PRJ↓

リスト・ファイル図3－40を参照すると，主プログラムのINCLUDE文の場所に，該当のファイルのプログラムが挿入され，マクロ命令，定数はそれぞれ置き換わっている。

④　リンクする。　ZL　ftest3. PRJ↓

```
TYPE FTEST3.LST

ZA ver 3.09 (c)Micro-org Inc.1990 [1992/ 9/20/12:23] [FTEST3.MAC]page[  1]
   1 0000                       ; ***************************************
   2 0000                       ; *   Sample program for Z80-CPU        *
   3 0000                       ; *   File name:       FTEST3.MAC       *
   4 0000                       ; *   LED ﾃﾝﾒﾂ ﾌﾟﾛｸﾞﾗﾑ                    *
   5 0000                       ; *          (ｲﾝｸﾙｰﾄﾞﾌｧｲﾙ ｦ ﾌｸﾑ)          *
   6 0000                       ; ***************************************
   7 0000                                                               ;
   8 0000                            INCLUDE TIMER.MAC                  ;
   1 0000                       ; ***************************************
   2 0000                       ; *  File name:    TIMER.MAC            *
   3 0000                       ; *          ﾏｸﾛﾌﾟﾛｸﾞﾗﾑ                   *
   4 0000                       ;.......................................
   5 0000                       ;    ﾀｲﾏｰ ﾏｸﾛ ﾌﾟﾛｸﾞﾗﾑ                  ;
   6 0000                            CSEG                              ;
   7 0000                                                              ;
   8 0000                            TIMER   macro   dstep , estep     ;
   9 0000                                    local   j2 , j3           ;
  10 0000                                    ld       d , dstep        ;
  11 0000                               j2:  ld       e , estep        ;
  12 0000                               j3:  dec      e                ;
  13 0000                                    jp       nz , j3          ;
  14 0000                                    dec      d                ;
  15 0000                                    jp       nz , j2          ;
  16 0000                                    endm                      ;
  17 0000                       ;***************************************
   9 0000                                                               ;
  10 0000                            INCLUDE Z80PIO.MAC                 ;
   1 0000                       ; ***************************************
   2 0000                       ; *  File name:    Z80PIO.MAC           *
   3 0000                       ; *     PIOｺﾝﾄﾛｰﾙｼﾞｮｳｽｳ ﾌﾟﾛｸﾞﾗﾑ          *
   4 0000                       ; .......................................
   5 0000                         CSEG                                  ;
   6 0000                         GLOBAL    CW,CWR,PORTC                ;
   7 0000  0090                   Z80PIO::  CW      EQU     90H         ;
   8 0000  0003                             CWR     EQU     03H         ;
   9 0000  0002                             PORTC   EQU     02H         ;
  10 0000                       ;***************************************
  11 0000                       ;.......................................;
  12 0000  3E 90                START:  LD      A,CW              ;
  13 0002  D3 03                        OUT     (CWR),A           ;
  14 0004  31 00 87                     LD      SP,8700H          ;
  15 0007  3E 00                J1:     LD      A,00H             ;
  16 0009  D3 02                        OUT     (PORTC),A         ;
  17 000B                                TIMER  0E7H , 5FH        ;
  17 000B                       a               local   j2 , j3    ;
  17 000B  16 E7                a               ld      d,0E7H    ;
  17 000D  1E 5F                a j2:           ld      e,5FH     ;
  17 000F  1D                   a j3:           dec     e         ;
  17 0010  C2 lp000F            a               jp      nz,j3     ;
  17 0013  15                   a               dec     d         ;
  17 0014  C2 lp000D            a               jp      nz,j2     ;
  17 0017                       a             endm                ;
  18 0017  3E FF                        LD      A,0FFH            ;
  19 0019  D3 02                        OUT     (PORTC),A         ;
  20 001B                                TIMER  0E7H , 5FH        ;
  20 001B                       a               local   j2 , j3    ;
  20 001B  16 E7                a               ld      d,0E7H    ;
  20 001D  1E 5F                a j2:           ld      e,5FH     ;
  20 001F  1D                   a j3:           dec     e         ;
```

```
ZA ver 3.09 (c)Micro-org Inc.1990 [1992/ 9/20/12:23] [FTEST3.MAC]page[  2]
  20 0020  C2 lp001F          a          jp       nz,j3              ;
  20 0023  15                 a          dec      d                  ;
  20 0024  C2 lp001D          a          jp       nz,j2              ;
  20 0027                     a               endm                          ;
  21 0027  C3 p0007                      JP       J1                 ;
  22 002A                                END                                 ;

      Offset  Ext No  Line-No SEG     Name     == LABEL list ==

      FFFFFFFF'                  10            CW
      FFFFFFFF'                  10            CWR
      0007'               15     CSEG     J1
      FFFFFFFF'                  10            PORTC
      0000'               12     CSEG     START
      0000"               10     CSEG     Z80PIO
```

図3—40　リスト・ファイル（ftest3. LST）

作られたマップ・ファイル，シンボル・ファイル，クロスリファレンス・ファイル，ヘックス・ファイルを図3—41(a)，(b)，(c)，(d)に示す。一連の作業で作られたファイルのリストを図3—42に示す。

この場合の注意すべきことは，ZASMでは，マクロ命令ファイルや定数定義ファイルは，それぞれ単独にアセンブルして機械語のリロケータブル・ファイルにできないこと，マクロ命令，定数をグローバル・ラベルとして，主プログラムの中で引用できないことである。そのため，リロケータブル・アセンブラの特徴である，リロケータブル・ファイルをリンクして目的のファイルを作るという作業ができない。

```
D>TYPE FTEST3.MAP
Module   --    ASEG     --   CSEG     --    IDSEG      ---  DSEG
FTEST3                     0000-0029
Program size : <0000>--<0029>  42(  2A)bytes.
Out put data : <0000>-<0029>
               <0000>-<0029>
```

図3—41(a)　マップ・ファイル（ftest3. MAP）

```
D>TYPE FTEST3.SYM
0000 Z80PIO
```

図３—41(b)　シンボル・ファイル（ftest3. SYM)

```
D>TYPE FTEST3.XRF
Z80PIO                          [ FTEST3 ]
```

図３—41(c)　クロスリファレンス・ファイル（fitest3. XRF)

```
D>TYPE FTEST3.HEX
:200000003E90D3033100873E00D30216E71E5F1DC20F0015C20D003EFFD30216E71E5F1D7C
:0A002000C21F0015C21D00C3070037
:00000001FF
```

図３—41(d)　ヘックス・ファイル（fiest3. HEX)

```
D>DIR FTEST3.*

 ドライブ D: のディスクのボリュームラベルはありません.
 ディレクトリは D:¥ZASMWORK

FTEST3   MAP      213  92-09-20   12:24
FTEST3   REL      225  92-09-20   12:23
FTEST3   LST     4588  92-09-20   12:23
FTEST3   BIN       42  92-09-20   12:24
FTEST3   SYM       12  92-09-20   12:24
FTEST3   XRF       23  92-09-20   12:24
FTEST3   HEX      123  92-09-20   12:23
FTEST3   MAC      944  92-09-20    1:50
FTEST3   PRJ        9  92-09-20   12:23
        9 個のファイルがあります.
    79872 バイトが使用可能です.
```

図３—42　ftest ファイル

例４　主プログラム，サブルーチン，定数定義部が，それぞれ独立のファイルになっている場合。
（例３のマクロ命令をサブルーチンに代えた場合で，リンク操作を経る例）

①それぞれのソース・ファイルをエディタでつくる。（図３—43(a)，(b)）
（ファイル名：ftest6. MAC timersub. MAC z80pio. MAC）
主プログラムには，定数定義の INCLUDE 文をかいておく。

```
TYPE FTEST6.MAC
; ************************************
; *   Sample program for Z80-CPU     *
; *  File name:   FTEST6.MAC         *
; *          ｻﾌﾞﾙｰﾁﾝ                 *
; *          ｼﾞｮｳｽｳ   ﾖ ﾌｸﾑ          *
; ************************************
    INCLUDE  Z80PIO.MAC  ;ｲﾝｸﾙｰﾄﾞﾌｧｲﾙ ;
; ************************************;
    CSEG                              ;
    EXT      TIMER                    ;
                                      ;
    START:   LD       A      ,CW      ;
             OUT      (CWR)  ,A       ;
             LD       SP     ,8700H   ;
       J1:   LD       A      ,00H     ;
             OUT      (PORTC),A       ;
             CALL     TIMER           ;
             LD       A      ,0FFH    ;
             OUT      (PORTC),A       ;
             CALL     TIMER           ;
             JP       J1              ;
            END                       ;
;************************************
```

図3－43(a)　主プログラムファイル（ftest6. MAC)

```
D>TYPE TIMERSUB.MAC
; ************************************
; * File name: TIMERSUB.MAC          *
; *          ｻﾌﾞﾙｰﾁﾝ ﾌﾟﾛｸﾞﾗﾑ          *
; ************************************
;    ﾀｲﾏｰ  ﾌﾟﾛｸﾞﾗﾑ                    ;
    CSEG                              ;
    GLOBAL  TIMER                     ;
    TIMER::                           ;
            ld       d , 0e7h         ;
       j2:  ld       e , 5fh          ;
       j3:  dec      e                ;
            jp       nz , j3          ;
            dec      d                ;
            jp       nz , j2          ;
            ret                       ;
           end                        ;
;************************************
```

図3－43(b)　サブルーチンプログラム（timersub. MAC)

② プロジェクト・ファイル（ファイル名：ftest6. PRJ 図３—44）を作る。今度はアセンブル，リンクする対象のファイルとして，主プログラムファイル，サブルーチンファイル名を書く。

```
D>TYPE FTEST6.PRJ
/z80
/XRF_LINE
/cseg 0
ftest6          ソース・ファイル
/Cseg 40H
timersub        サブルーチン・ファイル
```

図３—44　プロジェクト・ファイル（ftest6. PRJ）

③ アセンブルする。　ZA ftest6. PRJ↓（図３—45）

主プログラムファイル，サブルーチンファイルは，個別にアセンブルされ，リロケータブル・ファイルが作られる。定数定義ファイルは，主プログラムファイルに，組み込まれる。それぞれのリスト・ファイルを，図３—46(a)，(b)に示す。

```
B>ZA ftest6.PRJ /A
 ZA.EXE ver 3.09 Cross-Assembler for Z80,64180,8085,8080
        (c)Micro-org Inc 1989 1990.  updated Mar 01 1991
A:¥ZASM¥Z80INST.BDF
A:¥ZASM¥Z80INST.INB
        Heep memory [ 419952 ]
ZA [ FTEST6.MAC ]                メイン・ファイルのアセンブル
        INCLUDE Z80PIO.MAC(11)
Pass - 1         0 Error(s)          24 bytes program.     FTEST6.MAC(22)
        INCLUDE Z80PIO.MAC(11)
Pass - 2         0 Error(s)          24 bytes program.     FTEST6.MAC(22)

        Heep memory [ 411744 ] -- [ 411744 ] bytes

                                     サブルーチンのアセンブル
ZA [ TIMERSUB.MAC ]
Pass - 1         0 Error(s)          13 bytes program.     TIMERSUB.MAC(16)
Pass - 2         0 Error(s)          13 bytes program.     TIMERSUB.MAC(16)

        Heep memory [ 411744 ] -- [ 411744 ] bytes

        Heep memory [ 419952 ]
End of ZASM.  time(  0:07 )
```

図３—45　アセンブルの手順

```
TYPE FTEST6.LST

ZA ver 3.09 (c)Micro-org Inc.1990 [1992/ 9/20/12:19] [FTEST6.MAC]page[  1]
    1 0000                          ; ************************************
    2 0000                          ; *   Sample program for Z80-CPU     *
    3 0000                          ; *  File name:   FTEST6.MAC         *
    4 0000                          ; *          ｻﾌﾞﾙｰﾁﾝ                 *
    5 0000                          ; *          ｼﾞｮｳｽｳ   ﾖ ﾌｸﾑ           *
    6 0000                          ; ************************************
    7 0000                              INCLUDE  Z80PIO.MAC  ;ｲﾝｸﾙｰﾄﾞﾌｧｲﾙ ;
    1 0000                          ; ************************************
    2 0000                          ; *  File name:    Z80PIO.MAC        *
    3 0000                          ; *     PIOｺﾝﾄﾛｰﾙｼﾞｮｳｽｳ ﾌﾟﾛｸﾞﾗﾑ      *
    4 0000                          ; ************************************
    5 0000                            CSEG                              ;
    6 0000                            GLOBAL    CW,CWR,PORTC            ;
    7 0000  0090                      Z80PIO::  CW      EQU     90H     ;
    8 0000  0003                                CWR     EQU     03H     ;
    9 0000  0002                                PORTC   EQU     02H     ;
   10 0000                          ;************************************
    8 0000                          ; ----------------------------------;
    9 0000                              CSEG                            ;
   10 0000                              EXT       TIMER                 ;
   11 0000                                                              ;
   12 0000  3E 90                   START:   LD       A,CW             ;
   13 0002  D3 03                            OUT      (CWR),A          ;
   14 0004  31 00 87                         LD       SP,8700H         ;
   15 0007  3E 00                   J1:      LD       A,00H            ;
   16 0009  D3 02                            OUT      (PORTC),A        ;
   17 000B  CD #0000                         CALL     TIMER            ;
   18 000E  3E FF                            LD       A,0FFH           ;
   19 0010  D3 02                            OUT      (PORTC),A        ;
   20 0012  CD #0000                         CALL     TIMER            ;
   21 0015  C3 p0007                         JP       J1               ;
   22 0018                                        END                   ;



      Offset  Ext No  Line-No SEG     Name     == LABEL list ==

      FFFFFFFF'                  7             CW
      FFFFFFFF'                  7             CWR
      0007'              15      CSEG    J1
      FFFFFFFF'                  7             PORTC
      0000'              12      CSEG    START
                0                Extern  TIMER
      0000"               7      CSEG    Z80PIO
```

図3—46(a)　主プログラムのリスト・ファイル (ftest6. LST)

```
D>TYPE TIMERSUB.LST

ZA ver 3.09 (c)Micro-org Inc.1990 [1992/ 9/20/12:19] [TIMERSUB.MAC]page[  1]
   1 0000                       ; **********************************
   2 0000                       ; * File name: TIMERSUB.MAC         *
   3 0000                       ; *          ｻﾌﾞﾙｰﾁﾝ ﾌﾟﾛｸﾞﾗﾑ          *
   4 0000                       ; ..................................
   5 0000                       ;     ﾀｲﾏｰ  ﾌﾟﾛｸﾞﾗﾑ                 ;
   6 0000                            CSEG                          ;
   7 0000                            GLOBAL  TIMER                 ;
   8 0000                            TIMER::                       ;
   9 0000  16 E7                        ld      d,0e7h          ;
  10 0002  1E 5F                j2:     ld      e,5fh           ;
  11 0004  1D                   j3:     dec     e               ;
  12 0005  C2 p0004                     jp      nz,j3           ;
  13 0008  15                           dec     d               ;
  14 0009  C2 p0002                     jp      nz,j2           ;
  15 000C  C9                           ret                     ;
  16 000D                               end                             ;

      Offset  Ext No  Line-No SEG     Name    == LABEL list ==

      0000"             8     CSEG    TIMER
      0002'            10     CSEG    j2
      0004'            11     CSEG    j3
```

図３－46(b)　サブルーチンのリストファイル（timersub. LST）

④　リンクする。　ZL ftest6. PRJ↓（図３－47）

二つのリロケータブル・ファイルはリンクされ，一つの機械語ファイルになる。HEXファイル，MAP ファイル，SYM ファイル，XR ファイルを，図３－48(a)，(b)，(c)，(d)に示す。

```
B>ZL ftest6.PRJ /A
 ZL.EXE Ver3.09  Linker for Z80 HD64180 8080 8085
                 (c)Micro-org Inc 1989 1990.  updated Mar 01 1991.
        Heep memory [ 490032 ]
Linking. [ FTEST6.BIN ]

Reading [ FTEST6.REL ]   } リンクされる REL ファイル
Reading [ TIMERSUB.REL ] }

Link Pass 1.      2 Global(s). Heep memory 490032 byte(s).
Link Pass 2.      2 Global(s). Heep memory 490032 byte(s).
2 Global Symbol(s)
[ FTEST6.BIN ]          < 0000 > - < 004C >
Symbol file. Cross reference. writting...
End of link.  time(  0:04 )
```

図３－47　リンクの手順

```
D>TYPE FTEST6.HEX
:180000003E90D3033100873E00D302CD40003EFFD302CD4000C3070083
:0D00400016E71E5F1DC2440015C24200C934
:00000001FF
```

図3—48(a)　ヘックス・ファイル

```
D>TYPE FTEST6.MAP
Module    --     ASEG      --    CSEG      --     IDSEG       ---  DSEG
FTEST6                           0000-0017
TIMERSUB                         0040-004C
Program size : <0000>--<004C>   77(   4D)bytes.
Out put data : <0000>-<004C>
               <0000>-<0017>
               <0040>-<004C>
```

図3—48(b)　マップ・ファイル

```
D>TYPE FTEST6.SYM
0040 TIMER       0000 Z80PIO
```

図3—48(c)　シンボル・ファイル

```
D>TYPE FTEST6.XRF
TIMERSUB.MAC      8:   TIMER:: FTEST6
FTEST6.MAC        7:   Z80PIO::
```

図3—48(d)　クロスリファレンス・ファイル

例5　主プログラム，サブルーチン，定数定義部が，それぞれ独立のファイルになっている場合。（例4と同じ場合で，ライブラリを利用する例）

ライブラリとは，サブルーチンが沢山あって，これがなかなか便利で，汎用にしばしば使えるぞという場合，そのサブルーチンを個々にリンクして使うのではなく，まとめてアセンブルして一つのリロケータブル・ファイルにしたものである。リンクのとき，ライブラリの中から，お呼びのかかったサブルーチンのみ自動的に呼び出されリンクされるという便利な点がある。

①　それぞれのソース・ファイルをエディタでつくる。（図3—49）

定数定義ファイルは主プログラム中の INCLUDE 文で繰り込み処理する。

```
D>TYPE FTEST5.MAC
; ************************************
; *   Sample program for Z80-CPU     *
; *   File name:   FTEST5.MAC        *
; *          ﾗｲﾌﾞﾗﾘﾖｳ ﾂｶｳ             *
; ************************************
     INCLUDE  Z80PIO.MAC  ;ｲﾝｸﾙｰﾄﾞ      ;
;.....................................;;
     EXT      TIMER       ;ｶﾞｲﾌﾞﾗﾍﾞﾙ   ;
                                       ;
     START:   LD      A      ,CW       ;
              OUT     (CWR)  ,A        ;
              LD      SP     ,8700H    ;
        J1:   LD      A      ,00H      ;
              OUT     (PORTC),A        ;
              CALL    TIMER            ;
              LD      A      ,0FFH     ;
              OUT     (PORTC),A        ;
              CALL    TIMER            ;
              JP      J1               ;
             END                       ;
;************************************

D>TYPE Z80PIO.MAC
; ************************************
; *  File name:    Z80PIO.MAC        *
; *     PIOｺﾝﾄﾛｰﾙｼﾞｮｳｽｳ ﾌﾟﾛｸﾞﾗﾑ       *
; ....................................
  CSEG                                 ;
  GLOBAL    CW,CWR,PORTC               ;
  Z80PIO::  CW      EQU     90H        ;
            CWR     EQU     03H        ;
            PORTC   EQU     02H        ;
;************************************
```

```
D>TYPE TIMERSUB.MAC
; ************************************
; * File name: TIMERSUB.MAC          *
; *          ｻﾌﾞﾙｰﾁﾝ ﾌﾟﾛｸﾞﾗﾑ          *
; ....................................
;     ﾀｲﾏｰ  ﾌﾟﾛｸﾞﾗﾑ                    ;
      CSEG                            ;
      GLOBAL  TIMER                   ;
      TIMER::                         ;
              ld      d , 0e7h        ;
         j2:  ld      e , 5fh         ;
         j3:  dec     e               ;
              jp      nz , j3         ;
              dec     d               ;
              jp      nz , j2         ;
              ret                     ;
             end                      ;
;************************************
```

図3－49　ソース・ファイル

② ライブラリの制作

(1)ソース・ファイルのあるものを，ライブラリにしたい場合は，先にライブラリ化する。そのために，そのソース・ファイルをアセンブルして，リロケータブル・ファイルにする。

(2)ZASM では，ライブラリ作成用のテキスト・ファイル（リスポンス・ファイル　拡張子　.MKL）を，作らなければならない。リスポンス・ファイルには，ライブラリ化するリロケータブル・ファイル名，作られるライブラリ名を書く。例を図3－50に示す。

(3)ライブラリの制作（ライブラリアンの実行）

ZLIB ftest5. MKL↓

上のように入力すると，図3－51のように表示され，ライブラリがつくられる。

```
D>TYPE FTEST5.MKL
/LIB_NAME ftest5.LIB
timersub
```

図3－50　リスポンス・ファイル

```
B>ZLIB ftest5.MKL /A
 ZLIB.EXE Ver3.09 Library Manager for ZASM.
                  (c)Micro-org Inc 1990.   updated Mar 01 1991.
[ TIMERSUB.REL ] 1つのﾌｧｲﾙをﾗｲﾌﾞﾗﾘｰに組み込みました。
End of ZLIB.  time(  0:02 )
```

図3－51　ライブラリ化の手順

③　プロジェクト・ファイル（ファイル名：ftest5. PRJ 図３—52）を作る。
プロジェクト・ファイルには，主プログラムのソース・ファイル名，リンクするライブラリ名を列ねて書く。（図３—53）

```
D>TYPE FTEST5.PRJ
/z80
ftest5
/LIBALL ftest5.LIB
```

図３—52　プロジェクト・ファイル

```
D>TYPE FTEST5.LIB
/TIMERSUB.      TIMERSUB.MAC(16)hTIMER6 j6 k[      6 [      6 e
```

図３—53　ライブラリ・ファイル

④　アセンブルする。　　ZA ftest5. PRJ↓（図３—54）
主プログラムのソース・ファイルは，アセンブルされる。

```
B>ZA ftest5.PRJ /A
 ZA.EXE ver 3.09 Cross-Assembler for Z80,64180,8085,8080
        (c)Micro-org Inc 1989 1990.  updated Mar 01 1991
A:¥ZASM¥Z80INST.BDF
A:¥ZASM¥Z80INST.INB
        Heep memory [ 420144 ]
ZA [ FTEST5.MAC ]
        INCLUDE Z80PIO.MAC(11)
Pass - 1        0 Error(s)          24 bytes program.     FTEST5.MAC(20)
        INCLUDE Z80PIO.MAC(11)
Pass - 2        0 Error(s)          24 bytes program.     FTEST5.MAC(20)

        Heep memory [ 411936 ] -- [ 411936 ] bytes

        Heep memory [ 420144 ]
End of ZASM.  time(  0:05 )
```

図３—54　アセンブルの手順

⑤　リンクする。　　ZL ftest5. PRJ↓（図３—55）
主プログラムのリロケータブル・ファイルとライブラリ中のサブルーチンはリンクされ，一つの機械語ファイルになる。HEX ファイル，MAP ファイル，SYM ファイル，XRF ファイルを，図３—56(a)，(b)，(c)，(d)に示す。リンクされたリスト・ファイルを図３—57に示す。

```
B>ZL ftest5.PRJ /A
 ZL.EXE Ver3.09  Linker for Z80 HD64180 8080 8085
                        (c)Micro-org Inc 1989 1990.  updated Mar 01 1991.
        Heep memory [ 490224 ]
Linking. [ FTEST5.BIN ]

Reading [ FTEST5.REL ]
[ FTEST5.LIB ]

Link Pass 1.      2 Global(s). Heep memory 490224 byte(s).
Link Pass 2.      2 Global(s). Heep memory 490224 byte(s).
2 Global Symbol(s)
[ FTEST5.BIN ]          < 0000 > - < 0024 >
Symbol file. Cross reference. writting...
End of link.  time(  0:06 )
```

図3—55　リンクの手順

```
D>TYPE FTEST5.HEX
:200000003E90D3033100873E00D302CD18003EFFD302CD1800C3070016E71E5F1DC21C0056
:0500200015C21A00C921
:00000001FF
```

図3—56(a)　ヘックス・ファイル

```
D>TYPE FTEST5.MAP
Module    --     ASEG      --    CSEG      --     IDSEG       ---  DSEG
FTEST5                           0000-0017
TIMERSUB                         0018-0024
Program size :  <0000>--<0024>  37(  25)bytes.
Out put data :  <0000>-<0024>
                <0000>-<0024>
```

図3—56(b)　マップ・ファイル

```
D>TYPE FTEST5.SYM
0018 TIMER      0000 Z80PIO
```

図3—56(c)　シンボル・ファイル

```
D>TYPE FTEST5.XRF
TIMER                                  [ TIMERSUB ] FTEST5
Z80PIO                                 [ FTEST5 ]
```

図3—56(d)　クロスリファレンス・ファイル

```
D>TYPE FTEST5.LST

 ZA ver 3.09 (c)Micro-org Inc.1990 [1992/ 9/20/12:17] [FTEST5.MAC]page[  1]
    1 0000                        ; ************************************
    2 0000                        ; *   Sample program for Z80-CPU     *
    3 0000                        ; *   File name:   FTEST5.MAC        *
    4 0000                        ; *          ﾗｲﾌﾞﾗﾘﾖｳ ﾂｶｳ             *
    5 0000                        ; ************************************
    6 0000                            INCLUDE  Z80PIO.MAC  ;ｲﾝｸﾙｰﾄﾞ     ;
    1 0000                        ; ************************************
    2 0000                        ; *  File name:    Z80PIO.MAC      *
    3 0000                        ; *     PIOｺﾝﾄﾛｰﾙｼﾞｮｳｽｳ ﾌﾟﾛｸﾞﾗﾑ      *
    4 0000                        ; ....................................
    5 0000                          CSEG                              ;
    6 0000                          GLOBAL    CW,CWR,PORTC            ;
    7 0000  0090                    Z80PIO::  CW      EQU     90H     ;
    8 0000  0003                              CWR     EQU     03H     ;
    9 0000  0002                              PORTC   EQU     02H     ;
   10 0000                        ;************************************
    7 0000                        ;....................................;;
    8 0000                             EXT      TIMER       ;ｶﾞｲﾌﾞﾗﾍﾞﾙ   ;
    9 0000                                                   ;
   10 0000  3E 90                 START:  LD       A,CW          ;
   11 0002  D3 03                         OUT      (CWR),A       ;
   12 0004  31 00 87                      LD       SP,8700H      ;
   13 0007  3E 00                 J1:     LD       A,00H         ;
   14 0009  D3 02                         OUT      (PORTC),A     ;
   15 000B  CD #0000                      CALL     TIMER         ;
   16 000E  3E FF                         LD       A,0FFH        ;
   17 0010  D3 02                         OUT      (PORTC),A     ;
   18 0012  CD #0000                      CALL     TIMER         ;
   19 0015  C3 p0007                      JP       J1            ;
   20 0018                                     END                ;


 Offset  Ext No  Line-No SEG     Name     == LABEL list ==

 FFFFFFFF'                  6             CW
 FFFFFFFF'                  6             CWR
 0007'              13   CSEG    J1
 FFFFFFFF'                  6             PORTC
 0000'              10   CSEG    START
          0              Extern  TIMER
 0000"               6   CSEG    Z80PIO
```

図３―57　リスト・ファイル（ftest5. LST）

ZASM では，ライブラリ化できるファイルは，サブルーチン・ファイルである。

マップ・ファイルとは，プログラムを構成する各ファイル（モジュール）が，どのようなアドレス（アロケーション）でリンクされたかを，確認するために使用する。この例では，ftest5 は，コード・セグメント（CSEG）のアドレス 0000-0017H に，timersub は，0018-0024H に納められ，プログラムサイズは 0000-0024H の 25H であることがわかる。クロスリファレンス・ファイルとは，外部シンボル(ラベル)，サブルーチン，グローバルなデータなどが，どのファイルで宣言されていて，どのファイルから参照されているのかの一覧を示す。形式は，

シンボル名　[定義ファイル名]　参照ファイル，……　である。

シンボル･ファイルは，インサーキュレットエミュレータ (ICE) などで，外部シンボル（ラベル）名などとその先頭アドレスを読み込み，デバック時に利用する。形式は，

アドレス　シンボル　タブ　である。

以上でプログラム開発のアセンブル作業の実例を示した。これに続いて ROM ライターを使用して，ROM にプログラムの書き込み作業，テスト作業，デバック作業などが続く。第 2 報にその実例を示す予定である。

参 考 文 献

1）MIFES Ver 5.0 ユーザーズガイド　メガソフト
2）Z80 系　クロスアセンブラ　ZASM　取扱説明書　マイクロ・オーグ
3）XA80 取扱説明書　システム・ロード
4）Z80　マイコンプログラミング実習　太平洋工業　日刊工業新聞
5）AV-Link ユーザーズマニュアル　アバール
6）ディスアセンブラ　ＺＺ　取扱説明書　マイクロ・オーグ