

第11章 CMT

CMT の基本的な制御は SUB CPU が行っています。Z80 側からは SUB CPU を通して 1Byte をテープに書き込むか、1Byte をテープから読み取る事しか出来ません。Z80 はまず最初に SUB CPU に対して、これから行おうとしている動作がテープの書き込み開始か、テープの読み取り開始なのかを伝える必要があります。伝える方法は、前の章で説明した Z80 から SUB CPU へのデータ送信で、1Byte もしくは連続して 2Byte のデータを送る事によって SUB CPU は CMT 制御モードになります。

コマンドコード	処理内容
3DH,39H	600baud SAVE START
1DH,19H	600baud LOAD START
3EH,39H	1200baud SAVE START
1EH,19H	1200baud LOAD START
3AH	SAVE STOP
1AH	LOAD STOP

CMT 制御のコマンドはテープの読み書きと開始と終了の組み合わせです。例えば 1200baud のテープ読み取り (LOAD) で開始した場合には、テープ読み取り (LOAD) 終了のコマンドを Z80 から SUB CPU に送らなければなりません。

この表からもわかるように、PC-6001 では 600 と 1200 の 2 種類の baud rate を指定する事が出来ます。BASIC の起動後は 1200baud に設定されています。読み書き速度の遅い 600baud の存在意義がわかりませんが、おそらく、先に発売された PC-8001 が 600baud (もしくは 300baud) であり、PC-8001 で保存されたテープを読み込めるようにしてあるのだと思われます。

CMT の書き込みと読み込み処理は SUB CPU に対して行うことになりますが、書き込みと読み込みでは処理の方法が大きく異なります。読み込みは SUB CPU からの割り込みですが、書き込みは Z80 が SUB CPU の端子の状態を常時監視して SUB CPU がデータ受け入れ可能な状態の時に Z80 から SUB CPU にデータを送出するという方式です。

■ テープへの書き込み

テープの管理は SUB CPU が行っているのもので、テープの操作は SUB CPU とのやり取りが主な処理になります。まず最初に、SUB CPU に対してテープ書き込み開始のコマンドを送ります。その後、SUB CPU の状態を監視し、SUB CPU がデータ受け入れ状態かどうかを調べます。この監視処理は BASIC ROM の 1AEDH から記述されているので、そのルーチンを使用するのが確実です。処理内容は次のようになっています。

```

1AEDH:
    LD B,00H

    LD A,08H
    OUT (093H),A

    IN A,(093H)
    BIT 3,A
    JR Z,01AFFH
    IN A,(093H)
    RLCA
    JR NC,01AFFH

; CMT OK 状態
DEC B

1AFFH:
    LD A,09H
    OUT (093H),A
    LD A,B
    OR A
    RET

```

この処理が呼び出された時点で SUB CPU がデータ受け入れ可能かどうかをゼロフラグで返しています。ゼロフラグが 1 の時、SUB CPU は書き込むデータの受け入れが可能な状態となっています。逆にゼロフラグが 0 の時は SUB CPU は受け入れの準備が出来ていないので、再度、このルーチンを繰り返します。もちろん、このルーチン内でチェックを繰り返すような処理に変更しても問題ありません。BASIC ROM では、CSAVE の最中に STOP キーが押されるとセーブ処理を中断する事ができるように作られています。そのため、この SUB CPU チェック処理と STOP キーが押されたかどうかのチェック処理を交互に呼び出しているので、このような実装方法になっています。STOP キーが押されたかどうかは SUB CPU からのキーボード割り込みによって検出されるので、その割り込み先処理で割り込み発生をフラグ管理しています。

SUB CPU が書き込みデータの受け入れ完了状態になったら、まず、SUB CPU に 38H を送ります。その後、続けてテープに書き込む 1Byte のデータを SUB CPU に送ります。以後、この SUB CPU の受け入れチェックと書き込みを繰り返し、書き込むデータが無くなったら、書き込み処理の終了コマンドを SUB CPU に送ると書き込み処理が終了となります。38H の意味は不明ですが、そのようにするルールとなっているようです。

```

LD A,038H
CALL 0E8FH
; 書き出すデータ
LD A,nnH
CALL 0E8FH

```

これら一連の 1Byte 書き込み処理は BASIC ROM の 1ACCH から記述されています。A レジスタにテープに書き込みたいデータを用意して 1ACCH を CALL すると A レジスタの内容がテープに書き込まれます。

テープへの書き込みを行う場合には、事前に書き込み開始のコマンドを送る必要があります、

また、リモート端子を有効にし、テープの安定走行させなければなりません。これら一連のテープへの書き込みを行う前にしなければいけないことは BASIC ROM 内の 1AB8H から記述されています。このように、テープへの書き込みは BASIC ROM に基本的なルーチンが用意されているので、これを利用するのが便利です。また、逆にテープへの書き込み処理を終了させる時の一連の処理は BASIC ROM の 1B06H から記述されています。

BASIC ROM を流用した方法をまとめると、以下の手順となります。

```

; SAVE 処理初期化ルーチン
CALL 1AB8H

LOOP:
; 書き込みデータを A レジスタに準備
LD A,xx
; SUB CPU の状態確認と 1Byte 送出
CALL 1ACCH
; テープへの書き込みを続ける場合には LOOP へ

; SAVE 処理終了ルーチン
CALL 1B06H

```

簡単な使用例ですが、BASIC ROM の内容をテープに書き出すサンプルプログラムです。BASIC ROM 16Kbyte が CMT に書き出されます。

```

LD HL,0000H
LD BC,4000H

CALL 1AB8H

$$loop1:
LD A,(HL)
INC HL
CALL 1ACCH
DEC BC
LD A,B
OR C
JR NZ,$$loop1

CALL 1B06H
RET

```

BASIC ROM 内の処理では、書き込み中に STOP キーが押されたかどうかのチェックも行っているので、このルーチンを利用する場合には CMT アクセス時の STOP キー割り込みについても考慮する必要があります。

また、BASIC ROM では、FA00H からの BASIC ROM ワークエリア (RAM 領域) を使用しているため、この領域まで利用するプログラムを作成する場合には注意が必要です。

ワークエリアの FA1FH が 00H の場合は 600Baud、00H 以外の場合は 1200Baud になります。BASIC ROM では CMT 書き込み処理中に STOP キーが押されると、CMT アクセス時の STOP キーによる割り込み処理によって FA18H が書き換えられます。FA27H には、I/O ポート BOH の値が記録され、テープからの読み込み開始 / 終了時に書き換えられます。

■ テープからの読み込み

テープへの書き込みは SUB CPU の状態監視が主な処理方式でしたが、読み込みの場合は割り込みを使います。SUB CPU はテープから 1Byte のデータを読み取ると、Z80 に対して割り込みを発生させます。その割り込み処理内で SUB CPU から 1Byte を読み取れば、1Byte 分の受信処理が完了します。以後、これを必要な Byte 数分だけ繰り返します。読み取りを終了させる判断は Z80 側で行います。読み取りたい Byte 数をカウントする方法や、読み取ったデータの内容で判断する方法があり、例えば BASIC の CLOAD 命令では 00H が 10 回連続していた場合は、その時点で読み取りを終了するようになっています。市販のソフトでは読み取った Byte 数で判断するものが多いようです。

テープからの読み取りを行うには、まず、SUB CPU にテープ読み取り開始のコマンドを送ります。その後、SUB CPU がテープ読み込みの処理を行い、Z80 に割り込みを発生させます。割り込み処理をしていない間、Z80 は別の処理を実行していても構いません。ただし、SUB CPU はテープ読み込みの処理に専念しているので、SUB CPU に関する他の処理、例えばキーボードの読み込みをする事はできません。また、CMT はその性質上、データの読み込みを一時停止する事ができないので SUB CPU から送られてくるデータを Z80 が随時読み込まないとデータの損失が起きるか、場合によっては SUB CPU がハングアップしてしまうので注意が必要です。このようにテープの読み込みと並行して他の処理を行っている例としては、BOTHTEC 社の EGGY があり、テープの読み込みをしつつ画面の更新を行っています。

テープからの読み込みが必要なくなった時点で Z80 から SUB CPU に対してテープ読み込み終了のコマンドを送ると、SUB CPU はテープの読み込み処理を終了します。ただし、タイミングによっては Z80 からテープ読み込み終了のコマンドを送っても、テープ読み込みの割り込みが発生する事があるので、その場合は SUB CPU から 1Byte の読み込みをしなければなりません。

これらのテープ読み込み開始とテープ読み込み終了に関係する一連の処理は BASIC ROM にルーチンがあるので、それを利用すると処理を簡略化できます。テープ読み取り開始の処理は 1 A61H です。このルーチン呼び出し後から随時、割り込み処理が発生します。割り込み処理で受信したデータを受け取るには、1A70H を呼び出します。このルーチンでは割り込み処理で SUB CPU からデータを受信するまで待機し、データを受信後は A レジスタに受信したデータを格納します。この際に、CMT の読み込みエラーが発生した場合には、ゼロフラグが 1 になっています。CMT からの読み込みを終了するには、1AAAH を呼び出します。

ただし、テープへの書き込み同様、BASIC ROM は、FA00H からの BASIC ROM 用ワークエリア (RAM 領域) を使用しているため、この領域を独自に利用するプログラムを作成する場合には注意が必要です。

ワークエリアの FA1FH が 00H の場合は 600Baud、00H 以外の場合は 1200Baud になり

ます。また、BASIC ROM では CMT 書き込み処理中に STOP キーが押されると、CMT アクセス時の STOP キーによる割り込み処理によって FA18H が書き換えられます。CMT 読み込みの割り込みが発生すると、割り込み処理内で FA19H の内容を書き換え、SUB CPU から送られてきた 1Byte データを FA1DH に待避しています。1A70H からのルーチンでは割り込みが発生するまで FA19H の内容をチェックし続けることで SUB CPU から 1Byte が送られてくるのを監視しています。FA27H には、I/O ポート B0H の値が記録され、テープからの読み込み開始 / 終了時に書き換えられます。

BASIC ROM 内のルーチンも用いた CMT からデータを読み取るプログラム例です。

```
ORG $Df00

; CMT READ OPEN
CALL $1A61

$$LOOP1:
CALL $1A70
CALL DISP_XX
JR $$LOOP1
```

DISP_XX のコードは記載していませんが、A レジスタを 2 桁の 16 進数で表示するサブルーチンです。CMT から 1Byte を受信して、それを画面上に表示するだけのプログラムです。CMT 側で読み込ませるテープはどのようなものでも構いません。テープの内容をひたすら表示していきます。

続いて、割り込みとデータ受信部分のみを独自に実装したプログラム例です。

```
ORG $DF00

; 割り込み設定
di
ld hl,VEC04
ld ($fa08),hl
ei

; CMT READ OPEN
call $1a61

$$loop1:
ld a,(wk00)
or a
jr z,$$loop1
ld a,(wk01)
call DISP_XX
xor a
ld (wk00),a
jr $$loop1

wk00:
db $00
wk01:
db $00
```

```

;-----
; 割り込み処理
;-----

; cmt read
VEC04:
push af
; read:sub cpu

call $0e78
ld (wk01),a

ld a,$ff
ld (wk00),a

pop af
ei
ret

```

割り込み処理では SUB CPU から 1Byte を受信して、受信データの保存と受信フラグを立てています。メインルーチン側では受信フラグを監視し、受信データを表示しています。先の BASIC ROM を利用した方法と処理の結果は同じですが、エラーチェック等々の細かい処理は行っていません。

SUB CPU から送られてくる 1Byte データの内容が正しいかどうかは保証されていません。テープの状態によってはデータが化ける可能性があります。磁気テープ上の記録方式は Byte データが bit 化されたものが単純に並んでいるのではなく、データの区切りとなるマーカーも含めて記録されているので、形式の破損であれば SUB CPU によって検出されますが、データ自体の破損を検出する事は出来ないので、Z80 側で判断する必要があります。市販ソフトではテープからのデータを全て読み終えた時点でデータのチェックサムを行っているものがあります。

SUB CPU がテープからデータを読み取り始める時には、テープ上に記録されている読み取り開始マーカー部分の検出を行っています。このため、テープの途中から再生しても SUB CPU はデータの読み取り処理は働いていますが、不適切なデータとして処理され、Z80 にデータを送りません。これはテープの内容がどのようなものであっても同じで、BASIC から CSAVE されたものであっても、独自形式でテープに書き込みしたものであっても同様です。また、マーカーの作成と検出は SUB CPU によって内部的に行われ、この処理自体を書き換える事は出来ないので、プログラマが意識する必要はありません。

■ CMT の特性

Z80 と SUB CPU とのやり取りによって CMT の制御が出来ますが、実際に CMT を使うには CMT の特性について知っておく必要があります。

PC-6001 本体と CMT は 3 本のケーブルで接続しますが、ケーブルがまったく接続されていなくても PC-6001 はテープ操作の処理を行う事ができます。もちろん、読み込みのケー

ブルがつながれていなければデータの読み込みはできませんが、書き込みのケーブルが未接続でも書き込み処理はできますし、書き込み途中で CMT を止めたりテープが終了してしまっても PC-6001 側、つまり Z80 や SUB CPU はそれを検出する事ができません。

3 本の CMT ケーブルのうち、黒い被服カバーのケーブルはリモート制御信号に用いられ、PC-6001 から CMT に対してテープの読み書きを行う事を伝えるために用いられます。信号は単純に ON/OFF だけであり、この信号が OFF 状態だと CMT の PLAY や REC がハード的に押された状態であってもテープが走行しないように CMT は作られています。PC-6001 がテープの読み書きを行う場合には、リモート制御信号を ON にする必要があり、読み書きが終了したら信号を OFF にしなければなりません。リモート制御は SUB CPU ではなく Z80 が I/O ポートを通してコントロールします。リモート制御信号のケーブルが接続されてなくても SUB CPU は CMT からデータを読み取ったり書き込んだりすることができます。つまり、SUB CPU はその時点でのテープの位置から読み取るか書き込むかを行うだけでテープの読み書き位置は考慮しないで動作していることになります。

カセットテープへのアクセスは、カセットテープの特性を考慮した上で読み書きを行わなければなりません。テープはモータが回り始めてからしばらくは不安定な回転速度になっているので、安定した回転になるまで時間を要します。特にテープへの書き込みを行う際には、テープが安定した走行をした状態から開始しなければなりません。先にも書いた通り、CMT が動作していなくても PC-6001 は CMT にデータを送る事が出来るので、テープが回り始めたばかりの状態で行うとテープ上のデータが不安定なものになってしまいます。書き込み処理はテープが安定動作してから行わなければならないので、そのためにリモート制御信号を使用して、まず最初にリモート端子を ON にしてしばらくしてから（テープが安定走行してから）読み込みや書き込みを行うようにします。

具体的にテープの回転が安定するまで、どの程度の時間を要するのか技術的な資料はないのですが、BASIC の CSAVE では最初に約 3.4 秒だけテープを空回ししているようです。

また、テープは一定速度で回転し続けているので、Z80 側の書き込みや読み取りが遅れてはいけません。テープへの書き込み時に Z80 側の書き込み処理、つまり、Z80 から SUB CPU へのデータ送信が遅れるとテープへの書き込みが間に合わなくなり、テープの読み込み時には Z80 側が SUB CPU からのデータを読み取らずにいと、テープは先に進んでしまうので読み取り失敗（データ抜け）となります。

読み取り失敗の一例を挙げると、プログラムが A パートと B パートに分離していて、A パートが B パートを読み取る（A パートがローダとして機能する）ようなソフトがあるとします。B パートの先頭数 Byte にファイル名が埋め込まれていて A パートはその数 Byte を読み取り、それを画面に表示するとします。この際、その数 Byte をテープから読み取った後に画面にファイル名を表示する処理を実行しますが、この間もテープは走行しているので、表示処理に手間取るとファイルの名に続くデータの読み取りが間に合わず、データの取りこぼしを起こす事になります。

PC-6001 本体とカセットデッキを接続するケーブルは、赤と白と黒の 3 本です。先に説明した様に、このうちの黒はテープのリモート制御端子で、PC-6001 側からカセットデッキへと送られる信号で、デッキの走行を ON/OFF する用途に用いられます。信号は ON と OFF の 2 種類だけで細かな制御信号はできません。リモート制御端子を接続しなくてもテープの書き込みや読み取りはできます。リモート端子の ON/OFF は、I/O ポートの B0h で制御します。

I/O PORT B0h

bit	7	6	5	4	3	2	1	0
I/O					out	out	out	out
内容					リモート	VRAM アドレス		TIMER
					0...OFF	01:E000H		割込み
					1...ON	00:C000H		0...ON
						11:A000H		1...OFF
						10:8000H		

■ CMT 制御中の STOP キー割り込み

SUB CPU が CMT を制御している間、通常のキーボード割り込みは発生しません。ただ、STOP キーだけは例外で、CMT 制御中の STOP による割り込みが発生するようになっています。SUB CPU が CMT 制御中に STOP キーが押されると、読み込みと書き出しそれぞれで異なる割り込みベクタが SUB CPU から発せられます。書き出し中には割り込みベクタ 0EH、読み込み中には割り込みベクタ 10H で BASIC ROM の初期状態では、FA0E-0FH, FA10-11H に 0EA8H が書き込まれていて、書き出しと読み込み両方とも同じ処理で CMT 制御が中断されるようになっています。STOP キーによって CMT 読み込みを中断した場合、その処理の大半は SUB CPU 側で行われるので Z80 側では割り込みの検出を行うだけで構いません。ただし、タイミングによっては SUB CPU から CMT 読み込みのデータが送られてくる事があるので、その場合には受け取らなければなりません。

BASIC ROM のルーチンでは、STOP キーが押された際に PSG 発音処理や STOP キー停止によるフック処理などで BASIC ROM ワークエリアを参照しているので、FA00H 以降のエリアを独自に利用する場合には注意が必要です。また、これ以降の処理が BASIC に戻っています。

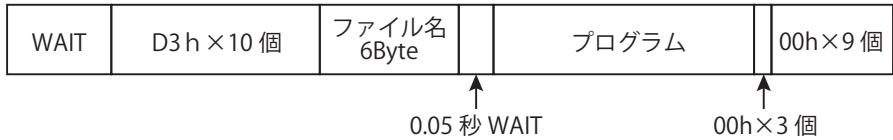
■ テープリードエラー

SUB CPU はテープに記録されてるデータ破損の検出は出来ませんが、テープの記録方式に問題があった場合、SUB CPU は Z80 に対して CMT エラー割り込みを発生させます。BASIC で読み込みや書き込み (CLOAD/CSAVE) の途中でテープを停止させると TR Error が表示されますが、これが記録方式の破損検出です。CMT エラーの割り込みはベクタ 12H で、BASIC 起動後は FA12H に書かれているアドレス 0FB7H にジャンプしています。0FB7H か

らの処理では BASIC ROM のワークエリア FA19H を書き換えた後に BASIC に戻るようになっています。

■ CLOAD 命令について

本書では BASIC の説明はしませんが、PC-6001 でテープからなんらかのプログラムを読み取るには CLOAD 命令を使うしかありません。「第 3 章：BASIC を少しだけ」で、BASIC プログラムの格納形式について説明しましたが、ここではさらにその前段階として BASIC プログラムがどのようにテープから読み込まれるのかを説明します。



先頭の WAIT 部分は無音 + 2400Hz で構成されていて、SUB CPU がテープの読み取り開始検出用に利用するので、Z80 側からは Z80 側からは管理できませんし、する必要もありません。このマーカがないと、SUB CPU はテープからの読み取りを開始しないので、例えばテープの途中から再生した場合、SUB は途中からのデータを CMT から読み取ってはいるのですが、そのデータを Z80 送る（割り込みをかける）事はしません。

続く D3h × 10 個 (Byte) が、これが BASIC のプログラムであると CLOAD 命令に判定させるデータです。CLOAD 命令では、SUB CPU から D3H が 10 個送られてくると、それが BASIC のプログラムであると判断しています。

ファイル名は、CSAVE 命令で付けられたファイル名で最大 6Byte です。CLOAD 命令は 10 個の D3H に続く 6Byte をファイル名とみなして画面上に表示します。ファイル名が 6Bbyte 未満の場合には 00H が格納されています。ファイル名の表示は Z80 側の処理ですが、この処理は画面のスクロールも伴う場合もあり、比較的重い処理になっています。また、ファイル名の表示後には内部的に NEW 命令を実行していることもあり、これらの重い処理の実行中に SUB CPU がファイル名に続けて BASIC プログラムを送ってきしまうと Z80 側でデータを取りこぼしてしまうので、ファイル名と BASIC プログラムの間に WAIT 期間が設けてあります。WAIT はカセットテープ上に存在するものなので、CLOAD で読み込まれるデータではありません。

WAIT 部分について補足すると、WAIT 時間は PC-Technow によると約 0.05 秒となっています。この WAIT は CSAVE 時に Z80 側で生成しています。これは CSAVE 命令が BASIC 開始マーカとファイル名を SUB CPU に送出した後、空ループ処理を実行しているだけで、Z80 と SUB CPU は特別な事はしていません。空ループ処理中も CMT は走行しているので、これが WAIT 期間となるようです。

BASIC プログラムは、「第 3 章：BASIC を少しだけ」で説明した BASIC 先頭アドレスと終了アドレスの間にあるデータがそのまま格納されています。BASIC のプログラムは最後の

行が指す次の行のリンクポインタが 0000 で、その行自体の終端が 00 になっているので、BASIC プログラム部の終端には 3Byte の 00H があることになります。

BASIC 終了マーカとは 9 個の 00H です。先の BASIC プログラム部の終端には 00H が 3 個ありますから、合計 12 個の 00H が並んでいる事になります。実際の CMT 上のデータを調べてみると 12 個以上の 00H が記録されているようなのですが、BASIC の CLOAD では、10 個の 00H を検出するとその時点で読み込みを終了します。例えばなんらかの不具合などによって BASIC プログラムの途中で 10 個連続して 00H が現れるようなことがあると、その時点で CLOAD は " 正しく " 終了します。もちろん、PC-6001 の BASIC は 10 個の 00H が連続して現れないような作りになっているので、そのような事は起こりません。

CLOAD 命令が CMT からの読み込みを終了する条件は 10 個の 00H です。ですので 10 個の連続した 00H がなければ読み込みを続けます。これを利用するとマシン語を含んだ BASIC を読み込む事ができます。CMT と SUB CPU はテープ上のデータが BASIC であるかどうかは判断していません。CLOAD 命令は BASIC の開始マーカを検出するとそれが BASIC のプログラムだと判断しますが、10 個の連続した 00HCMT から読み込まれるまではひたすら SUB CPU から送られてくるデータを RAM 上に書き込んでいくだけです。なので、なんらかの方法で BASIC プログラム部分の後に続けてマシン語部分があって、その後に 10 個の 00H が記録されているような CMT を作成すれば、CLOAD 命令だけでマシン語も読み込ませる事が可能です。

この方法であれば、BASIC マーカとファイル名のあとに BASIC 無しでマシン語部分を記録してもよさそうに思えますが、CLOAD 命令は BASIC 終了マーカの検出後、CMT からの読み込みを終了させてから BASIC プログラムのリンクポインタの確認と修正を実行するため、最低限度の BASIC プログラムが必要になります。

具体的にこのような BASIC+ マシン語構成のテープを作成する方法はいろいろあります。CSAVE 命令では、BASIC プログラムの格納開始アドレスから終了アドレスまでをテープに書き出すので、終了アドレスが格納されているワークエリアを書き換えれば CSAVE 命令によって BASIC プログラム以降のメモリ上のデータもテープに記録出来ます。この BASIC プログラムに続く領域は通常、BASIC の変数エリアとなるので BASIC プログラムを実行すると書き換わってしまいますが、変数を用いない POKE 命令では変化しません。

もしくは BASIC プログラムの POKE 命令と DATA 命令で他の領域にマシン語部分を用意しておき、BASIC を打ち込み直してから、その中間言語等々を含めた BASIC プログラム自身を先のマシン語領域の前にコピーしてから、CSAVE 命令のテープ書き出し開始・終了アドレスを変更して CSAVE 命令を実行するか、CMT 制御のマシン語プログラムを作成して CMT に記録する方法もあります。注意点は、CLOAD 命令は 00H が 10 個並んでいるとその時点で CMT からの読み込みを終了してしまうということです。マシン語データ部に 00H が 10 個並んでいるような箇所があると、その時点で読み込みをが終了してしまいます。

PC-6001 単体では、メモリの少なさもあって、あまり長い内容の BASIC+ マシン語部を作成するのは難しいので、市販ソフトにあるような BASIC+ マシン語部と、それに続くマシン

語部という 2 段構成のテープの方が作りやすいでしょう。この場合、後者のマシン語データ部は CSAVE の書き出し領域の書き換えか、独自に CMT 制御すれば作成できます。

