

マイクロコンピュータを使用したゲートウェイ機能の開発

電子部 永吉弘己

Development of Gateway function using Microcomputer

Hiromi NAGAYOSHI

異機種のコピュータにリモートログインするには、ホストコンピュータで通信プロトコルを変換する必要があり、この処理のためにホストコンピュータの負荷が増大するなどの短所がある。そこで今回、マイクロコンピュータでこの通信プロトコル変換を行うゲートウェイ機能を開発した。漢字コードを含めた通信プロトコル変換用のマイコンボードには、(株)日立製作所製の8ビットシングルチップマイクロコンピュータHD647180XOを使用した。またリモートホストへ接続するための通信手段としては、モデムを介して公衆回線を使用し、通信格としてはRS-232Cによる調歩同期式を使用した。

このゲートウェイ機能の開発によって、センター内のホストコンピュータVAX8350にLANで接続されているパソコンから異機種のコピュータにリモートログインし、仮想端末として使用できることを確認した。

1. はじめに

当センターでは、日本デジタルイクイップメント(株)製のスーパーミニコンVAX 8350をホストとする所内LANを構築しており、EWSとしてVAX station 2000、そして25台のパソコン(日本電気(株)製 PC9801)を接続している。また光ディスクファイルシステム(株)東芝製TOSFile 3200及び550)をモデムで接続している。

LANの規格としてはThin Ethernet(10BASE 2)で接続し、通信プロトコルとしてDECnetを使用している。

今回、このホストコンピュータVAX8350をセンター外の異機種のコピュータに接続し、リモートホストの仮想端末として利用するための、ゲートウェイ機能の開発を行った。

リモートホストへ接続するための通信手段としては、モデムを介して公衆回線を使用し、通信規格としてはRS-232Cを使用した。またリモートログインを実現するために、漢字コードを含めた

通信プロトコル変換用のマイコンボードには日立製作所の8ビットシングルチップマイクロコンピュータHD647180XOを使用した。

2. 通信プロトコルの変換

通信プロトコルとして、調歩同期式は簡易であるためよく利用されているが、異機種のコピュータの接続においては、キャラクタビット長やストップビット長、パリティビットの有無あるいは漢字コードなどが異なっているため、変換を行う必要がある。

一般的には、ホストコンピュータでソフトウェアを作成し、リアルタイムで漢字コード変換を含めた全ての通信プロトコルの変換を行っている。あるいは、ゲートウェイサーバで通信プロトコル変換を行っている。ところが、通信プロトコル変換のためにホストコンピュータに負荷がかかったり、リアルタイムでの通信プロトコル変換には複雑なソフトウェア処理を行わざるを得なくなるなど

の問題がある。

そこで今回開発したのは、図1に示すように、ホストコンピュータで漢字コード変換を行うのではなく、ホストコンピュータとモデムの間に入れたマイコンボードで通信プロトコル変換を行うもので、ホストコンピュータとしては特別な処理を行う必要はない。

VAX8350で使用している漢字コードは、DEC漢字と呼ばれるコード体系で、MS-DOSなどでよく使われているシフトJIS漢字、あるいはメンフレームなどで使用されているJIS漢字とは異なっている。したがって漢字コードの変換をしないで、相手のコンピュータに接続するとASCIIコードは問題ないが、漢字になると文字化けが生じる。

DECコードとJISコードはきまった関係があり、
 $DECコード = JISコード + 8080h$
 という式で表現できる。

例として、「亜」はシフトJISコードでは889Fh、JISコードでは3021hであるが、DECコードではBOA1hとなる。

一方半角カタカナの表示は、シフトJISコードでは基本的に半角カタカナと全角文字の混合ができるようになっている。また、JISコードでは漢字IN (KI) と漢字OUT (KO) を使用して混合を実現している。そしてDECコードでは、エスケープシーケンスを使って混合を実現できるようになっている。具体的には、エスケープコードESC (1Bh) に続けて{(7Ch)で全角文字モード、エスケープコードESC (1Bh) に続けて{(7Dh)で半角カタカナモードになる。

このことから、シフトJIS漢字からDEC漢字へ、あるいはDEC漢字からシフトJIS漢字へ容易に変換できることが予想される。

この様子を図2に示す。

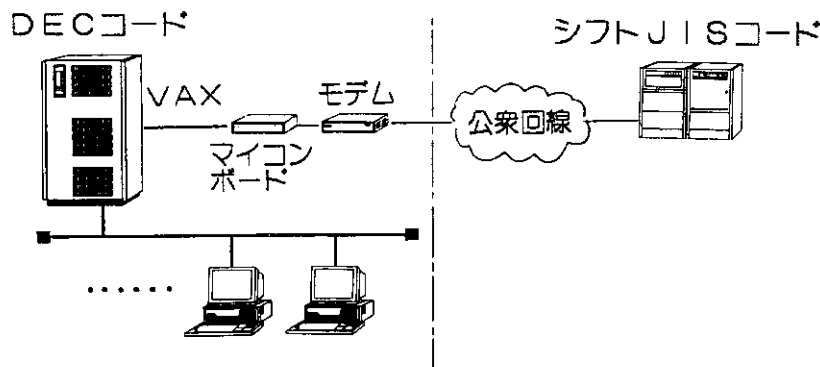
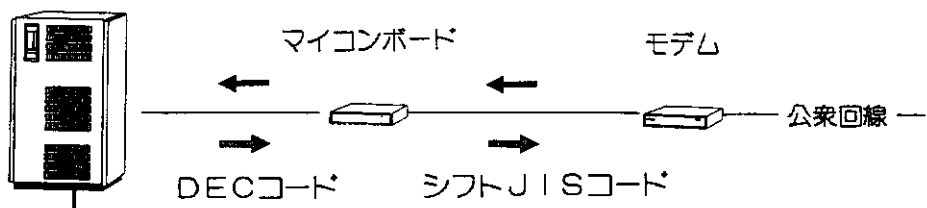


図1 マイコンボードによる通信プロトコル変換



(例)	A	亜	ESC	{	7		A	亜	7	
	41	B0	A1	1B	7D	B1	41	88	9F	B1

図2 漢字コードの変換例

3. マイコンボードのハードウェア

マイコンには日立の8ビットシングルチップマイクロコンピュータHD647180XOを使用した。このHD647180XOには8kバイトのROMと512バイトのRAM、そして2チャンネルのそれぞれ独立した非同期シリアルポート(ASCII)が内蔵されており、それぞれにキャラクタビット長やストップビット長、パリティビットの有無や通信速度の

設定ができる。このため、外付けのハードはRS-232C用のMAX235CPGとリセット回路の74LS14だけですべてである。

図3は今回試作した全回路で、多機能のマイコンを使用したので、周辺回路が少なくなっていることがわかる。

写真1は今回試作したマイコンボードで、写真2は実際にモデムに接続したものである。

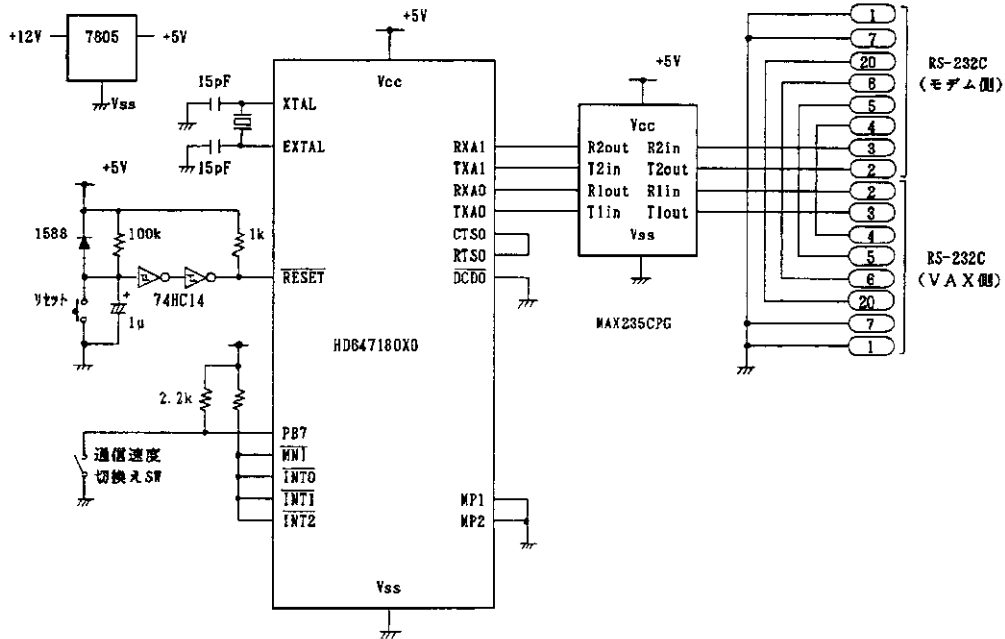


図3 マイコンボードの回路図

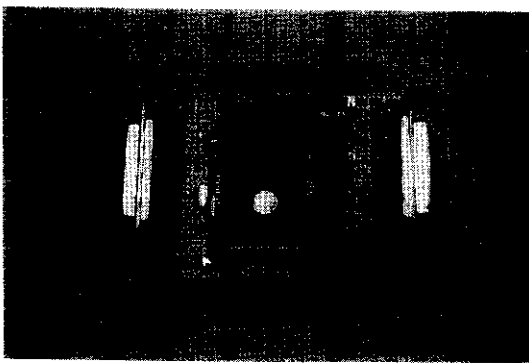


写真1 マイコンボード

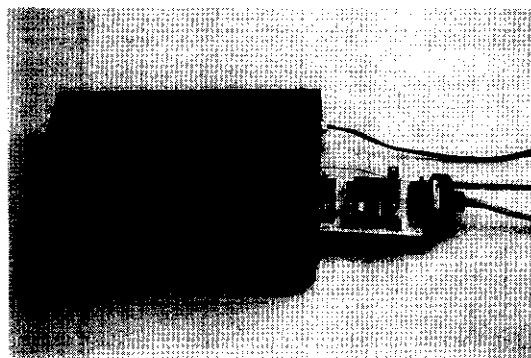


写真2 モデムに接続したマイコンボード

4. マイコンボードのソフトウェア

マイコンボードのプログラムをリスト1に示す。

通信プロトコルは、1スタートビット、8キャラクタビット、1ストップビット、パリティビットなしに設定した。これは、一般的に使用されている調歩同期式である。また、ポートBのビット7を通信速度設定ビットに使用しており、ビット7がHならチャンネル1（モデム側）を1200bps、チャンネル0（VAX側）を2400bpsに、Lならばチャンネル1を2400bps、チャンネル0を4800bpsに設定できるようにした。

チャンネル0の通信速度を2倍にしているのは、半角カタカナと全角文字の混合を可能にするためであり、半角カタカナから全角文字へ（またはその逆）受信文字が変化するとき、漢字コードで説明したように、DECコードではエスケープシーケンスを必要とするために、実際の受信データ（シフトJIS）より2バイト多くのデータをVAX側へ送信するためである。もし、受信と送信を同じ転送速度にしたら、オーバーランエラーが生じてしまう。

この問題を解決するために、通常はRS/CS制御やXフロー制御を行うが、今回はこれらの制御を行わなかった。

5. 動作の確認

試作したマイコンボードを実際に、VAXとモデムのRS-232Cポートに接続し、モデムからシフトJISでの漢字や半角カタカナをマイコンボードに送信した場合、通信速度と漢字コードの変換を実行するかを試験してみた。

通信プロトコルは、8データビット、1ストップビット、パリティビットなしで、通信速度はモデム側（チャンネル1）を1200bpsにセットした。

写真3と写真4はこの様子を示したもので、通信速度を2400bpsに変換していることがわかる。また、モデムからの送信文字（マイコンボードへの受信文字）を、「(SPACE)」(20h)「亜」(889Fh)「亜」(889Fh)とした場合、半角カタカナモードになっていたため、全角文字にするためのエスケープシーケンスが途中に入っている。このため、マイコンボードからの送信（VAXへの入力）は「[SPACE]」(20h)「エスケープシーケンス」(1B7Ch)「亜」(BOA1h)「亜」(BOA1h)となっている様子が見える。これと逆に、全角文字モードになっているときに半角カタカナを受信した場合もやはり、エスケープシーケンス(1B7Dh)を挿入し、全角文字と半角文字の混在を可能にしている。

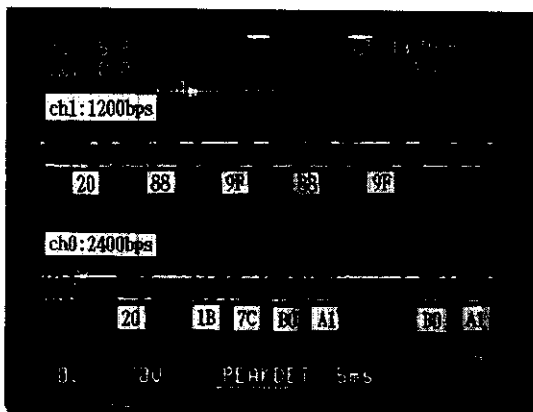


写真3 シフトJIS漢字→DEC漢字の変換

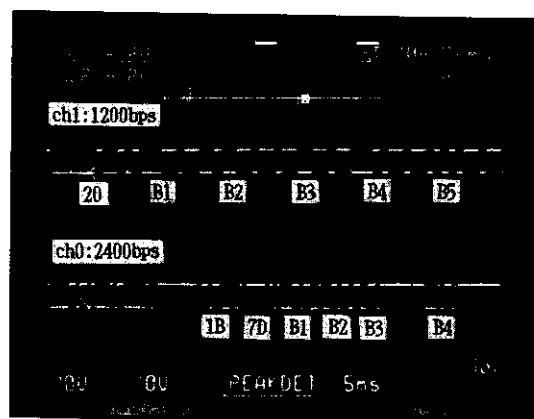


写真4 半角カタカナの変換

6. VAXのプログラム

VAXを電話回線を通じて、他のコンピュータに接続するには次のDCLコマンドを使用した。

```
$ SET HOST/DTE/DIAL=(NUMBER:...,  
MODEM:HAYES)
```

このコマンドを実行すると、VAXのターミナルポートに接続されているヘイズATタイプのモデムに対して、NUMBERで示される番号に自動ダイヤルし、相手のコンピュータシステムに接続でき、相手局の仮想端末となる。

実際にはコマンドプロシージャにしているので、一般ユーザは接続する相手先をメニューで選択できるようにすることができる。コマンドプロシージャはインタプリタであるDCLコマンドをファイルにしたものである。

7. おわりに

今回試作したマイコンボードを使うことによって、VAXにLANで接続されたパソコン端末から異機種のコンピュータに接続し、仮想端末として利用できることを確認した。実用化する場合、次のような問題点が考えられる。

- (1) 通信速度の変更を行う場合、現在のプログラムではマイコンボード上のハードウェアスイッチを使う必要がある。これはプログラムの修正によって、解決することは可能である。ただ最近のモデムは2400bpsが一般的になってきている。
- (2) マイコンボード上でVAX側の通信速度をモデム側の通信速度の2倍の速度に設定してあることで、モデムからVAXへのデータ転送は問題

ないが、VAXからモデムへのデータ転送の場合は、オーバーランエラーが起こることが予想される。実際に日本語フロントエンドプロセッサATOK6で試してみた結果、18バイトを連続送信したときにエラーが起きることを確認した。このことは全角文字で8文字までなら連続文字を送信できることを示している。

したがって、ファイル内容を一括して送信しない場合、仮想端末として使用できると考えられる。今回はVAX側の通信速度をモデム側の2倍にして、エスケープシケンス挿入をできるようにしたためにオーバーランエラーの問題が生じた。この問題については、ハードウェアによるRS/CS制御やソフトウェアによるXフロー制御で解決できる。

この試作によって、各種のコンピュータが採用している通信プロトコルについても、マイコンのプログラムにより対処できる。

また、ホストコンピュータによる複雑な処理を行う必要やゲートウェイサーバを必要としないため、異機種のコンピュータ通信における簡易なゲートウェイ機能として使用できると考えられる。

参考文献

- 1) 日立製作所HD64180XOハードウェアマニュアル
- 2) 渡辺幸保, 他: マイコンコンピュータ, 22, 154 (1986)
- 3) Digital Equipment Corp. DCL dictionary

```

1 *****
2 *          通信プロトコル変換プログラム (SJIS-DEC)          *
3 *          MCU: HD647180XO      PROGRAM: H.Nagayoshi      *
4 *****
5 ;
6 00:0053 DERA EQU 53H ;ポートAのイェーブレジスタ
7 00:0000 CNTL0 EQU 00H ;ASCIIコントロールレジスタA チャンネル0
8 00:0001 CNTL1 EQU 01H ;ASCIIコントロールレジスタA チャンネル1
9 00:0002 CNTL2 EQU 02H ;ASCIIコントロールレジスタB チャンネル0
10 00:0003 CNTL3 EQU 03H ;ASCIIコントロールレジスタB チャンネル1
11 00:0004 STAT0 EQU 04H ;ASCIIステータスレジスタB チャンネル0
12 00:0005 STAT1 EQU 05H ;ASCIIステータスレジスタB チャンネル1
13 00:0006 TDR0 EQU 06H ;ASCIIトランスミッターレジスタ チャンネル0
14 00:0007 TDR1 EQU 07H ;ASCIIトランスミッターレジスタ チャンネル1
15 00:0008 RDRO EQU 08H ;ASCIIレシーバレジスタ チャンネル0
16 00:0009 RDR1 EQU 09H ;ASCIIレシーバレジスタ チャンネル1
17 00:000E IDRB EQU 0EH ;ポートB入力レジスタ
18 00:000F ODRB EQU 0FH ;ポートB出力レジスタ
19 00:0010 DDRB EQU 10H ;ポートBデータレクションレジスタ
20 ;
21 ; 初期設定
22 ;
23 0000 ORG 0000H
24 0000 F3 START: DI ;割り込み禁止
25 0001 31 FF FF LD SP,FFFFH ;スタックインデックス設定
26 0004 1E 00 LD E,00H ;漢字モード(レジスタ0)に設定
27 0006 3E 03 LD A,03H ;ポートAをASCIIとして使用
28 0008 ED 39 53 OUT0 (DERA),A ;DERAに03Hをライト
29 0008 3E 64 INIT0: LD A,64H ;RX/TXイェーブレジスタ、1スタート+8ビット+1ストップ
30 000D ED 39 00 OUT0 (CNTL0),A ;ASCIIコントロールレジスタA(チャンネル0)設定
31 0010 ED 39 01 OUT0 (CNTL1),A ;ASCIIコントロールレジスタA(チャンネル1)設定
32 0013 ED 38 61 INO A,(IDRB) ;ポートBをリット
33 0016 CB 7F BIT 7,A ;ポートBのビット7(通信速度設定スイッチ)をテスト
34 0018 28 0C JR Z,INIT1 ;ポートBのビット7が0ならINIT1へジャンプ
35 001A 3E 04 LD A,04H ;1200bps
36 001C ED 39 03 OUT0 (CNTL2),A ;ASCIIチャンネル0の設定(1200bps)
37 001F 3E 03 LD A,03H ;2400bps
38 0021 ED 39 02 OUT0 (CNTL3),A ;ASCIIチャンネル1の設定(2400bps)
39 0024 18 0A JR MAIN0 ;MAIN0へジャンプ
40 0026 3E 03 INIT1: LD A,03H ;2400bps
41 0028 ED 39 03 OUT0 (CNTL2),A ;ASCIIチャンネル0の設定(2400bps)
42 0028 3E 02 LD A,02H ;4800bps
43 002D ED 39 02 OUT0 (CNTL3),A ;ASCIIチャンネル1の設定(4800bps)
44 ;
45 ; メインプログラム
46 ;
47 0030 CD E4 00 MAIN0: CALL RDRFO ;チャンネル0の受信の有無を確認
48 0033 28 03 JR Z,MAIN1 ;受信が無ければMAIN1へジャンプ
49 0035 CD 42 00 CALL RXOTX1 ;受信があったので処理のためRXOTX1をコール
50 0038 CD EA 00 MAIN1: CALL RDRF1 ;チャンネル1の受信の有無を確認
51 003B 28 F3 JR Z,MAIN0 ;受信が無ければMAIN0へジャンプ
52 003D CD 6F 00 CALL RXITX0 ;受信があったので処理のためRXITX0をコール
53 0040 18 EE JR MAIN0 ;MAIN0へジャンプ
54 ;
55 ; チャンネル0 (VAX側) 受信 / チャンネル1 (モデム側) 送信処理
56 ;
57 0042 ED 00 08 RXOTX1: INO B,(RDRO) ;チャンネル0のRDRO(受信データ)をレジスタへ入力
58 0045 78 LD A,B ;受信データをレジスタへ転送
59 0046 FE 80 CP 80H ;受信データを80Hと比較
60 0048 30 0D JR NC,TX12 ;受信データが80Hより大きければTX12へジャンプ

```

61	004A	FE 7F		CP	7FH		;受信データを7FHと比較
62	004C	20 02		JR	NZ, TX11		;受信データが7FHでなければTX11へジャンプ
63	004E	06 08		LD	B, 08H		;受信データが7FHなら受信データを08Hに変更
64	0050	CD F8 00	TX11:	CALL	TDRE1		;送信バッファの空くのを待つためにTDRE1をコール
65	0053	ED 01 07		OUTO	(TDR1), B		;受信データを送信(+r+r1)
66	0056	C9		RET			;ASCIIコードでの処理終了
67	0057	CD E4 00	TX12:	CALL	RDRFO		;2ndバイトのチェックするためRDRFOをコール
68	005A	28 FB		JR	Z, TX12		;2ndバイトを受信するまで待つ
69	005C	ED 08 08		INO	C, (RDRO)		;2ndバイトをレジスタへ入力
70	005F	CD 30 01		CALL	DEC_SJIS		;DEC→SJIS変換のためDEC_SJISをコール
71	0062	CD F8 00		CALL	TDRE1		;+r+r1の送信バッファが空くのを待つ
72	0065	ED 01 07		OUTO	(TDR1), B		;変換後の1stバイトを送信(+r+r1)
73	0068	CD F8 00		CALL	TDRE1		;+r+r1の送信バッファが空くのを待つ
74	006B	ED 09 07		OUTO	(TDR1), C		;変換後の2ndバイトを送信(+r+r1)
75	006E	C9		RET			;DECコードでの処理終了
76							-----
77							; チャンネル1 (モデム側) 受信 / チャンネル0 (VAX側) 送信処理
78							-----
79	006F	ED 00 09	RX1TX0:	INO	B, (RDR1)		;+r+r1のRDR1(受信データ)をレジスタへ入力
80	0072	78		LD	A, B		;受信データをレジスタに転送
81	0073	FE 80		CP	80H		;受信データを80Hと比較
82	0075	30 07		JR	NC, TX02		;受信データが80Hより大きければTX02へジャンプ
83	0077	CD F0 00	TX01:	CALL	TDRE0		;送信バッファが空くのを待つためにTDRE0をコール
84	007A	ED 01 06		OUTO	(TDRO), B		;受信データを送信(+r+r0)
85	007D	C9		RET			;ASCIIコードでの処理終了
86	007E	FE A0	TX02:	CP	A0H		;受信データをA0Hと比較
87	0080	30 20		JR	NC, TX03		;受信データがA0Hより大きければTX03へジャンプ
88	0082	7B		LD	A, E		;文字モードをチェックする
89	0083	FE 01		CP	01H		;漢字モードかカナモードか?
90	0085	20 03		JR	NZ, TX020		;漢字モードならTX020へジャンプ
91	0087	CD D1 00		CALL	F_KANA		;カナモードなら漢字モードに設定
92	008A	CD EA 00	TX020:	CALL	RDRF1		;+r+r1の受信を確認するためにRDRF1をコール
93	008D	28 FB		JR	Z, TX020		;受信するのを待つためにTX020へジャンプ
94	008F	ED 08 09		INO	C, (RDR1)		;+r+r1のRDR1(受信データ)をレジスタへ入力
95	0092	CD 00 01		CALL	SJIS_DEC		;SJIS→DEC変換のためSJIS_DECをコール
96	0095	CD F0 00		CALL	TDRE0		;送信バッファが空くのを待つためにTDRE0をコール
97	0098	ED 01 06		OUTO	(TDRO), B		;+r+r0から変換後の1stバイトを送信
98	009B	CD F0 00		CALL	TDRE0		;送信バッファが空くのを待つためにTDRE0をコール
99	009E	ED 09 06		OUTO	(TDRO), C		;+r+r0から変換後の2ndバイトを送信
100	00A1	C9		RET			;SJISでの処理を終了
101	00A2	FE E0	TX03:	CP	E0H		;受信データをE0Hと比較
102	00A4	30 0F		JR	NC, TX05		;受信データがE0Hより大きければTX05へジャンプ
103	00A6	7B		LD	A, E		;文字モードをチェックする
104	00A7	FE 00		CP	00H		;漢字モードかカナモードか?
105	00A9	20 03		JR	NZ, TX04		;漢字モードならTX04へジャンプ
106	00AB	CD BE 00		CALL	H_KANA		;カナモードに設定するためにH_KANAをコール
107	00AE	CD F0 00	TX04:	CALL	TDRE0		;送信バッファが空くのを待つためにTDRE0をコール
108	00B1	ED 01 06		OUTO	(TDRO), B		;+r+r0から半角カナ(受信データ)を送信
109	00B4	C9		RET			;半角カナでの処理終了
110	00B5	06 3F	TX05:	LD	B, 3FH		;ASCIIコードでE0H以上のコードは"?"(3FH)
111	00B7	CD F0 00		CALL	TDRE0		;送信バッファが空くのを待つためにTDRE0をコール
112	00BA	ED 01 06		OUTO	(TDRO), B		;+r+r0から"?"(3FH)を送信
113	00BD	C9		RET			;ASCIIコードのE0H以上のコード 処理終了
114							-----
115							; 半角カタカナ受信処理
116							-----
117	00BE	18 1B	H_KANA:	LD	D, 1BH		;エスケープコードESC(1BH)
118	00C0	CD F0 00		CALL	TDRE0		;送信バッファが空くのを待つためにTDRE0をコール
119	00C3	ED 11 06		OUTO	(TDRO), D		;+r+r0からエスケープコードESC(1BH)を送信
120	00C6	16 7D		LD	D, 7DH		;";"

```

121 00C8 CD F0 00      CALL  TDREO      ;送信バッファが空くのを待つためにTDREOに
122 00CB ED 11 06      OUTO  (TDRO),D  ;チャネルから"0"を送信
123 00CE 1E 01         LD     E,01H    ;チャネルに設定
124 00D0 C9           RET                    ;チャネルの設定処理終了
125
126 ; -----
127 ; 全角文字受信処理
128 ; -----
129 00D1 16 1B         F_KANA: LD      D,1BH    ;エスケープコードESC(1BH)
129 00D3 CD F0 00      CALL  TDREO      ;送信バッファが空くのを待つためにTDREOに
130 00D6 ED 11 06      OUTO  (TDRO),D  ;チャネルからエスケープコードESC(1BH)を送信
131 00D9 16 7C         LD     D,7CH    ;"1"
132 00DB CD F0 00      CALL  TDREO      ;送信バッファが空くのを待つためにTDREOに
133 00DE ED 11 06      OUTO  (TDRO),D  ;チャネルから"1"を送信
134 00E1 1E 00         LD     E,00H    ;漢字モードに設定
135 00E3 C9           RET                    ;漢字モードの設定処理終了
136
137 ; -----
138 ; チャンネル0受信有無チェック
139 00E4 ED 38 04      RDRFO: INO     A,(STAT0) ;チャネルのステータスレジスタを入力
140 00E7 CB 7F         BIT     7,A      ;RDRF(レシブデータレジスタフラグ)をチェック
141 00E9 C9           RET                    ;
142
143 ; -----
144 ; チャンネル1受信有無チェック
145 00EA ED 38 05      RDRF1: INO     A,(STAT1) ;チャネルのステータスレジスタを入力
146 00ED CB 7F         BIT     7,A      ;RDRF(レシブデータレジスタフラグ)をチェック
147 00EF C9           RET                    ;
148
149 ; -----
150 ; チャンネル0送信待ち
151 00F0 ED 38 04      TDREO: INO     A,(STAT0) ;チャネルのステータスレジスタを入力
152 00F3 CB 4F         BIT     1,A      ;TDRE(トランスミットデータレジスタエンティ)をチェック
153 00F5 28 F9         JR     Z,TDREO   ;TDREが1(エンティ)になるのを待つ
154 00F7 C9           RET                    ;
155
156 ; -----
157 ; チャンネル1送信待ち
158 00F8 ED 38 05      TDRE1: INO     A,(STAT1) ;チャネルのステータスレジスタを入力
159 00FB CB 4F         BIT     1,A      ;TDRE(トランスミットデータレジスタエンティ)をチェック
160 00FD 28 F9         JR     Z,TDRE1   ;TDREが1(エンティ)になるのを待つ
161 00FF C9           RET                    ;
162
163 ; -----
164 ; シフトJIS→DECコード変換プログラム
165 0100
165 0100 SJIS_DEC:
166 0100 79           LD     A,C      ;受信データの低位バイトをレジスタに転送
167 0101 FE 7F       CP     7FH      ;低位バイトを7FHと比較
168 0103 38 01       JR     C,SD1    ;低位バイト≠7FHならばSD1へジャンプ
169 0105 0D         DEC     C        ;低位バイトを-1する
170 0106 78         SD1: LD     A,B      ;受信データの上位バイトをレジスタに転送
171 0107 FE E0       CP     E0H      ;上位バイトをE0Hと比較
172 0109 38 03       JR     C,SD2    ;上位バイト≠E0HならばSD2へジャンプ
173 010B D6 40       SUB    40H      ;上位バイトを-40Hする
174 010D 47         LD     B,A      ;上位バイト-40Hをレジスタに転送
175 010E 79         SD2: LD     A,C      ;低位バイトor低位バイト-1をレジスタに転送
176 010F D6 1F       SUB    1FH      ;低位バイトor低位バイト-1を-1FHする
177 0111 4F         LD     C,A      ;(低位バイトor低位バイト-1)-1FHをレジスタに転送
178 0112 78         LD     A,B      ;上位バイトor上位バイト-40Hをレジスタに転送
179 0113 D6 60       SUB    60H      ;上位バイトor上位バイト-40Hを-60Hする
180 0115 47         LD     B,A      ;(上位バイトor上位バイト-40H)-60Hをレジスタに転送

```



```

181 0116 D6 21      SUB    21H      ;(上位n'位or上位n'位-40H)-60Hを-21Hする
182 0118 CB 27      SLA    A        ;((上位n'位or上位n'位-40H)-60H-21H)×2
183 011A C8 21      ADD    A,21H    ;((上位n'位or上位n'位-40H)-60H-21H)×2+21H
184 011C 47         LD     B,A      ;((上位n'位or上位n'位-40H)-60H-21H)×2+21Hをレジスタに転送
185 011D 79         LD     A,C      ;(下位n'位or下位n'位-1)-1FHをレジスタに転送
186 011E FE 7F      CP     7FH      ;(下位n'位or下位n'位-1)-1FHを7FHと比較
187 0120 38 05      JR     C,DS3    ;(下位n'位or下位n'位-1)-1FH≠7FHならDS3へジャンプ
188 0122 04         INC    B        ;((上位n'位or上位n'位-40H)-60H-21H)×2+21Hを+1する
189 0123 79         LD     A,C      ;(下位n'位or下位n'位-1)-1FHをレジスタに転送
190 0124 D6 5E      SUB    5EH      ;(下位n'位or下位n'位-1)-1FHを-5EHする
191 0126 4F         LD     C,A      ;(下位n'位or下位n'位-1)-1FH-5EHをレジスタに転送
192 0127 78         DS3: LD     A,B      ;((上位n'位or上位n'位-40H)-60H-21H)×2+21H+1をレジスタに転送
193 0128 C8 80      ADD    A,80H    ;((上位n'位or上位n'位-40H)-60H-21H)×2+21H+1を+80Hする
194 012A 47         LD     B,A      ;((上位n'位or上位n'位-40H)-60H-21H)×2+21H+1+80Hをレジスタに転送
195 012B 79         LD     A,C      ;(下位n'位or下位n'位-1)-1FH-5EHをレジスタに転送
196 012C C6 80      ADD    A,80H    ;(下位n'位or下位n'位-1)-1FH-5EHを+80Hする
197 012E 4F         LD     C,A      ;(下位n'位or下位n'位-1)-1FH-5EH+80Hをレジスタに転送
198 012F C9         RET
199
200 ; -----
201 ; DEC→シフトJISコード変換プログラム
202 ; -----
203 0130      DEC_SJIS:
204 0130 78         LD     A,B      ;受信データの上位n'位をレジスタに転送
205 0131 D6 80      SUB    80H      ;上位n'位を-80Hする
206 0133 47         LD     B,A      ;上位n'位-80Hをレジスタに転送
207 0134 79         LD     A,C      ;受信データの低位n'位をレジスタに転送
208 0135 D6 80      SUB    80H      ;低位n'位を-80Hする
209 0137 4F         LD     C,A      ;低位n'位-80Hをレジスタに転送
210 0138 78         LD     A,B      ;上位n'位-80Hをレジスタに転送
211 0139 57         LD     D,A      ;上位n'位-80Hをレジスタに転送
212 013A CB 47      BIT    0,A      ;上位n'位のビット0をテスト
213 013C 28 01      JR     Z,DS1    ;上位n'位のビット0が0ならDS1へジャンプ
214 013E 3C         INC    A        ;上位n'位-80Hを+1する
215 013F CB 3F      DS1: SRL    A      ;上位n'位-80Hor上位n'位-80H+1を1/2する
216 0141 C6 70      ADD    A,70H    ;上位n'位-80Hor上位n'位-80H+1を+70Hする
217 0143 47         LD     B,A      ;(上位n'位-80Hor上位n'位-80H+1)+70Hをレジスタに転送
218 0144 7A         LD     A,D      ;上位n'位-80Hをレジスタに転送
219 0145 CB 47      BIT    0,A      ;上位n'位-80Hのビット0をテスト
220 0147 20 04      JR     NZ,DS2   ;上位n'位-80Hのビット0が1ならDS2へジャンプ
221 0149 79         LD     A,C      ;下位n'位-80Hをレジスタに転送
222 014A C6 5E      ADD    A,5EH    ;下位n'位-80Hを+5EHする
223 014C 4F         LD     C,A      ;下位n'位-80H+5EHをレジスタに転送
224 014D 79         DS2: LD     A,C      ;下位n'位-80H+5EHをレジスタに転送
225 014E C6 1F      ADD    A,1FH    ;下位n'位-80H+5EHを+1FHする
226 0150 4F         LD     C,A      ;下位n'位-80H+5EH+1FHをレジスタに転送
227 0151 79         LD     A,C      ;下位n'位-80H+5EH+1FHをレジスタに転送
228 0152 FE 7F      CP     7FH      ;下位n'位-80H+5EH+1FHと7FHを比較
229 0154 38 01      JR     C,DS3    ;下位n'位-80H+5EH+1FH≠7FHならDS3へジャンプ
230 0156 0C         INC    C        ;下位n'位-80H+5EH+1FHを+1する
231 0157 78         DS3: LD     A,B      ;(上位n'位-80Hor上位n'位-80H+1)+70Hをレジスタに転送
232 0158 FE A0      CP     A0H      ;(上位n'位-80Hor上位n'位-80H+1)+70HとA0Hを比較
233 015A 38 02      JR     C,DS4    ;(上位n'位-80Hor上位n'位-80H+1)+70HがA0HならDS4へジャンプ
234 015C C6 40      ADD    A,40H    ;(上位n'位-80Hor上位n'位-80H+1)+70Hを+40Hする
235 015E 47         DS4: LD     B,A      ;(上位n'位-80Hor上位n'位-80H+1)+70Hをレジスタに転送
236 015F C9         RET
; ----- END OF LIST -----

```

Lines Assembled : 236

Assembly Errors : 0