

1. 概略

パソコンは大量生産と競争のおかげで価格低下が激しく、ウインドウ・システムなど人間に使いやすいG U I (グラフィカル・ユーザー・インターフェース)を伴って、コスト・パフォーマンスが良くなってきています。

ただ、パソコンのC P Uだけで制御を行なうには、小規模システムではコストの面で、大規模システムでは拡張性やリアルタイム制御性に問題がでてきます。そのため、制御には専用のC P Uを使い、データ設定等のユーザーインターフェースや、収録したデータ処理・解析をパソコンにと、使い分ける方法があります。

データ処理・解析などには表計算等の一般のアプリケーション・ソフトを使うこともできます。ただ、一般のアプリケーション・ソフトのデータ入力、基本的にフル・キーボード入力を前提にしています。独自にパソコン用のソフトを組む場合も、キーボード入力にすると開発・デバッグをしやすいというメリットを得られます。そのため、キーボードと同等の信号を出し、アプリケーション・ソフトを制御すれば、簡単に低価格で高機能なシステムを組み上げることができるようになります。

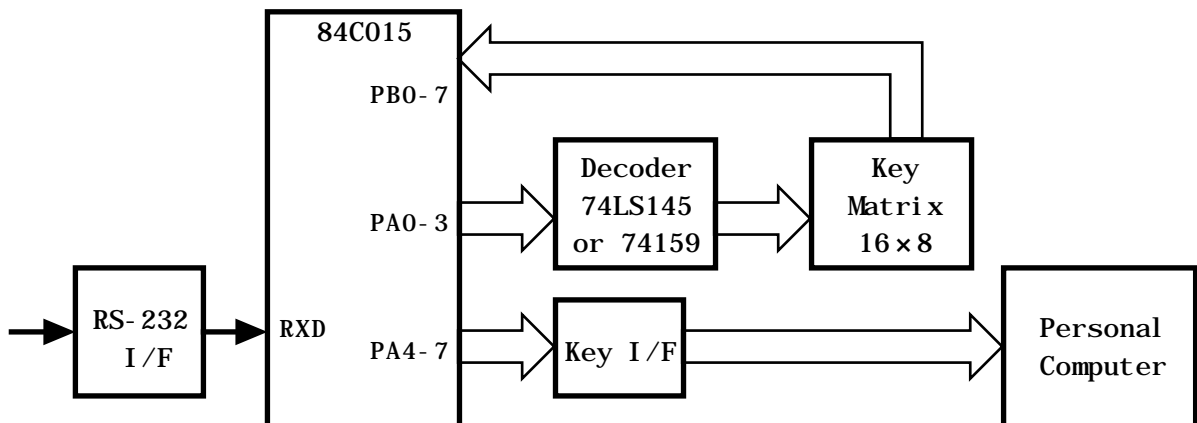
キーボード入力は各メーカーが独自の仕様で、通常ワンチップC P Uを使って制御しています。キー制御プログラムを組んだワンチップC P Uは一般的には手に入れにくく、もし入手しても専用の特種機能の追加は不可能です。

ユーザーが自分で同等の機能を開発しようとする、仕様が完全に公開されていない面もあり、時間や経費がかかり、躊躇することになります。そこで、汎用のC P Uでパソコンのキーボードと同様の信号を出すルーチンを用意しました。

2. 構成

汎用C P Uとして、もっとも一般的なZ80を用意しました。具体的には東芝の84C015 内蔵のI/Oで全て制御します。接続ブロック図を図1に示します。

図1) 接続ブロック図



パソコンは、DOS/V・PC9801・Macintoshの3種類に対応できます。

キーボード数は最高 $16 \times 8 = 128$ 接点とします。キーコードはキー接点番号をROM上のテーブルによってパソコン・キー・コードに置き換えます。そのため、各キーに任意のキーコードを割り当てることができます。パソコン標準キーコードは全てをサポートします。必要ならば標準以外のキーコードを追加することもできます。

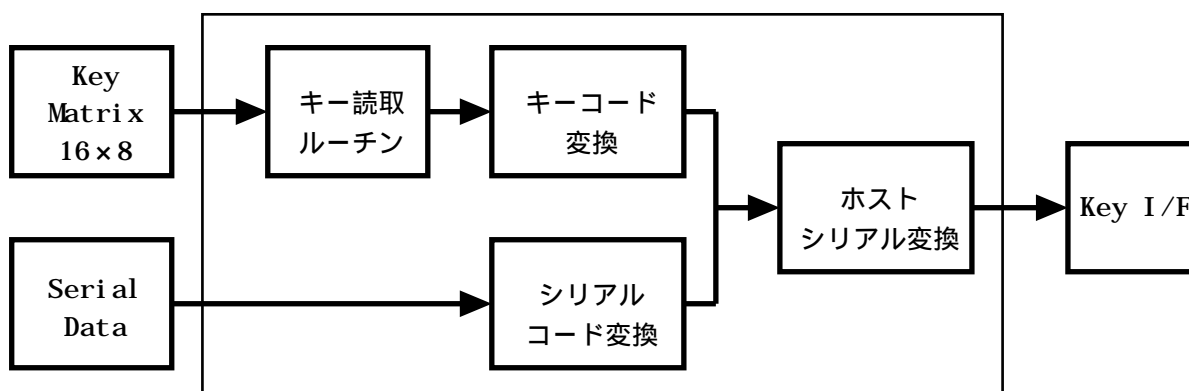
1つのキー接点で複数のキーコードをパソコンに送ることもできます。例えばALT+F4のような組合せキーや、"END"のように一連の文字列のキーコードを、1つのキー接点に割り当てることができます。この機能によって、ワンストロークで一連の命令等をあたえる、専用の特殊機能を持つことができます。

別のシステムからRS-232等のシリアルポートを通して入ってきた8ビットコードを、キーコードに変更する機能も持ちます。この機能によって、外部装置によって収録されたデータの自動入力を行なうことができます。シリアルポートコードをキーコードに変換するテーブルは、キー接点を変換するテーブルとは別に設けています。ここでも、独立に複数キーコードに変換することもできます。

これらのデータの流れを図2に示します。一般的に、ユーザーはデータテーブルを構築するだけで、必要な機能を得ることができます。

制御CPUとしてはZ80を使っているのは、世間で一番一般的であるという理由だけです。他のCPUでも同様の動作を行なわせることも難しくありません。ユーザー要望に従って、ワンチップCPUを含め、各種CPUに対応することもできます。

図2) データの流れ



3. 販売方式

ソースコードのライセンス販売を基本とします。ユーザーは自分でソースプログラムを改変し、自社製品に組み込むことができます。ただし、ソース・プログラムを読んで、他のCPUに置き換えたり、第三者にソースプログラムを譲ったり、開示したりはできないという制限があります。

標準プログラムの改造を、ユーザーに変わり、弊社で行うこともできます。その場合、ユーザーの仕様にしたがって、個別に見積りを行ないます。お問い合わせください。

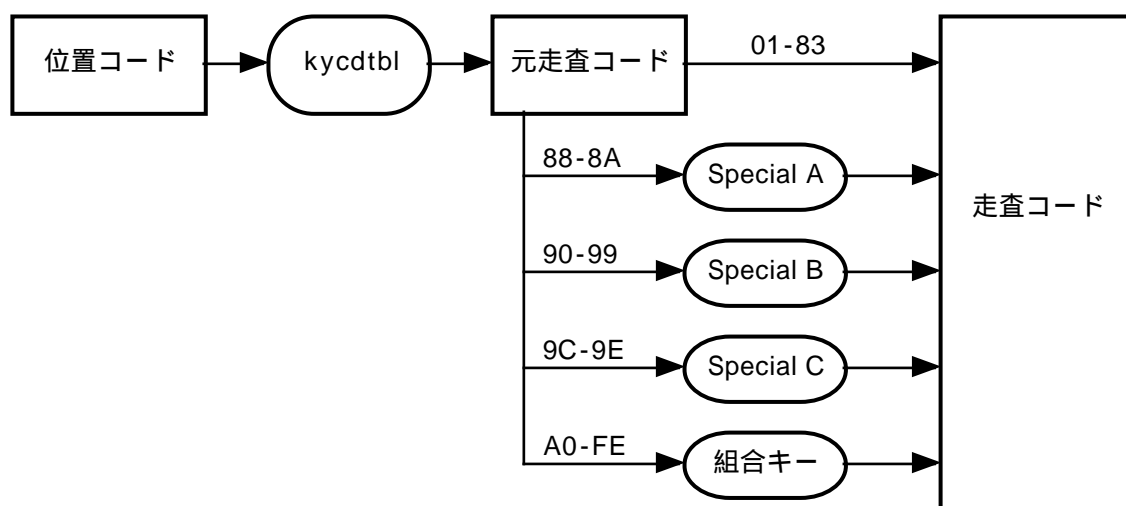
DOS/V対応キーエミュレータ コード変換・解説書

1) キー・コード変換の流れ

1. キーボードの物理的な配置を元にした読みだしコードを「位置コード」と述べます。これは16進数で01-7Fhの最大127個のキー接点に対応します。位置コード"00"は内部ルーチンでキーOFFと同様に認識されるため、使用できません。
2. 「位置コード」はコード変換テーブル "kycdtbl" で「元走査コード」に変換します。元走査コードはホストCPUへ転送する「走査コード」が1バイトメークコードの時は、そのまま「走査コード」になります。
3. 十数個の複数バイトメークの走査コードキーは、追加変換ルーチンで最終的な「走査コード」になります。複数バイトメークキーはそのコードパターンやシフト系のキー (Control, Alt, Shift, Num Lock)によるコード修正の方法によって3種類に分かれます。
4. 元走査コード A0-FE は、オリジナルでは複数のキーが必要な操作を1つのキーで行う組合せキーとなります。組合せキーのメークコードはcpmktblでブレークコードはcpbktblで走査コードを設定します。組合せキーはリピート機能は持ちません。
5. DOS/Vの走査コードは規格ではCode Set1, 2, 3の3種類を規定しています。Code Set1はIBMPC XTの仕様、Code Set3は特殊なコード仕様となっており、DOS/Vでは使うことはまずありません。そのため、本器ではCode Set 2の走査コードのみ送信します。走査コード切り替え命令(Slect Alternate Scan Codes)がホストから来ても、正常に応答はしますが、実際には切り替わりません。また、Code Set3の時にもみ意味のある命令(Set All Key, Set Key Type)にも正常応答しますが、実際に影響は及ぼしません。

以上の変換の流れを下図に示します。

コード変換流れ図



キーコード変換表 (元操作コード順)

走査Code	キー	走査Code	キー	元走査Code	走査Code	キー
0		40		80		
1	F9	41	,	81		
2		42	K	82		
3	F5	43	I	83		F7
4	F3	44	O	84		
5	F1	45	0	85		
6	F2	46	9	86		
7	F12	47		87	[Special A]	
8		48		88	E0 11	R. Alt
9	F10	49	0	89	E0 14	R. Control
A	F8	4A	/	8A	E0 5A	Enter
B	F6	4B	L	8B		
C	F4	4C	;	8C		
D	Tab	4D	P	8D		
E	半/全角	4E	-	8E		
F		4F		8F	[Special B]	
10		50		90	E0 69	End
11	L. Alt	51	(\)	91	E0 6B	
12	L. Shift	52	:	92	E0 6C	Home
13	かな	53		93	E0 70	Insert
14	L. Control	54	@	94	E0 71	Delete
15	Q	55	^	95	E0 72	
16	1	56		96	E0 74	
17		57		97	E0 75	
18		58	CapsLock	98	E0 7A	Page Down
19		59	R. Shift	99	E0 7D	Page Up
1A	Z	5A	Enter	9A		
1B	S	5B	[9B	[Special C]	
1C	A	5C		9C	E0 4A	/
1D	W	5D	(Reserved)	9D	E0 7C	PrintScreen
1E	2	5E]	9E	E1 14 77 E1 F0 14 F0 77	Pause
1F		5F		9F	[Compaund Code]	
20		60		A0		
21	C	61	(Reserved)	A1		
22	X	62		A2		
23	D	63		A3		
24	E	64	変換	A4		
25	4	65		A5		
26	3	66	Back Space	A6		
27		67	無変換	A7		
28		68		A8		
29	Space	69	1	A9		
2A	V	6A	(\)	AA		
2B	F	6B	4	AB		
2C	T	6C	7	AC		
2D	R	6D		AD		
2E	5	6E		AE		
2F		6F		AF		
30		70	0	B0		
31	N	71	(.)	B1		
32	B	72	2	B2		
33	H	73	5	B3		
34	G	74	6	B4		
35	Y	75	8	B5		
36	6	76	Esc	B6		
37		77	NumLock	B7		
38		78	F11	B8		
39		79	+	B9		
3A	M	7A	3	BA		
3B	J	7B	-	BB		
3C	U	7C	*	BC		
3D	7	7D	9	BD		
3E	8	7E	ScrollLock	BE		
3F		7F		BF		

キー位置コード

					7F	77	6F	67	5F	57	4F	47	3F	37	2F	27	1F	17	0F	07			
					7E	76	6E	66	5E	56	4E	46	3E	36	2E	26	1E	16	0E	06			
					7D	75	6D	65	5D	55	4D	45	3D	35	2D	25	1D	15	0D	05			
					7C	74	6C	64	5C	54	4C	44	3C	34	2C	24	1C	14	0C	04			
					7B	73	6B	63	5B	53	4B	43	3B	33	2B	23	1B	13	0B	03			
					7A	72	6A	62	5A	52	4A	42	3A	32	2A	22	1A	12	0A	02			
					79	71	69	61	59	51	49	41	39	31	29	21	19	11	09	01			
					78	70	68	60	58	50	48	40	38	30	28	20	18	10	08	#00			
8	7	6	5	4	3	2	1	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1

1. キー位置コードを16進数で示します。
2. ただし、キー位置コード 00 は使用できません。
3. これを Nkycdtbl でホストへの元走査コードに変換します。

試作品キー配列

		Nみ	Mも	'ね	.る	/め]む	LCtl	かな	-PUp	+	.	0	AF4									
	Cpsl	Hく	Jま	Kの	Lり	;れ	:け	Scll	RAlt	*	Inst	9	6	3	AI半								
	Tab	Yん	Uな	Iに	Oら	Pせ		Pase	Nml	hom	8	5	2	LSft									
	半全	6お	7や	8ゆ	9よ	0わ	Entr	Delt		/	PDn	7	4	1									
	Bこ	spc	-ほ	^へ		BS				Zつ		Xさ	Cそ	Vひ	Entr								
	Gき	@`	[`	F10	F11	F12	RSft	PrS		Aち	End	Sと	Dし	Fは	LSft								
	Tか	F5	F6	F7	F8	F9	¥_	\ろ	LAlt	Qた	変換	Wて	Eい	Rす	MI								
	5え	Esc	F1	F2	F3	F4		RCrl		1ぬ	無変	2ふ	3あ	4う									
8	7	6	5	4	3	2	1	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1

1. キー位置コードを基準にソフトテーブルの kycdtbl でホストへの元走査コードに変換します。
2. 元走査コードから走査コードへの変換は自動的に行なうため、ユーザーは気にすることは有りません。
3. 使用しないキーはFFhに設定してください。
4. ここでは上記配列での変換テーブルを示します。

;7.2 キー入力コード変換テーブル

;

;*****

Key In Code Convert Table *****

;

```

kycdtbl: db      0FFh, 0A2h, 012h, 08Ah, 0FFh, 059h, 0A1h, 0A0h      ; 00-07
           db      025h, 02Dh, 02Bh, 02Ah, 069h, 072h, 07Ah, 070h      ; 08-0F
           db      026h, 024h, 023h, 021h, 06Bh, 073h, 074h, 071h      ; 10-17
           db      01Eh, 01Dh, 01Bh, 022h, 06Ch, 075h, 07Dh, 079h      ; 18-1f
           db      067h, 064h, 090h, 095h, 098h, 092h, 093h, 099h      ; 20-27
           db      016h, 015h, 01Ch, 01Ah, 09Ch, 077h, 07Ch, 07Bh      ; 28-2f
           db      0FFh, 011h, 0FFh, 091h, 097h, 09Eh, 088h, 013h      ; 30-37
           db      089h, 051h, 09Dh, 096h, 094h, 0FFh, 07Eh, 014h      ; 38-3f
           db      0FFh, 06Ah, 012h, 0FFh, 05Ah, 0FFh, 052h, 05Dh      ; 40-47
           db      00Ch, 001h, 007h, 066h, 045h, 04Dh, 04Ch, 04Ah      ; 48-4f
           db      004h, 00Ah, 078h, 0FFh, 046h, 044h, 04Bh, 049h      ; 50-57
           db      006h, 083h, 009h, 055h, 03Eh, 043h, 042h, 041h      ; 58-5f
           db      005h, 00Bh, 05Bh, 04Eh, 03Dh, 03Ch, 03Bh, 03Ah      ; 60-67
           db      076h, 003h, 054h, 029h, 036h, 035h, 033h, 031h      ; 68-6f
           db      02Eh, 02Ch, 034h, 032h, 00Eh, 00Dh, 058h, 0FFh      ; 70-77
           db      0FFh, 0FFh, 0FFh, 0FFh, 0FFh, 0FFh, 0FFh, 0FFh      ; 78-7f

```

;

2) 組合せキーの設定

1. 組合せキーは、"Alt+F4" や "M+l+Enter"のようにオリジナルでは複数のキーが必要な操作を1つのキーに割り当てることができます。組合せキーはリピート機能は持たず、キーONでメイクコード、OFFでブレークコードのみ送信します。
2. メイクコードは cpmkmtbl でブレークコードは cpbkmtbl で走査コードを設定します。
3. キーコードの順番は通常のキー入力と同様に設定します。文字列の時はメイクコードと同じ順序でブレークコードを送ります。シフト系のキー (Control, Alt, Shift, Num Lock)はメイク時は先に、ブレーク時は後に置きます。
4. メイク/ブレークコードの最大長は各々 $cpmkmxnm=2^{cpmkmxex}$, $cpbkmxnm=2^{cpbkmxex}$ で示します。通常ブレークコードの方がプリフィックス (F0h) が追加され、大きくなるため、独立に設定できるようにしました。各々必要なコード以外は FFh で埋めます。

組合せキー設定の例を下に示します。

組合せキーの設定例

```
;  
;*****  Compound Make Code Convert Table *****  
;  
cpmkmxnm equ 8 ; maximum code number  
cpmkmxex equ 3 ; maximum code number (log2) cpmkmxnm  
org ($/256+1)*256  
cpmktbl: db 011h, 00Eh, 0FFh, 0FFh, 0FFh, 0FFh, 0FFh, 0FFh ; A0 Alt+半全角  
          db 011h, 00Ch, 0FFh, 0FFh, 0FFh, 0FFh, 0FFh, 0FFh ; A1 Alt+F4  
          db 03Ah, 043h, 05Ah, 0FFh, 0FFh, 0FFh, 0FFh, 0FFh ; A2 E+l+Enter  
;  
;  
;*****  Compound Break Code Convert Table *****  
;  
cpbkmxnm equ 8 ; maximum code number  
cpbkmxex equ 3 ; maximum code number (log2) cpbkmxnm  
org ($/256+1)*256  
cpbkmtbl: db 0F0h, 00Eh, 0F0h, 011h, 0FFh, 0FFh, 0FFh, 0FFh ; A0 半全角+Alt  
          db 0F0h, 00Ch, 0F0h, 011h, 0FFh, 0FFh, 0FFh, 0FFh ; A1 F4+Alt  
          db 0F0h, 03Ah, 0F0h, 043h, 0F0h, 05Ah, 0FFh, 0FFh ; A2 E+l+Enter  
;
```

3) シリアル入力

1. 別のシステムからシリアルポートを通して入ってきた8ビットコードを、キーコードに変更する機能も持ちます。外部システムによって収録されたデータの自動入力に便利です。
2. シリアルポートコードをキーコードに変換するテーブルは、キー接点を変換するテーブルとは別に srcdtbl に設けました。ここで8ビット256種類のコードを元走査コードに変換します。元走査コードから走査コードへの変換はキーコードと共通となります。
3. シリアルポートから入力した正常データはコード変換した後、キー・メーカーコードを送信し、約10m秒後、ブレークコードを送り、その後シリアルポートに srakcd (06h)を送り返します。シリアルポート側はそれを確認後次のコードを送るようにしてください。
4. シリアル入力でエラーが出た時は srnkcd (15h)を送り返し、キーコードは送信しません。シリアルポート側はシリアルコードを再び送るようにしてください。シリアルポートはハード的なプロトコルは一切使用していません。ACK(06h), NAC(15h) コードのみ確認してください。

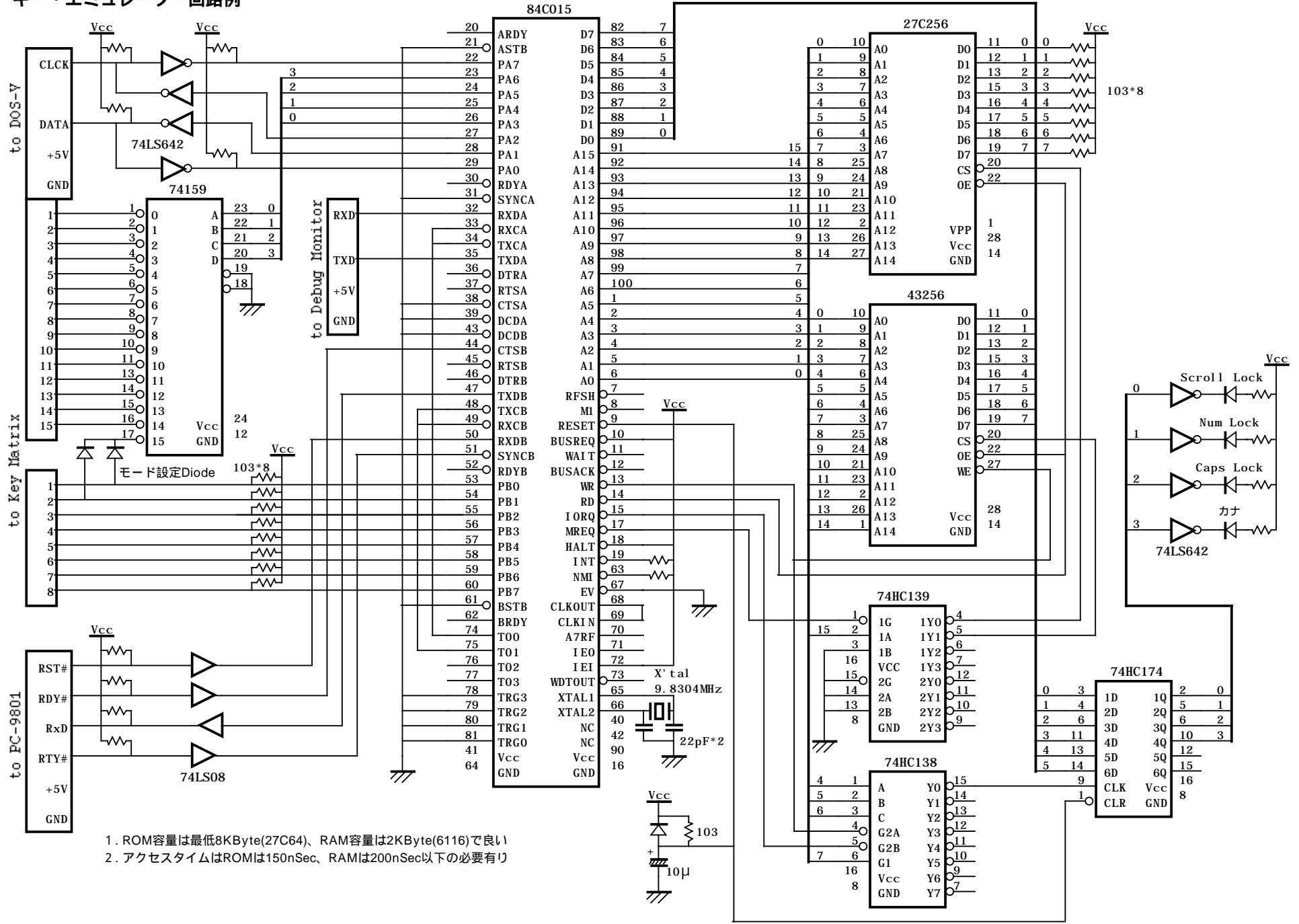
シリアル入力コード変換テーブルの例を下に示します。

英数字記号のASCIIコードをキーコードに変換しています。

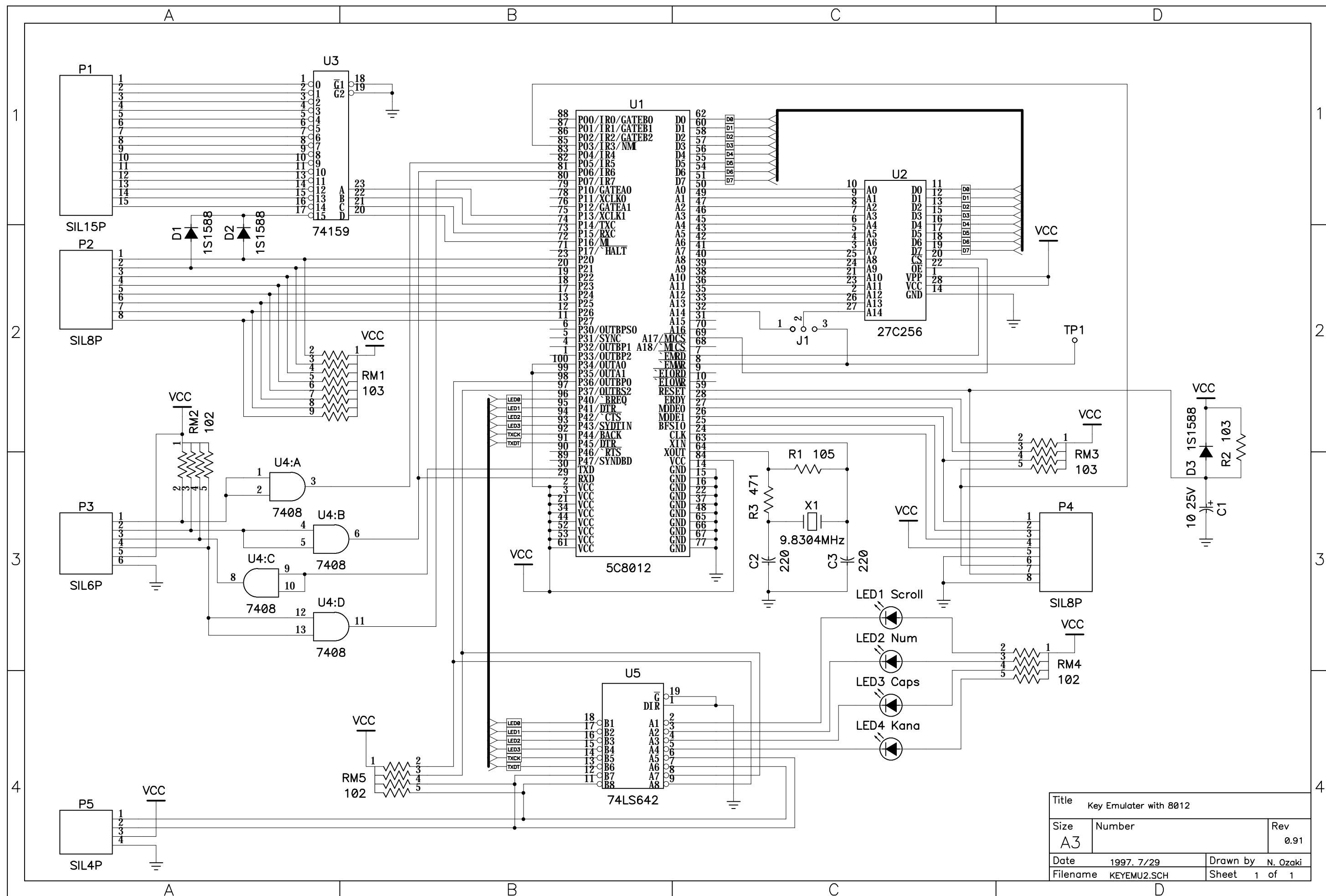
シリアル入力コード変換テーブルの設定例

```
7.1      シリアル入力コード変換テーブル
;
;*****  Serial In Code Convert Table *****
;
      org      table
srcdtbl: db      0ffh, 0ffh, 0ffh, 0ffh, 0ffh, 0ffh, 0ffh, 0ffh      ; 00- 07
          db      066h, 0ffh, 0ffh, 0ffh, 0ffh, 05Ah, 0ffh, 0ffh      ; 08- 0F
          db      0ffh, 0ffh, 0ffh, 0ffh, 0ffh, 0ffh, 0ffh, 0ffh      ; 10- 17
          db      0ffh, 0ffh, 0ffh, 0ffh, 0ffh, 0ffh, 0ffh, 0ffh      ; 18- 1f
          db      0ffh, 016h, 01Eh, 026h, 025h, 02Eh, 036h, 03Dh      ; 20- 27
          db      03Eh, 046h, 07Ch, 079h, 041h, 04Eh, 049h, 04Ah      ; 28- 2f
          db      070h, 069h, 072h, 07Ah, 06Bh, 073h, 074h, 06Ch      ; 30- 37
          db      075h, 07Dh, 052h, 04Ch, 0ffh, 0ffh, 0ffh, 0ffh      ; 38- 3f
          db      054h, 01Ch, 032h, 021h, 023h, 024h, 02Bh, 034h      ; 40- 47
          db      033h, 043h, 03Bh, 042h, 04Bh, 03Ah, 031h, 044h      ; 48- 4f
          db      04Dh, 015h, 02Dh, 01Bh, 02Ch, 03Ch, 02Ah, 01Dh      ; 50- 57
          db      022h, 035h, 01Ah, 05Bh, 06Ah, 05Dh, 055h, 0ffh      ; 58- 5f
          db      054h, 01Ch, 032h, 021h, 023h, 024h, 02Bh, 034h      ; 60- 67
          db      033h, 043h, 03Bh, 042h, 04Bh, 03Ah, 031h, 044h      ; 68- 6f
          db      04Dh, 015h, 02Dh, 01Bh, 02Ch, 03Ch, 02Ah, 01Dh      ; 70- 77
          db      022h, 035h, 01Ah, 05Bh, 06Ah, 05Dh, 055h, 0ffh      ; 78- 7f
          db      0ffh, 0ffh, 0ffh, 0ffh, 0ffh, 0ffh, 0ffh, 0ffh      ; 80- 87
          .
          .
          db      0ffh, 0ffh, 0ffh, 0ffh, 0ffh, 0ffh, 0ffh, 0ffh      ; F0- F7
          db      0ffh, 0ffh, 0ffh, 0ffh, 0ffh, 0ffh, 0ffh, 0ffh      ; F8- Ff
;
```


キー・エミュレーター回路例



1. ROM容量は最低8KByte(27C64)、RAM容量は2KByte(6116)が良い
2. アクセスタイムはROMは150nSec、RAMは200nSec以下の必要有り



Title Key Emulator with 8012		
Size A3	Number	Rev 0.91
Date 1997. 7/29	Drawn by N. Ozaki	
Filename KEYEMU2.SCH	Sheet 1 of 1	