

つくる
シリーズ
6

つくるCRTディスプレイ

綴込付録2色刷原寸パターン図付 | トランジスタ技術
編集部 編



RTTY & マイコン I/O 用—— CRT ディスプレイの製作

新津茂夫 * 藤田悦美
伊藤正晴 * 吉田克信
佐田清彦 * 湯山俊夫

一昔前のアマチュア無線といえば、モールスを使った電信や電話ゴッコを思い浮かべるのですが、最近ではこれらの他に RTTY (Radio Teletype) や、 SSTV (Slow Scan Television) など、あるいは FAX、サテライト通信と、楽しみの範囲にも幅が広げています。とくに RTTY は電信と類似点があるとはいいうもののモールスとは違った機械装置に合った符号を使っています。この信号を受信すると即座にテレタイプ装置の受信紙上にある文字が印字されるもので、その速度は 1 秒間に 6 ~ 10 文字くらいにおよびます*。身近なものにはコンピュータ用の I/O 装置としてのテレタイプや電報などを送受信するテレックスがあります。

当然のことながら送信もタイプライタ式のキーボードを打って送ったり、さん孔した紙テープを利用して送信したりするのです。したがって、他のハムの通信方式とはちょっと異なった趣きがあります。プロの世界でも、この RTTY 通信方式は利用されており、世界中の天気概況や世界各地で起こったニュースなどが、各国の気象庁、通信社から送られています。

しかし、私たちアマチュアからしてみれば、このメカニカル式テレタイプも難点があります。まずマシンが高価であり入手しにくいこと、さらには作動音が意外と大きく、スペース(装置の容積)が大きいため、狭い日本の住宅事情にはそぐわないということです。そこでこれらを電子化しようと考へたのが、これから紹介する CRT ディスプレイ装置なのです。

本来、この種の CRT ディスプレイ装置は、コンピュータ用の簡易表示用として使われているものですが、回路はすべて電子回路ですから、私たちにも容易に手がけることができます。もちろん電子回路ですから、作動音もなく、スペースもそれほどとらないということになります。

製作した CRT ディスプレイは、グループで数台作

る関係上、基板はすべてプリント・パターン化しました。これは記事中にも示しましたが、すでに 5 台ほど完成していますから、再現性についてはかなりの自信を持っています。また、これをマイクロコンピュータ用 I/O 装置に改造した例についても述べたいと思いま

CRT ディスプレイのあらまし

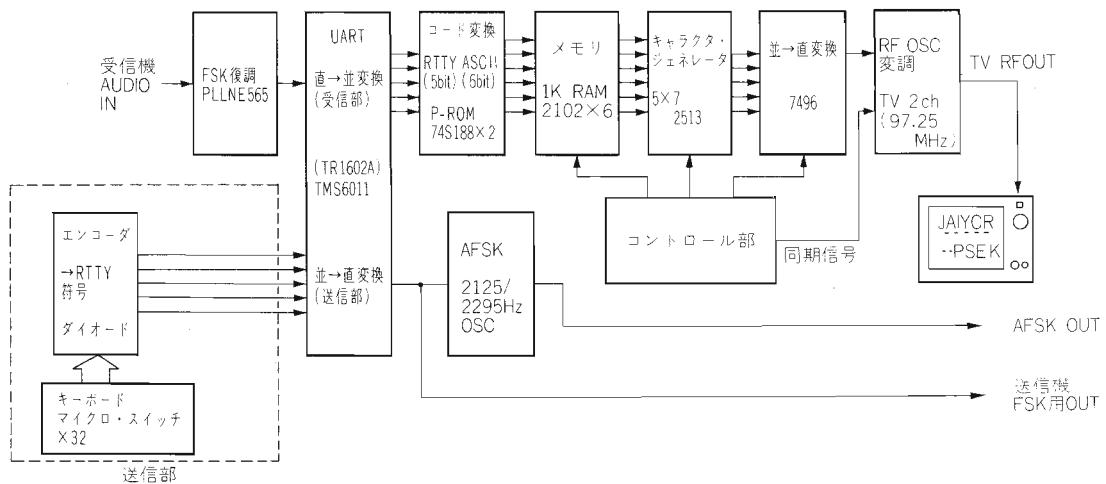
図 1 は私たちが製作した CRT ディスプレイ・システムのブロック・ダイヤグラムです。このシステムは、大別すると受信部と送信部の二つに分けられます。受信部は入力に直列のテレタイプ・コードが入るとそれを並列信号に変換し、TV のブラウン管上に表示する部分、そして送信部はキーボードの文字キーを押すと、それに対応したコードが直列に送出される部分です。

受信部の FSK 復調器や UART (Universal Asynchronous Receiver/Transmitter) による直列-並列信号変換回路の部分は、従来から使われているメカニカルな TTY マシーン、モデム装置の関連技術として参考になると思われます。しかし、このシステムのかなめとなる部分はなんといっても受信部、つまりビデオ・ディスプレイのための回路部(キャラクタ・ディスプレイ部)でしょう。

この部分は並列化されてくる ASCII コードを(入出力は 5 単位の Baudot コードだが、内部で ASCII コードに変換して処理している) メモリにストレージし、それからキャラクタ信号を乗せたビデオ信号を得る部分です。このビデオ信号の出力は、TV のアンテナ端子につなぐだけで画像が得られるよう、TV の空きチ

* 短波回線を利用する関係上、ポー・レイ特は 45.5 ~ 75 ポーと比較的遅い値が使われている。それでも従来の電信による通信よりも 2 倍以上速度が速いという意。

図1 CRTディスプレイのブロック・ダイヤグラム



チャンネル周波数の RF 信号として出力しています。

この CRT ディスプレイ装置は、前述したようにハム用 RTTY 通信に用いることを主目的としたため、通信用の機能、たとえばマーク、スペース表示、アンシフト・オンスペース〔スペース信号を受信したら、必ずレター・ケース(LTR, 下段ケース)へ自動復帰する回路〕、キャラッジ・リターン(CR, 行復帰)、ライン・フィード(LF, 改行)、およびそれらの無視といった付属回路が設けられています。

送信部はキーボードからの信号をダイオード・マトリクスで RTTY コード化し、UART を使って並列-直列交換して送り出しています。

自作の場合、もっとも問題になるキーボードは、市販のケースとマイクロ・スイッチで自作しました。

しかし、先のことを考えるとこれはあまり賢明な方法とはいはず、ジャンクの ASCII コードのキーボードを利用したほうがよかったと考えています。

送信と受信部はコードの直列-並列、並列-直列交換をする UART 部が共通になっており、送信時はキーボードの出力を直接コード変換器に入れ、送り出しの確認をしています。送信機へ送る FSK 用信号（直列になったテレタイプ・コード）は、同時に AFSK 発振回路へ行き、オーディオ・トーンのテレタイプ信号を作っています(AFSK)。これは、紙テープの替りとして、オーディオ・カセットを使うためのテープ・インターフェース、あるいは SSB 送信機へ接続して簡単にアマチュア RTTY を楽しむために設けました。

RTTY の信号

テレタイプに用いられている符号(コード)にはいろいろありますが、ここではアマチュア無線などで使われている標準形ともいいうべき、Baudot(ボード)コード

について説明します。

符号の構成

Baudot コードは、スタート・ストップ式の 5 単位(ビット)コードが使われており、情報が何もないときは常にマーク(電流が流れている、“1”)の状態にあります。1 符号は図2のように文字の種類を表わす五つの等長なパルス(文字コード——後述)と、その前後にある同期用のスタート・パルス、ストップ・パルスから成り立っています。

スタート・パルスは文字コード・パルスと等長ですが、ストップ・パルスの値は $1.41\tau_p$ 、つまり 1.41 倍の長さです。しかし、この値はさほど厳密である必要はありません、ことに長い分にはいくら長くてもかまいません(効率を無視すればの話ですが…). IC でこのような半端な符号を得ることは面倒ですので、一般的には 1.5 倍長、2 倍長(1 倍長;すべての符号が等長)の 2(3)種類のストップ・パルスが使われています。

ここで紹介するシステムは、使用する IC (UART) によって一義的に決まり、2 倍長のストップ・パルスとしています(1 倍長も可能)。

RTTY の文字コードを図3に示しました。この図はさん孔テープを模した形で示しましたが、黒い孔が“H”レベルと対応しています。つまり図2の文字コー

図2 テレタイプの符号("R"のパルス波形とタイミング)

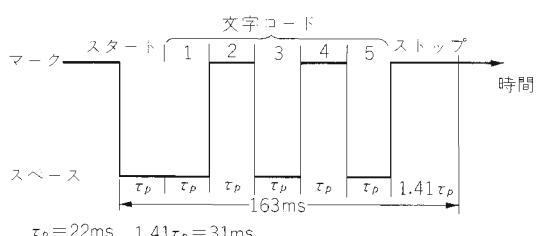
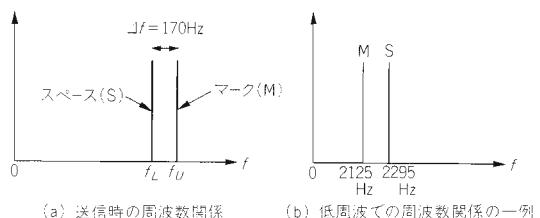


図3 5単位テレタイプ・コード

図4 FSKの周波数関係



ドか、各文字によって図3のように変っていくというわけです。

なお、図3のコードは5単位テレタイプ・コードの中でもポピュラーな American Communications というものです。

通信速度

通信速度の表わし方には二通りの方法があります。ひとつは技術上の、もうひとつは運用上の表わし方です。前者は1秒間に送られる最小パルス幅、つまり図2の $\tau_P(\text{sec})$ の逆数 B (Baud: ボード)で表わすものです。

$$B = \frac{1}{\tau_P} \quad (\text{Baud})$$

短波回線での τ_P は20, 22, 13msのレートが標準的で、アマチュア RTTY では $\tau_P=22\text{ms}$ が用いられています。 $\tau_P=22\text{ms}$ の場合は $B=45.45$ となります(20ms→50B, 13ms→75B)。

後者は1分間に送れる文字数による表わし方 OPM(Operation Per Minutes)と、6文字(5文字+1スペース)を1語として1分間に送れる語数(Word Per Minutes)の二つがあります。図2の例では1文字送るに要する時間が163msですから、

$$\text{OPM} = \frac{60}{0.163} = 368 \text{(文字)}$$

$$\text{WPM} = \frac{368}{6} = 61.3 \text{(語)}$$

となり、ふつう60 WPMと呼んでいます。もちろん、これはストップ・パルス長を規定どおりとして間隔なく、つまり最高速度で符号を送信したときの値ですから、ストップ・パルス長が変ったりすると、若干異なってきます。また、実際のアマチュア RTTY は手打ちで通信を行なうので、もっと遅いレートになりますが符号の時間関係は変わらず、ストップ・パルスが規定よりも長く送られていると考えます。

周波数関係

図2の直列データを短波帯を使って送信するわけですが、これにはFSK(Frequency Shift Keying)*という電波型式を用います。つまりFSK電波は送信データがない(マーク)ときは、それに相当する周波数 f_U の電波が出ていますが[図4(a)]、符号が送られるとスペースのときだけ周波数が f_L まで偏移します。たとえば

f_U が14.1MHzから14.083MHz(f_L)に動くわけです。この $|f_U-f_L|$ の幅 Δf のことをシフト幅と呼び、短波帯ではふつう Δf として170Hz, 425Hz, 850Hzが用いられています。

この信号を短波受信機で復調すると図4(b)のように、 Δf は変わらず低周波信号で得られます。 $f_U > f_L$ のときはロワ・シフト、 $f_U < f_L$ のときはアップ・シフトと呼んでいます。

なお、FS波の占有周波数帯域はシフト幅だけでは決まりません、通信速度が上がるにつれて広くなります。

各部の回路動作

RTTY信号を受信・復調した受信機からのFSK信号は、オーディオ信号ですから、このままではデジタル回路には使用できず、パルスに変換しなければなりません。この部分はPLLを用いて簡単にすませています。PLLによる復調回路は、製作や調整にさほど手間をとらず、安定性や再現性の点ですぐれています。PLLの出力は簡単なCRフィルタを通して受信信号に含まれている不要な短い周期の雑音成分を除いたのち、コンバレータを通して波形整形します。

このようにして得られた直列のRTTYコードは、UARTへと導びかれます。UARTはコンピュータのデータ変換用として開発されたLSIですが、汎用のMSIやSSIで組むと面倒な直列-並列、並列-直列符号変換を、それぞれ独立して行なうことができるICです。取り扱う符号が5ビットのコードなので、UARTの語長を5ビットに選択していますが、このほかに6~8ビットにも選択できるようになっています。

さて、UARTから出力された5ビットの並列RTTYコードは、データ・セレクタを経てP-ROMによるコード変換回路で、入力したRTTYコードに対応したASCIIコードとなります。また、この段階で同時に文字以外の制御信号をも検出します。さきほどのデータ・セレクタは、送信時にはキーボードからの出力をセレクトし、このコード変換部に入力します。

並列のASCIIコードとなった文字信号は、順々にRAMに記憶されていきますが、RAMの容量は1K×6ビットあり、画面2ページ分(1ページは512文字)が記憶できます。そして、通常は半分を使用しており、

パネル面のスイッチによって画面をページ1, 2という具合に切り替えることができます。

さて、RAMに記憶した内容を文字としてTVに表示するためには、TVの走査の動きに合わせて、表示したいときに出力してやらなければなりません。そのため、コントロール部からの信号でアドレスをスキヤンさせ、適当な時期に出力し、キャラクタ・ジェネレータを通して文字表示信号の出力を得ます。なお、キャラクタ・ジェネレータも、コントロール部からの信号によって文字ドットを出力していきます。

このようにして得られたキャラクタ信号と垂直、水平の同期信号を合成して、RF発振器出力を振幅変調し、TVアンテナ端子への出力信号としています。

ブラウン管上の文字の表示は横32字、縦16行で合計512文字が表示できます。また1文字はキャラクタ・ジェネレータの関係で 5×7 のドットで構成されています。

文字信号が入力されると、ブラウン管の左上から順に文字を表示していく、管面の最下行まで一杯になると表示されている文字全体が1行ずつ上にシフトされ(最上行がなくなる)、空いた最下行に新しく文字が表示されていきます。これ以降はこの動作を繰り返し、オペレータは一番下の行だけを見ていればよいということになります。

FSK復調部

受信機やテレコからの低周波FSKの復調はPLLで行ないます(図1左上)。NE565の入力電圧は $100\text{mV}_{\text{rms}}$ もあれば充分なのですが、DCバイアス用の抵抗により入力インピーダンスが低下しているので、マッチングに注意しなければなりません。

PLLの位相検波器の入力周波数とVCOの自走周波数(フリー・ランニング周波数)の差に応じて、出力ピン⑥-⑦間の電圧が正負に変化します。さらにこの電圧をμA741によるコンバレータによって、正負の出力のRTTY信号とします。

コンバレータに続く二つのトランジスタは、レベル・シフタで、基板上のINV, N.INVは、それぞれコンバレータ出力に対する反転、非反転を表わしています。入力周波数とコンバレータ出力、INV, N.INV端子のレベル関係を表1に示します。

入力周波数 $f(\text{Hz})$	$\mu\text{A}741$ (ピン⑥)	INV 端子	N.INV 端子
$2295(\text{S}) > f_0$	+ 5 V	"L"	"H"
$2125(\text{M}) < f_0$	- 5 V	"H"	"L"

表1
FSK復調部の周波数と論理の関係
(f_0 :VCOの自走周波数
[Hz]
 f_0 :2210Hz)

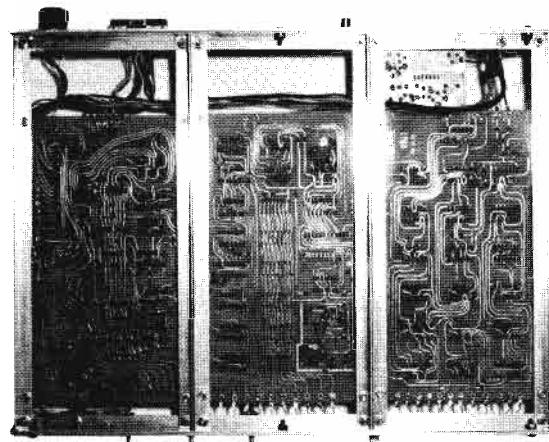
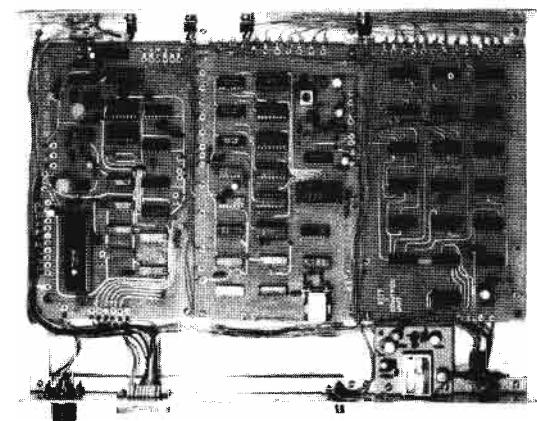


写真1 3枚の基板を平面状に配置した製作例

直列-並列変換部

この部分はUARTが手に引き受けています(図5)。UARTは各社で作られていますが、ここではTIのTMS6011NCを用いました。このほかにもWD(Western Digital)で1602A, GI(General Inst.)でAY5-1013などのピンコンパチブルのものを製造しています。各社各様にピンの名称などは異なるのですが、動作はまったく同じです。ここではTI社のピン名称によって説明します。

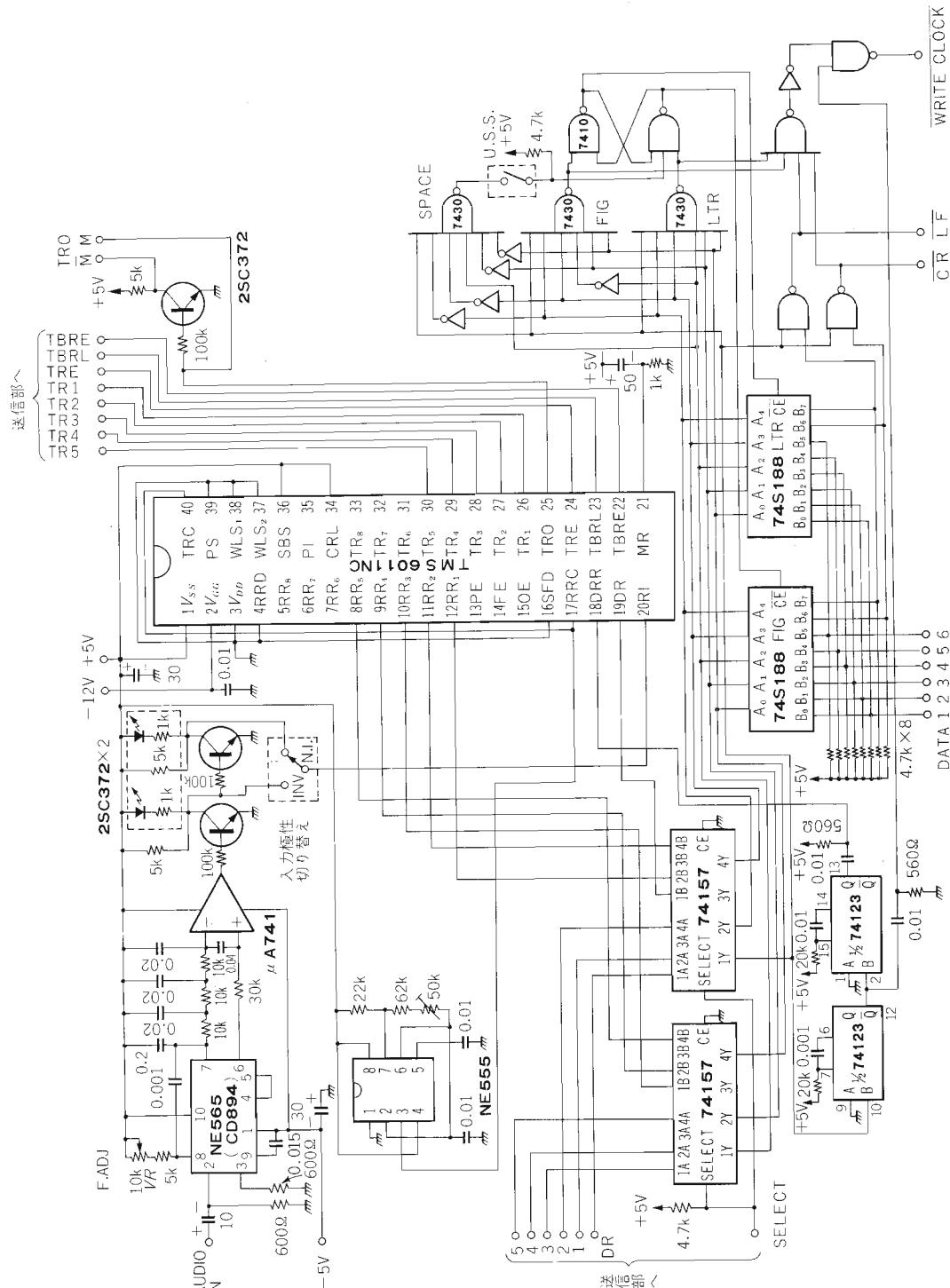
UARTのデータ端子では、マークを"H", スペースを"L"として扱います。また各入力ピンは内部にプルアップ抵抗を持っていますので、開放すれば"H"レベルを与えたことになります。

5ビットの直列テレタイプ符号をRI(Receiver Input)端子に入力すると、IC内部で自動的にスタート・ビットが検出され、RRC(Receiver Register Clock)端子に加えたクロックによって読み込まれていきます。このクロックは入力信号1ビット分の $\frac{1}{16}$ (つまりポートレイトの16倍)の信号を用いることになっています。

45.45 ポーの信号の場合は 727Hz (45.45×16) のクロックが必要です。UART の性能だけで考えるならば、クロックを変えるだけで、この部分で 10k ポー以上の速度まで、符号の直列-並列変換が可能です。さて、UART はクロックのレイトがちょうど 16 倍のときに、

入力された各ビットの中央でサンプリングされるようになっていますので、5ビットの符号の場合はクロック周波数は±7%ほどの範囲の確度であればよいようです。

しかし、復調信号の歪によるミスなどを考えると、



正確であることのほうが好ましいでしょう。

クロックの発振には、安定性が比較的よく、発振周波数も変えやすいタイマ用 IC NE555による回路を用いています。NE555によるクロックの発振周波数は、

$$f_{osc} = \frac{1.46}{(R_1 + 2R_2)C} \quad (\text{Hz})$$

で決定されます。ここで R_1 はピン⑧-⑦間の抵抗、 R_2 はピン⑦-⑥間の抵抗、 C は②-GND間に接続されているコンデンサです。このコンデンサにはマイカ型か、マイラ・フィルム型のものを用いてください。

さて、UART の RI に与えられた直列データが、すべて RR₁(Receiver Holding Register Data)～RR₅まで出力されると(つまり、完全に 1 符号が受信されたら)、DR(Data Ready) 端子が“L”→“H”に変わりま

す。

この信号はデータ・セレクタを通ってワンショット・マルチをトリガし、一定時間後にメモリへのライト・クロックを出力します。さらに一定時間後に負のパルスを出し、DRR(Data Received Reset) 端子に加え、DR 端子を“L”レベルに戻し、次の受信にそなえます。

MR(Master Reset)端子にはすべての動作を始めるまえにパルスを加え、UART 内部をリセットしておくものです。これには C と R を使って、電源投入時に MR 端子が他の部分よりも遅く“L”に落ちるようにしただけですが、確実に動作しています。

このほか UART はコントロール端子(語長、フラグ制御)を備えており、各端子を用途に応じて“H”, “L”の各レベルにセットしておきます。

P-ROMによるコード変換器

メモリに ASCII コードをストアして、キャラクタ・ジェネレータを働かせている関係上、入力のテレタイプ符号(Baudot Code)を ASCII コード変換しなければなりません。

コード変換には P-ROM 74S188(8×32 ビット)を用いています。この P-ROM にはあらかじめ RTTY コードを ASCII コードに従うよう、表 2 のデータを書き込んであります。なお、表 2 中“X”は“0”, “1”どちらでもよいことを表わしていますが、74S188 は“1”を書き込むタイプの P-ROM ですので、何も書き込まない状態すなわち“X=0”と考えてかまいません。

また、*印の符号は ASCII ではありませんが、データを簡略化するためこのようにしました。

P-ROM の A₀～A₄ のアドレス端子に RTTY コードが入力されると、B₀～B₇までの出力に ASCII コードが outputされるというものです。表からもわかるように、RTTY コードは同一コードでアルファベット(LTRS)と数字、記号(FIGS)を示しますが、いずれも表示させるように二通りの P-ROM を用意します。これらのチップは、RTTY 信号の中の LTR, FIG のコードを別途検出し、その信号によって P-ROM の CE 端子を“L”に落とすことによって、選択しています。

さきの図 5 の下側はコード変換部の回路ですが、UART の出力とゲートの間にコードにしたがってインバータを入れ、図の右側の多入力 NAND ゲート 7430 を 3 個用い、SP, FIG, LTR の各符号を検出しています。それぞれの符号が検出されると、7410 による RS フリップフロップをセット、あるいはリセットします。

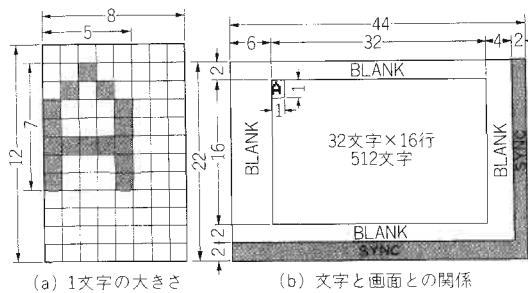
スペース符号の検出は、さきに述べたアンシフト・オンスペース回路のためです。この NAND の出力をスイッチで切り離すことにより、この機能を停止させることになります。

さて P-ROM の出力の B₆, B₇ は、それぞれキャラ

表 2 RTTY(Baudot)-ASCII コード変換表(74S188)

RTTY 74S188 のアドレス A ₄ ～A ₀	LTRS	FIGS	P-ROM1 ASCII LTRS	P-ROM2 ASCII FIGS
0 0 0 0 0	UNP TAPE		B ₇ B ₆ B ₅ B ₄ B ₃ B ₂ B ₁ B ₀ 1 0 0 0 0	B ₇ B ₆ B ₅ B ₄ B ₃ B ₂ B ₁ B ₀ 1 0 0 0 0
1 0 0 0 1	T 5		0 1 0 1 0	1 1 0 1 0 1
2 0 0 0 1 0	CARR RET		*0 1 x x x x x x	*0 1 x x x x x x
3 0 0 0 1 1	O 9		0 0 1 1 1 1	1 1 1 0 0 1
4 0 0 1 0 0	SPACE		1 0 0 0 0 0	1 0 0 0 0 0
5 0 0 1 0 1	H #		0 0 1 0 0 0	1 0 0 0 1 1
6 0 0 1 1 0	N ,		0 0 1 1 1 0	1 0 1 1 0 0
7 0 0 1 1 1	M .		0 0 1 1 0 1	1 0 1 1 1 0
8 0 1 0 0 0	LINE FEED		*1 0 x x x x x x	*1 0 x x x x x x
9 0 1 0 0 1	L)		0 0 1 1 0 0	1 0 1 0 0 1
10 0 1 0 1 0	R 4		0 1 0 0 1 0	1 1 0 1 0 0
11 0 1 0 1 1	G &		0 0 0 1 1 1	1 0 0 1 1 0
12 0 1 1 0 0	I 8		0 0 1 0 0 1	1 1 1 0 0 0
13 0 1 1 0 1	P \$		0 1 0 0 0 0	1 1 0 0 0 0
14 0 1 1 1 0	C :		0 0 0 0 1 1	1 1 1 0 1 0
15 0 1 1 1 1	V :		0 1 0 1 1 0	1 1 1 0 1 1
16 1 0 0 0 0	E 3		0 0 0 1 0 1	1 1 0 0 1 1
17 1 0 0 0 1	Z "		0 1 1 0 1 0	1 0 0 0 1 0
18 1 0 0 1 0	D \$		0 0 0 1 0 0	1 0 0 1 0 0
19 1 0 0 1 1	B ?		0 0 0 0 1 0	1 1 1 1 1 1
20 1 0 1 0 0	S		0 1 0 0 1 1	
21 1 0 1 0 1	Y 6		0 1 1 0 0 1	1 1 0 1 1 0
22 1 0 1 1 0	F /		0 0 0 1 1 0	1 0 0 0 0 1
23 1 0 1 1 1	X /		0 1 1 0 0 0	1 0 1 1 1 1
24 1 1 0 0 0	A -		0 0 0 0 0 1	1 0 1 1 0 1
25 1 1 0 0 1	W 2		0 1 0 1 1 1	1 1 0 0 1 0
26 1 1 0 1 0	J ▼		0 0 1 0 1 0	1 0 0 1 1 1
27 1 1 0 1 1	FIGURES		x x x x x x	x x x x x x
28 1 1 1 0 0	U 7		0 1 0 1 0 1	1 1 0 1 1 1
29 1 1 1 0 1	Q 1		0 1 0 0 0 1	1 1 0 0 0 1
30 1 1 1 1 0	K (0 0 1 0 1 1	1 0 1 0 0 0
31 1 1 1 1 1	LETTERS		x x x x x x	x x x x x x

図6 画面の構成



(a) 1文字の大きさ

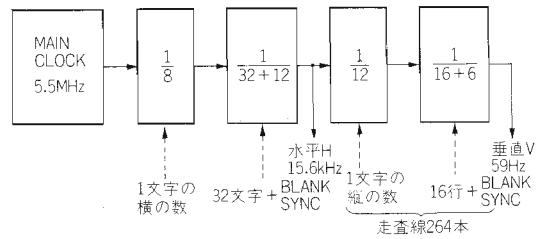
(b) 文字と画面との関係

タッジ・リターン(復帰), ライン・フィード(改行)のとき, "H"を出力します。この出力をUARTのDRパルスと同期させ, コントロール部に導きます。なお, SP, FIG, LTR の各信号も, DR パルスと AND をとって検出されます。

ところで, P-ROMの出力をメモリに書き込ませるには, 書き込ませたい瞬間にメモリ部へライト・クロック用のパルスを与えます。このパルスは, UARTに出力データが並んでいる間に出来ねばなりません。そのため, 74123によるワンショット・マルチのパルスは, DR端子が "H"になってわずかな時間後に, ライト・クロック・パルスを出力します。

しかし, 入力された信号がSP, FIG, LTR, CR, LFなどの命令信号である場合は, ライト・クロックが出ては困ります。そのため4入力NANDゲート7420で各命令信号のORをとって, ライト・クロックを出力するゲートを閉じています。さきの74123は, DRパ

図7 周波数関係



ルスから OR 出力までの遅れを補つてることになるわけです。

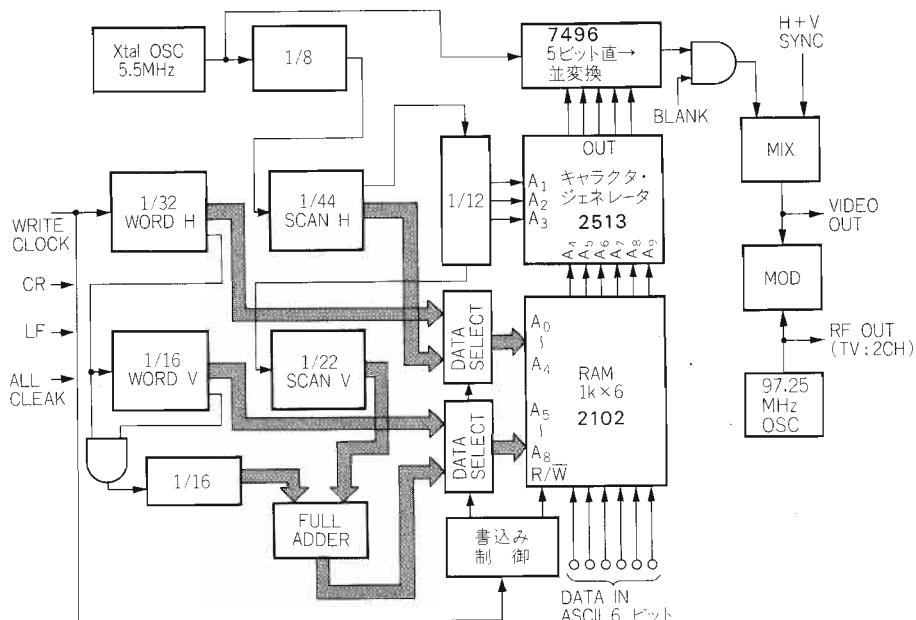
その後, さらにワンショット・マルチを通して負のパルスを DRR に加え, DR を "L"にして次の入力信号が入ってくるのに備えます。

キャラクタ・ディスプレイ部

この部分が本稿の主たるところで, ここをどう設計するかで回路の機能や複雑さが変わってきます。

●読み出し動作 画面上に何文字表示させるかという問題があります。TVの解像度, 字のつりあい(大きさなどの), MOS ICの遅れ時間などを考え合わせて, 32文字×16行の 512 文字を 1 画面に表示しました。そして文字は 5 × 7 のドット・マトリクスによる表示としたため, 最終的に文字と画面の関係は図6のようになりました。また水平, 垂直の周波数はだいたい決まっていますから, 原発の水晶の発振周波数は 5.5MHz 前後になります。そして回路を簡単にするため, 飛び越し走査は行なっていませんから, 走査線は 264 本となります(図7)。

図8
キャラクタ・ディスプレイ部のブロック・ダイヤグラム



ブランク部を上下、左右に図6のように多くとっても、実際にTVの画面に表わされるのは文字の部分だけだ、ブランクと同期の部分は画面の外に出て何も見えません。つまり、有効画面は非常に少ないわけです。絵素ならば周囲が少々はみ出したとしても、さほど気にはならないのですが、CRTディスプレイの場合は文字が見えなくなるわけですから、大きな問題となります。

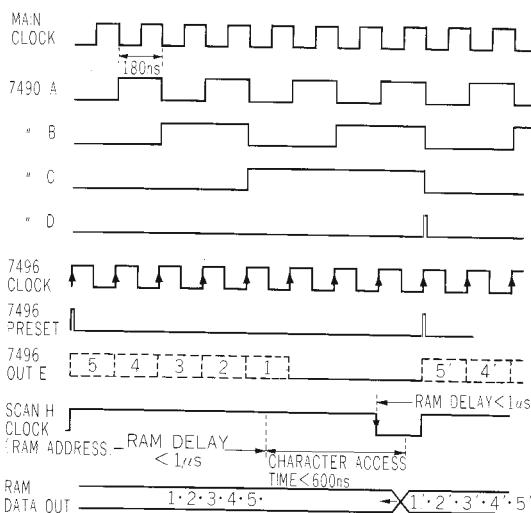
さて図8はキャラクタ・ディスプレイ部のブロック・ダイヤグラムを示したものですが、入力のコマンドとしてはライト・クロック、CR、LF、およびマニュアルによるオール・クリアの四つだけです。したがって、ワード・カウンタは通常のバイナリ・カウンタを用い、CRとLFは途中でゲートを用いて挿入しています。

このWORDカウンタが、今書いている（表示している）文字の場所を記憶していることになります。一方、これと並列にSCANカウンタがありますが、このカウンタはRAMの内容の読み出し専用のカウンタで、図7のカウンタの一部として、同期やブランク信号を作っています。そして、これら二つのカウンタをデータ・セレクタで切り替えて、RAMのアドレスとします。

RAMのデータ入力にはP-ROMによって変換されてきた6ビットのASCIIコードが入ります。ここではRAMにIntelの2102を6個使用していますから、1K語（文字）を記憶することができます。従来のCRTディスプレイのメモリには、シフトレジスタがおもに使われていました。この場合には、メイン・メモリのほかにライン（行）・メモリも必要でした。

というのは1行32文字を完全に表示させるには、32

図9 キャラクタ部のタイミング・チャート



文字のメモリを12回スキャンさせ、そして次の行の32文字を…という動作をさせるためです。ところが、RAMの場合は単純にアドレスを繰り返しスキャンさせるだけなので、この部分の回路が簡単になるのです。

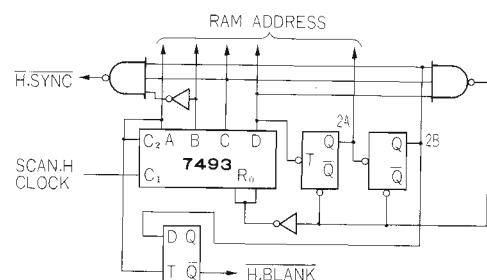
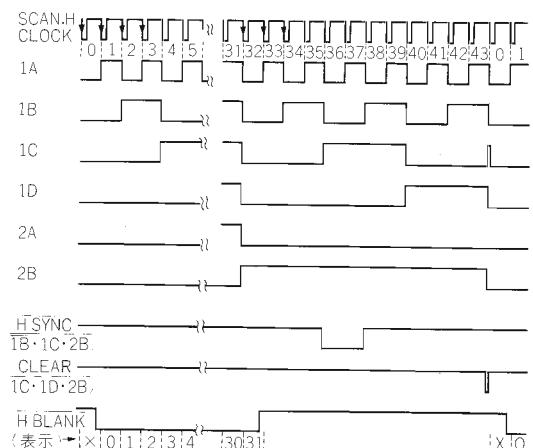
RAMの出力はキャラクタ・ジェネレータ2513のA₄～A₉に入ります。これによって選択された文字が5本、並列に出てきますから、これを7496によって並列-直列変換して、シリアル・データとして取り出します。簡単に説明すると以上のとおりなのですが、ここに大きな問題があるのです。

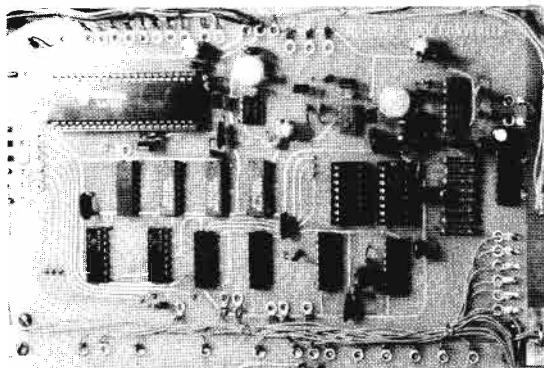
それはMOS ICによる、データの遅れです。TTL同士の場合は数10nsの遅れ時間だけなので、ほとんど無視することができるのですが、MOS ICの場合はμsオーダにまで食い込むことがあります。

この回路の場合は、RAMのアドレスを変えてから、キャラクタ・ジェネレータの出力にデータが現われるまでの時間は、内部にラッチ機能を持っていないため加算され、最大1.6μsとなってしまいます。画面の1字分が1.4μsですので、これは無視することができません。そこで、ある程度の遅れ時間を見込んで、少し早めにアドレッシングをしています。

ただ、この1.6μsは最大遅れ時間であり、実際はMOS ICもこれよりかなり速い動作をしてくれます。以上の動作および水平、垂直のタイム・チャードを図9、

図10 水平スキャンのタイミング・チャート





〈写真2〉受信部、コード変換部、左上がUART

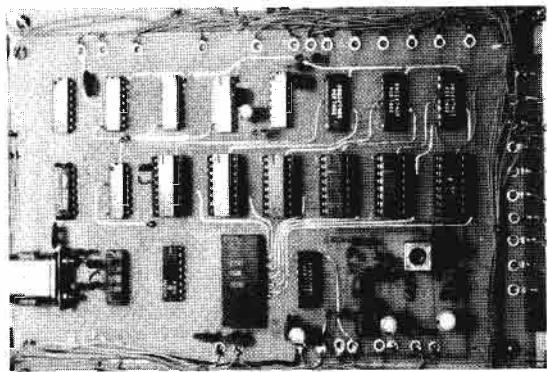
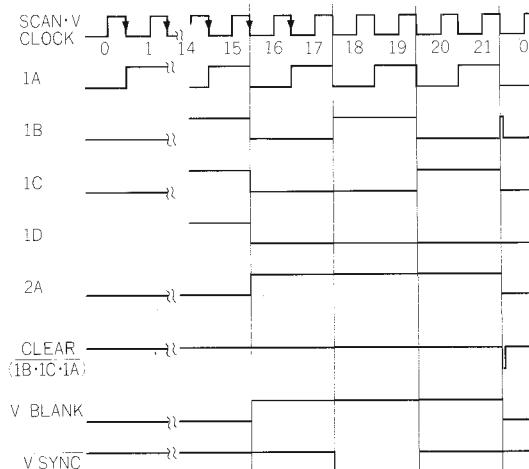
図10、および図11に示します。

●書込みの動作 以上が読み出しの動作ですが、書き込みの動作は次のようにになります。ライト・クロックによって単安定マルチを働かせ、一定時間RAMのアドレスをデータ・セレクタにより、スキャン・カウンタからワード・カウンタに固定します。そしてRAMのR/W端子を“L”にして入力のデータを書き込みますが、このときC-MOSゲートの遅れを使って、両者のタイミングをずらさせています。

この書き込み方法は、少し強引なところがあります。それはたとえば、単安定マルチの時定数を1秒にした場合には、この時間の間入力データの字が画面全部に出てしまうということです。しかし、実際にはこの時間を数μs程度に選びますので、視覚上はまったくわからなくなっています。

オール・クリアをするのに、RAMの内容をすべて“0”にしてしまうと画面上には@マークが表示されますので、キャラクタ・ジェネレータのブランクのデータ("100000")をスキャン状態で書かせます。以上で一

〈図11〉垂直スキャンのタイミング・チャート



〈写真3〉キャラクタ・ディスプレイ部、右下がRF発振部

応この部分は動作するのですが、このままでは512文字書き終わったあと、上からまた前の文字を書き替えながら…というように非常に見にくい動作をします。

そこで、一番下まで書き終えたなら、全体を1行上へずらして、いつも一番下の行に書かせるスクローリング回路をつけ、画面が下から上へ流れしていく感じにしました。これを実現させるためには、アドレスのSCAN-CANカウンタをフローティングさせるような形で使います。

図12をごらんください。WORD-Vカウンタが、オール・クリアされてから、16個目のクロックでフルアダマーの入力のカウンタがカウント1されます。するとSCAN-Vカウンタに1加えられたものがRAMの行アドレスとなるので、表示文字全体が1行上にズれます。しかし、16行目には前の1行目が表示されますので、この行だけをクリアしなければなりません。

これは前述のオール・クリアのときと同じようにしますが、1行が走査線12本、つまり12回同じアドレスをSCANしますから、クリアする時間を100μs前後に選べば充分となります。以上キャラクタ・メモリ、コントロール回路を図13、および図14に示します。

〈図12〉スクローリング回路

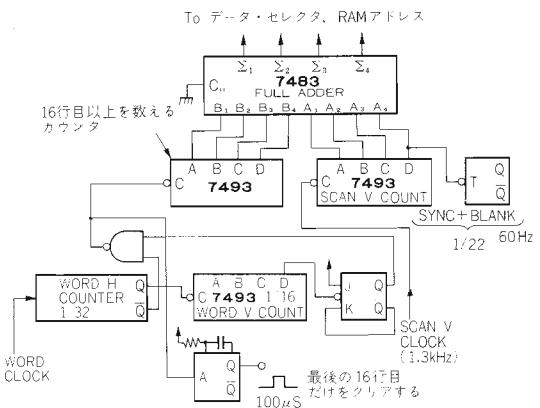
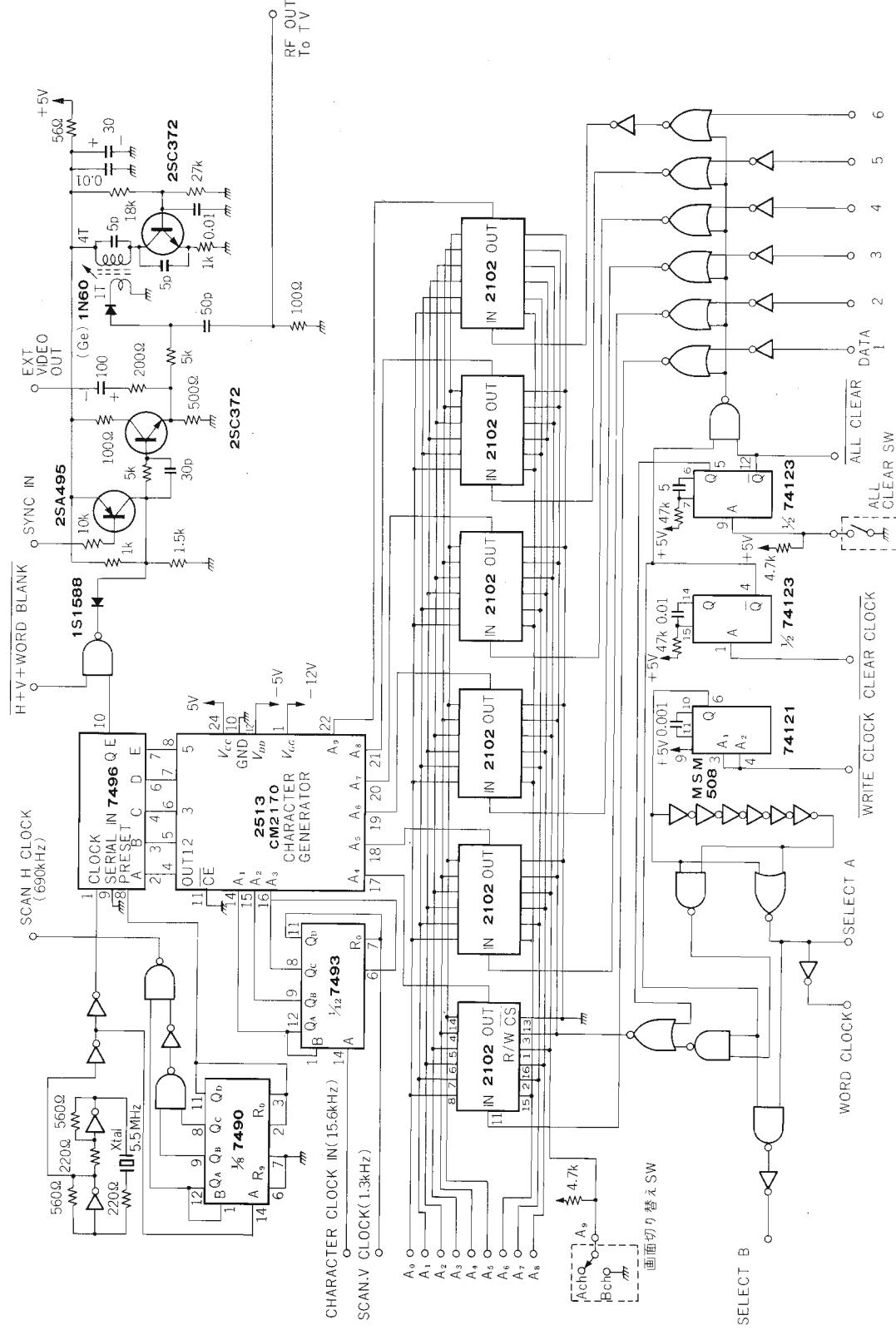


図13 キャラクタ・メモリ回路



<図14> コントロール回路

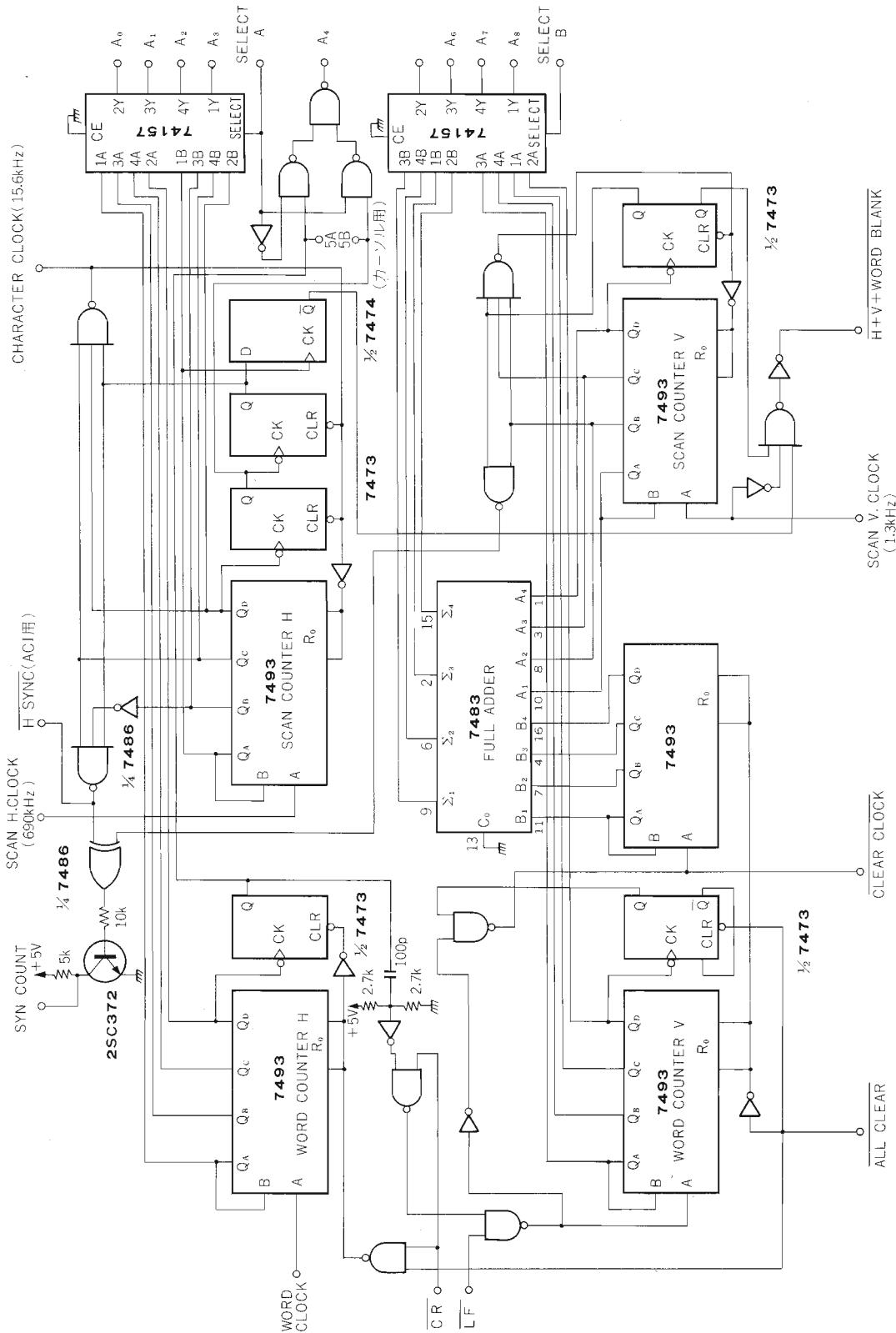
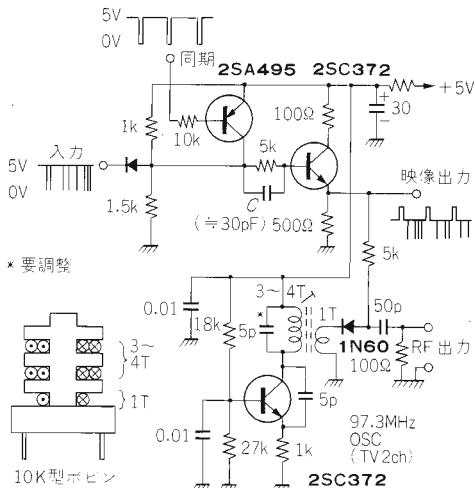


図15 出力回路(RF発振部)



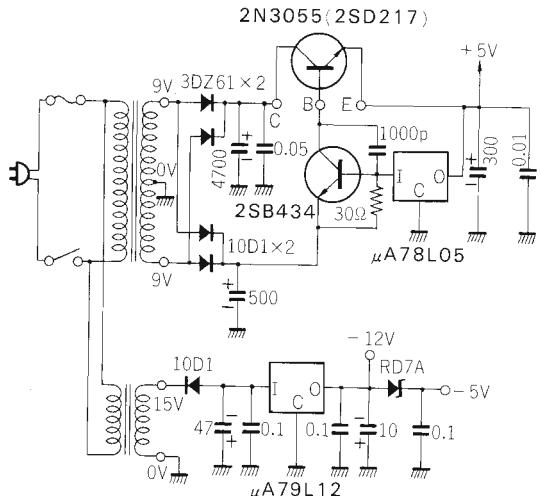
出力回路

7496から現われるキャラクタ・ジェネレータの直列データとブランク信号とで、映像のデジタル信号はできあがったわけですが、これらの信号に同期信号を混合(コンバイン)しなければなりません。この部分で手を抜くと、画面に電源ノイズなどが乗って、非常に見にくくなります。この部分は図15のような回路で混合しています。

なお、この映像出力は負極性信号ですから、トランジスタ式のTVセットの場合は、検波したあとのトランジスタのベースに与えれば、品質のよい画面が得られます。ただし、ふつうの放送電波が映らなくなったりしても、保証の限りではありません。そこで、このような危険(?)をおかさないように、TVのアンテナ端子へ電波で結合をすることにしました。

97.25MHz(2チャンネル:関東地方の空きチャンネル)を発振させ、これと映像信号とをダイオードによって混合するという単純な回路なのですが、比較的

図16 電源回路



きれいな画面が得られます。ここに使用したコイルは、VIF用の10Kタイプのボビンを流用し、0.1mmのエナメル線を巻きました。

電源部

システム電源としては+5V(2A), -5V(30mA), -12V(30mA)の3電源が必要です。図16に回路を示します。

送信部

図17に送信部の回路図を示します。この回路においてキースイッチが押されると、その出力は図3にしたがって組んだダイオード・マトリクスによってBaudotコードに変換されます。そして、スイッチのチャタリングのないときにデータを得るために、ワンショット・マルチを2個入れてあります。つまりキーが押されたのを7430で検出してワンショット・マルチをトリガし、これによって得られたストローブでUARTを制御するのです(図18)。

UARTのTR(Transmitter Register Input)_{1~5}端子に並んだBaudotコード信号はTBRL(Transmitter Buffer Register Load)に加えられた負のパルスによって直列信号に変換され、TRO(Transmitter Register Output)から直列RTTY信号になって出て行きます。この直列信号の速度はTRC(Transmitter Register Clock)によって決まり、通常、送信速度と受信速度は等しいので、RRCと同じ信号を与えておきます。

Baudotコードでは、LTRS, FIGSキーを使って、文字と記号を切り換えて使用しますので、今どちらの状態にあるのか覚えていなければなりません。そこでこの回路では、LTRS, FIGS信号で動作するFFを用いてLEDを点灯させ、記号か文字か一目でわかるようにしてあります。さらにキーを押すと“ピッ”と音

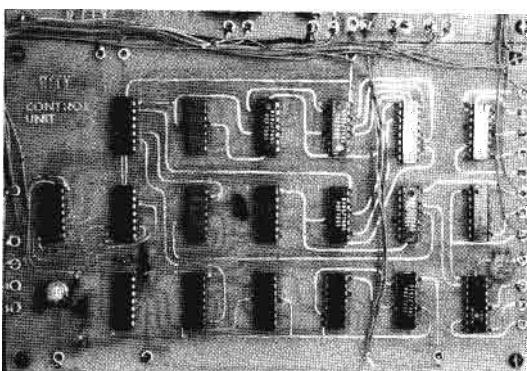
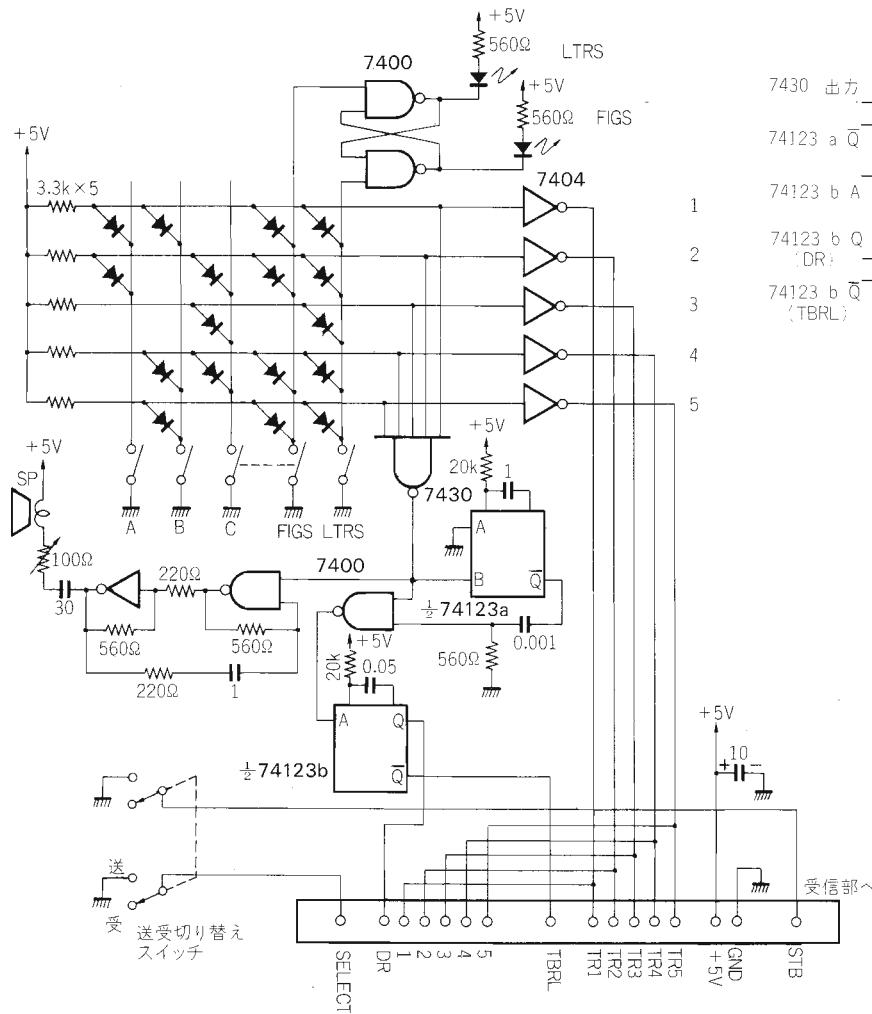


写真4 コントロール部

が出る発振器も組み込んであります。

AFSK(Audio Frequency Shift Keyer)

AFSK部はUARTからの直列信号を、“H”, “L”のレベルに対応させたトーン信号にし、オーディオ信号で取り出す部分です。これによってデータを、ふつうのカセット・テレコなどに録音することができます。



(a) AFSKの回路

また、SSB送信機のマイク端子に接続すれば、臨時に、アマチュア RTTY を楽しむこともできます。

図19(a)がその回路で、移相型の発振回路が基本ですが、FETのゲート・バイアスの電圧変化によって、ドレイン-ソース間の抵抗値が変化することを応用したものです* 図19(b)は移相型発振回路の例ですが、こ

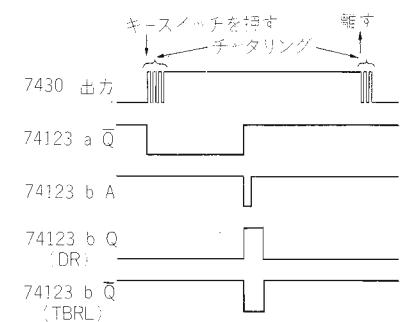
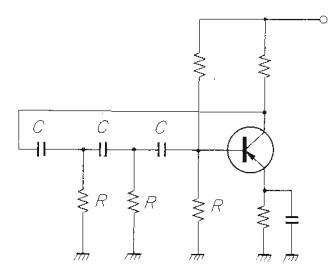


図18
キーボードのタイミング・チャート

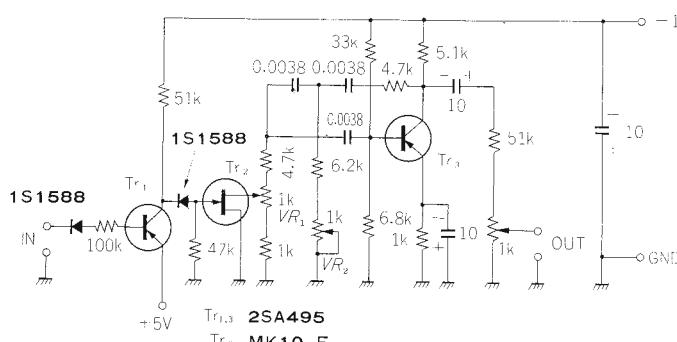
図17
送信部の回路

図19
AFSK回路と移相発振回路*

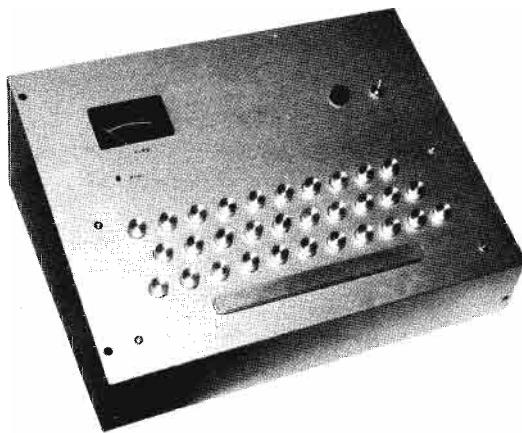
(Ham Journal No.4
に掲載されたもの
をもとにした)



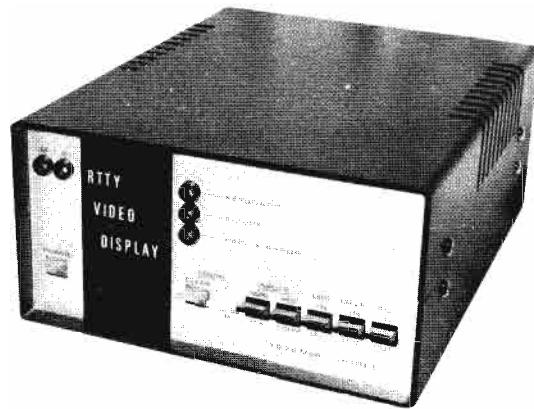
(b) 移相型発振回路



(a) AFSKの回路



〈写真5(a)〉自作キーボード・ユニット



〈写真5(b)〉製作したディスプレイ・システムの外観

の回路において、発振周波数 f_0 は、

$$f_0 = \frac{1}{2\pi\sqrt{6CR}}$$

で求まります。そしていま図(a)の回路において、入力が“H”であるとするならば、Tr₁はC-E間がOFF、よってTr₂のゲートはほぼ-5Vのバイアスが加わり、ドレイン-ソース間は高抵抗を示しますので、この回路では前述の周波数で発振します。

次に入力が“L”になるとTr₁はONですから、Tr₂のゲートはほぼ0Vとなり、ドレイン-ソース間の抵抗が発振回路の要素抵抗と並列に加わり、発振周波数が高くなります。これらの動作によって入力のデジタル信号に応じたFSトーンの出力が得られるわけです。

こうして得られた信号は出力を51kΩと1kΩ(VR)で分割して取り出し、負荷の影響を受けないようにしてあります。発振周波数が2kHz付近での動作ならばこの回路定数でうまく動作するはずですが、周波数を変更される場合は、周波数偏移率をあまり大きくしないように注意してください。

この回路の調整を少し説明しておきましょう。まず入力を“H”にしておき、回路が発振することを確認したならVR₂を回して、発振周波数が2125Hzとなるように調整します。そして入力を“L”に落とし、VR₁で周波数が2295Hzになるようにします。二つの周波数において出力波形がきれいな正弦波となり、かつ出力レベルの変化が小さくなるように、Tr₃のバイアス、帰還抵抗などを調整します。そして、Tr₃には h_{fe} の充分高いものを使用し、VR_{1,2}は安定度を確保するため、めんどうでも固定抵抗と組み合せて使用します。

製作＆調整テスト

だれにでもまちがいなく製作できるよう、原寸のプリント・パターンを示しましたので、試してみてくだ

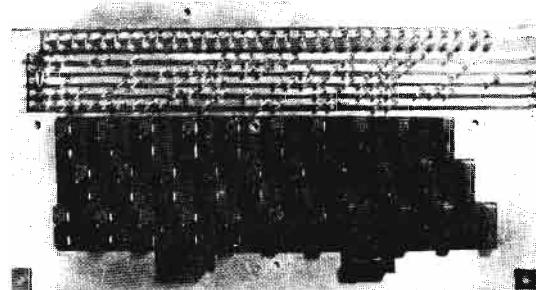
さい（折込ページ）。

なお、ICをハンダ付けする前に、テスターなどを使ってパターンの切れ、短絡などを徹底的にチェックしてください。

キーボード

電気的な部分についてはとくに問題なくできるのですが、メカがからんでくるキーボード部には若干問題があります。この部分はタイプライタのかなめとなる部分ですから、金をかける気になればいくらでもデラックスになるし、そうでなければ……という具合で、ピンからキリまであるのがこの部分でしょう。

ピンのほうはエンコーダ用のICが付いたフルキー



〈写真6〉キーボードのマイクロSWとダイオード・マトリクス

〈図20〉キートップの配列

1 Q	2 W	3 E	4 R	5 T	6 Y	7 U	8 I	9 O	φ P
-		\$	/	&	ニ	'	()	
A	S	D	F	G	H	J	K	L	CR
FIGS	Z	X	C	V	B	N	M	TRS	LF

SPACE

ボード(ただしコードは ASCII, または JIS)があります。私たちの方法はどちらかといえばキリのほうで、マイクロ・スイッチをキー数分だけ並べ、アルミ棒を切断したキートップを取りつけています(写真)。

マイクロ・スイッチはジャンクで@30円くらいのものでしたから、かなり安上がりです。ただし、チャタリングがありますし、キータッチも若干悪いので、問題といえば問題なのですが…。このほかにも方法はありますかと思いますので、各自のくふうによって何とでもなることでしょう。図20にキートップの配置例を示します。上側に記してあるのが FIGS 時の記号、中央に記してあるのが LTRS 時の文字です。

*

調整といっても、大半が完成された(?)デジタル回路ですから、問題は少ないと思います。したがって、ポイントのみ順を追って述べることにします。トラブル・シューティングについては、前掲のタイム・チャートなどを参考にしてください。

では、まずキャラクタ・メモリ基板から始めます。キャラクタ・ジェネレータと RAM はまだソケットに入れないのでおきます。

RF OSC の周波数調整

関東地方では TV の空きチャンネルが 2 チャンネルですから、このチャンネルに信号を入れることになります。この場合は映像の中心周波数は 97.25MHz ですから、ディップ・データを用いてこの周波数に合せます。この際 LC 回路がトランジスタにつながっていて

は測定できませんから、一時切り離して周波数を合せます。おおまかに合せたのち、回路をつないでそれ以降の調整はコイルのコアで合せてください(ディップ・データとにらめっこするよりも、TV に接続して調べたほうが案外簡単にいくはずです)。

東京地区以外の地方では、当然そのエリヤの空きチャンネルを発振するようにしてください。

コントロール、キャラクタ・メモリ基板のテスト

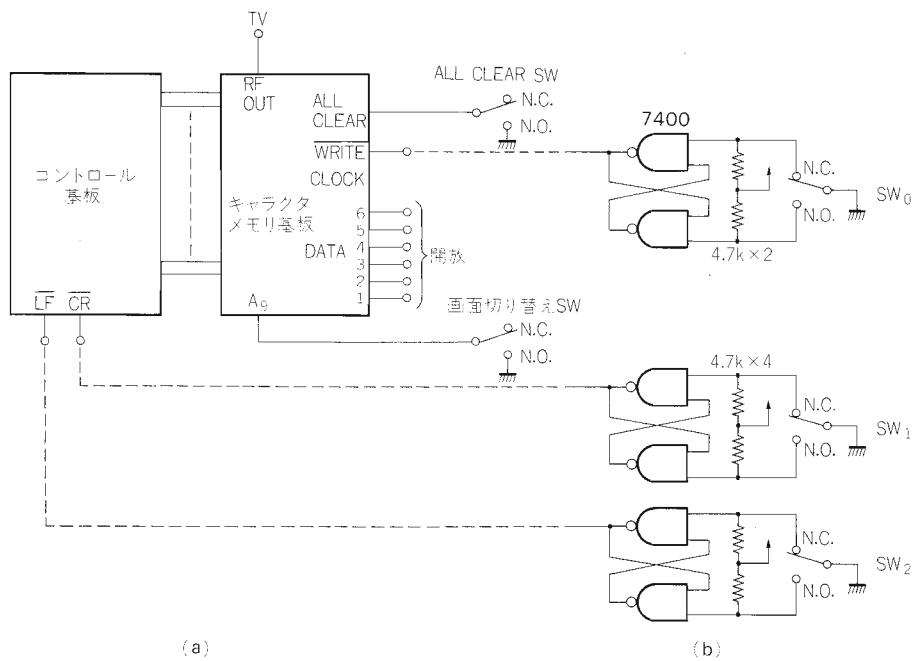
コントロール基板とキャラクタ・メモリ基板間の配線を全部済ませます(図21(a))。(2513と2102はまだソケットに挿さない) TV を接続し電源スイッチを入れます。TV の同期がとれ、小さな白い長方形が画面に 32×16 個出れば OK です。もしだめな時はスキヤン・カウンタ周辺をあたります。

つぎに図21(b)の回路を付加し、キャラクタ・ジェネレータと RAM をソケットに挿します。電源を入れると画面にランダムに文字が出るので、ALL CLEAR SW を押して画面がクリアされるのを確認します。画面を切り替えて同様にテストします。なお、回路が正常に動作していても、電源投入時に文字が出ないことが稀にあります。そのような時には、次に述べる文字の表示テストをした後クリアしてみます。

書き込みのテストは SW₀ を押して行ないます。SW₀ を押すたびに画面に“?”が一つずつ表示されて、512 文字目が書かれると同時に最下行がクリアされます(実は 1 段上にシフトしているのだが、全部同じ文字なのでこのように見える)。それ以後は常に最下行に書かれ

図21

コントロール、キャラクタ・メモリ基板のテスト



THE QUICK BROWN FOX JUMPS OVER
A LAZY DOG. 1234567890

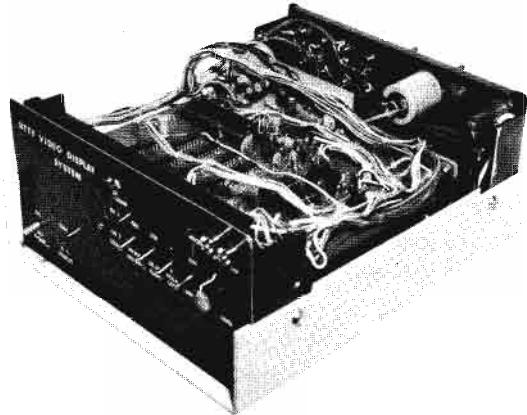
THE QUICK BROWN FOX JUMPS OVER
A LAZY DOG. 1234567890

THE QUICK BROWN FOX JUMPS OVER
A LAZY DOG. 1234567890

THE QUICK BROWN FOX JUMPS OVER
A LAZY DOG. 1234567890

THE QUICK BROWN FOX JUMPS OVER
A LAZY DOG. 1234567890

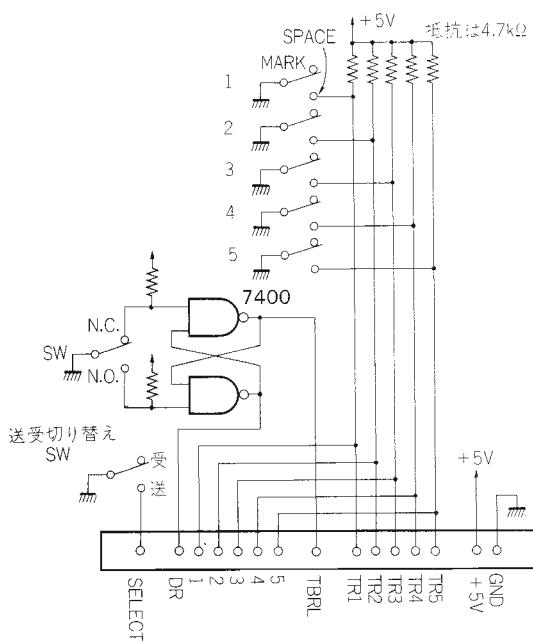
〈写真 7〉表示例 1



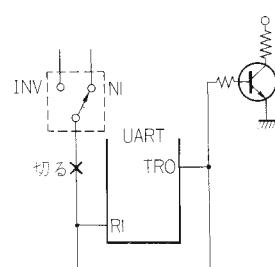
〈写真 8〉自作システムの内部

ていくことを確かめます。

SW₁, SW₂を押して復帰, 改行が行なわれるかどうか、またD₁～D₆を適当にGNDに落としてそれに相当する文字(表2のB₀～B₆をD₁～D₆と読みかえる)が表示されるかどうかを調べます。



〈図22〉
簡易キーボード



〈図23〉
UART のテスト

ここまでうまく動作すればCRTディスプレイの心臓部は完成です。

次のテストのために図21(b)の付加回路ははずしておきます。

キーボードとコード変換部のテスト

UARTコード変換部基板に、コントロール基板とキヤラクタ・メモリ基板、さらにキーボードの出力を接続します。UARTはソケットに挿さないでおきます。送受切り替えSWを送信側に倒し、キーを押してみます。この時画面にそのキーと同じ文字・記号が出ればOKです。CR, LFについても動作を確認します。

もしキーボードがない場合には、図22のような簡易キーボードを作ります。図3を見てデータをセットし、SWを押すとその文字が画面に表示されます。

UART のテスト

図23のようにRIとTROとを接続しておきます。UARTをソケットに挿し、送受切り替えSWを受信側にしてキーボードのキーを押すと、画面上に文字・記号が表われます。このテストはUART(送信部)で並列から直列に変換された信号を再びUART(受信部)に入れ直列から並列信号に変換しているものです。

ここまで終わったらRI, TROの配線を元に戻しておきます。

PLLの自走周波数調整

マークとスペースの周波数を2125および2295Hzとしたので、この間の中心にPLLの自走周波数を持てきます。つまり、AF IN端子に入力がない状態で、NE565のピン④の波形が2210HzになるようにVRを回します。マーク、スペースの周波数を他のものにした場合(たとえばもっと低くした場合)は、それらの周波数の中心に合せてください。

UARTのクロック周波数の調整

受信するテレタイプ信号のポートレイトの16倍のク

〈表3〉 無線気象通報(アマ無線バンド近傍のもの)

周波数	出力	周波数偏移	
JMI-2	7376kHz	5kW	±400Hz アッパーシフト
JMG	3670kHz	2kW	±425Hz アッパーシフト
JMG-3	7402.5kHz	5kW	±425Hz アッパーシフト

なお符号速度はいずれも50ボートで、内容は5桁の数字を打っていることが多い。

ロックを与えます。UARTは、送信部と受信部は別個に動作させることができるので、送受信を同一のクロックとし、周波数を固定にしてあります。受信専用とする場合は、クロック回路をVR可変型とし、外部から調整できるようにするのも方法かと思います。

クロックは正確であるほうが望ましいのですが、1%程度の誤差は気にする必要はありません。調整はNE555の半固定VRを回して周波数を727Hzにセットするだけです。

なお、後述の気象通報やニュースキャストを受信する時は800Hzに合せてください。

TV画面上の位置調整

さて、めでたくTVの画面上に文字が出たとしても、その位置は必ずしも中央部に位置するとは限りません。そのTVが博物館でしか見られないような(?)真空管式の白黒TVであったり、最近のカラーTVの場合ならことは簡単に解決できます。なぜならば、それらのTVには「ポジション」や「サイズ」などの調整用VRが必ず並んでいるからです。

問題はTVのケース幅がCRTギリギリといったような小型のトランジスタ、IC化されたタイプのTVです。このようなTVの場合、回路に手を加えないで、簡単にサイズを変更する方法をお教えしましょう。それは適当なフェライト・コアに、充分太いエナメル線を10~20回くらい巻き、これを偏向コイルと並列に入れれば、ピタリと画面内に収まってくれるはずです。簡単な方法ですが、覚えておいて損のない方法です。

使 い 方

さて、いよいよ受信です。受信機のスピーカと並列にUART・コード変換基板のAF INをつなぎます。アマチュアRTTYの電波が聞こえない時には無線気象通報を受信してみましょう(表3)。

本システムと送信機を結合するには次の二つの方法があります。一つの方法はSSB送信機のマイク端子にAFSKからの出力を加える方法です。他方はUARTからのシリアル出力を用いて、ダイオードSW回路によって、送信機のVFOや水晶発振器にコンデンサ(小容量の)を付加したり、切り離したりする方法です。いずれの方法でもかまいません。

RTTYといえば、必ずロール・ペーパー、インク・リボン、油さし……などを思い浮かべるのですが、電子化された本システムの場合はそういった消耗品がありませんので保守が楽です。もっともテレタイプの特長であるハード・コピーはとれないのですが…。

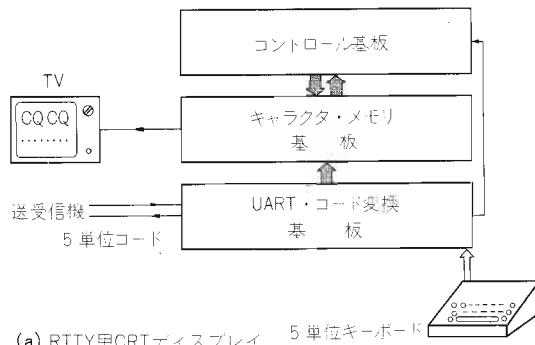
マイコン用I/Oへの改造

ではラジオ・テレタイプ(RTTY)用CRTディスプレイを、マイクロコンピュータ用I/Oとして使用できるように改造した結果をこれから述べることにします。

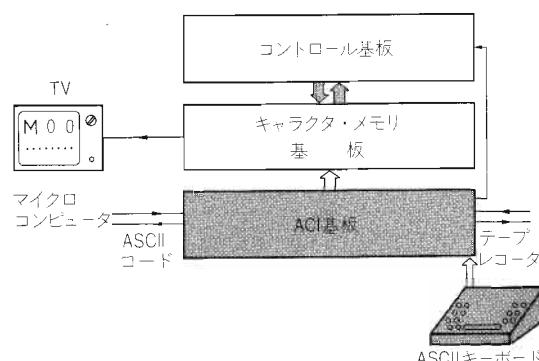
結論からいえば図24のように、キーボードと基板を1枚入れ替えることになります。この新たに製作する基板をACIユニット(Asynchronous Communication Interface)と呼ぶことにします。ボードの名称はいかめしいのですが、やらせていることはデータの直-並列変換と、オーディオ・テープレコーダ・インターフェースだけなのです。

本稿ではさらにマイコンのI/Oと直接には関係がないのですが、画面にカナ文字とカーソルを出す方法に

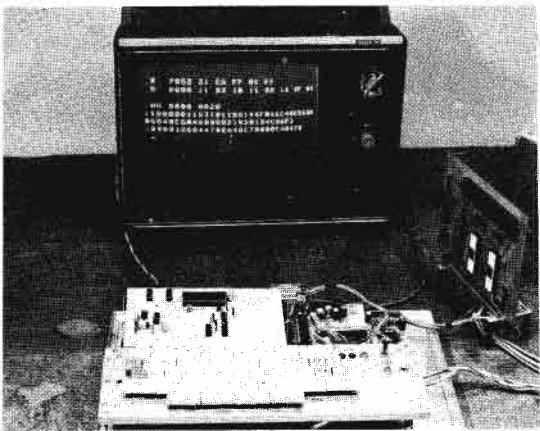
〈図24〉 RTTY用CRTディスプレイとマイコン用I/O
CRTディスプレイ



(a) RTTY用CRTディスプレイ 5単位キーボード



(b) マイクロコンピュータ用CRTディスプレイ



〈写真9〉操作中の改造マイコン用システム

ついても述べたいと思います。

マイコンのI/O

現在、マイクロコンのI/Oとして、もっともポピュラーなものとしてはASR-33があります。このI/Oは入力、出力および共通線の3本の電線をつなぐことによって、計算機と完全にインターフェースができる便利なものです。しかし高価で、アマチュアのポケット

・マネーではおいそれと購入できるものではありません。

そこで、このASR-33と完全コンパチブルなものを目標に改造したのが、今回のCRTキャラクタ・ディスプレイです。ただし、入出力はカレント・ループではなく、TTLレベルとなっています。

使用コード

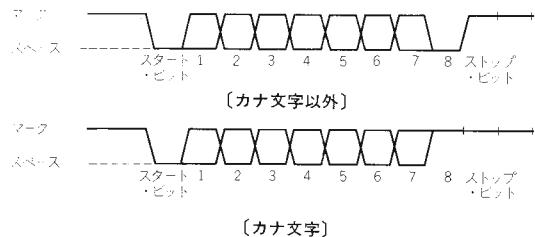
ASR-33ではASCII(American Standard Code for Information Interchange)コードと呼ばれるものを使っていますが、今回使用するコードは、JISコードの中の英数字、カナ文字、および機能コード(CR, LF)を用います。JISコードを選んだのは、JISコードの中の英数字と機能コードが、一部を除いて(JISで“¥”の記号が、ASCIIでは“＼”の記号となっている)共通に使えること、カナ用のキャラクタ・ジェネレータICを増設するだけで、カナ文字の表示が可能になることの二つの理由からです。表4にJISコードとASCIIコードとの対応表を示します。

また、パリティ・ビットは設けず、コード長は8ビットとなります。したがって、パリティ・ビットが付加されたASCIIコードでは、パリティ・ビットをすべて“0”としたものと等価になります。この様子を図25に示します。図からもわかるように、8ビット目(パ

〈表4〉JISコードとASCIIコードの対応

JISコード											ASCIIコード												
ビット番号	b ₈	b ₇	b ₆	b ₅	b ₄	b ₃	b ₂	b ₁	行	列	0	0	0	0	0	0	0	1	1	1	1	1	1
	0	0	0	0	0	0	0	0	(TC ₇)DEL	S P	φ	@	P	P	↑	↑	↑	↑	↑	↑	↑	↑	↑
	0	0	0	1	1	0	0	1	(TC ₁)SOH	DC ₁	†	1	A	Q	a	q	0	1	0	1	1	0	1
	0	0	1	0	2	0	0	2	(TC ₂)STX	DC ₂	”	2	B	R	b	r	0	1	1	0	0	1	0
	0	0	1	1	3	0	0	3	(TC ₃)ETX	DC ₃	#	3	C	S	c	s	0	1	0	1	0	1	0
	0	1	0	0	4	0	0	4	(TC ₄)EOT	DC ₄	\$	4	D	T	d	t	0	1	1	1	1	1	1
	0	1	0	1	5	0	0	5	(TC ₅)ENQ	(TC ₈)NAK	%	5	E	U	e	u	0	1	1	1	1	1	1
	0	1	1	0	6	0	0	6	(TC ₆)ACK	(TC ₉)SYN	&	6	F	V	f	v	0	1	1	1	1	1	1
	0	1	1	1	7	0	0	7	BEL	(TC ₁₀)ETB	,	7	G	W	g	w	0	1	1	1	1	1	1
	1	0	0	0	8	0	0	8	FE ₀ (BS)	CAN	(8	H	X	h	x	1	0	0	1	0	1	0
	1	0	0	1	9	0	0	9	FE ₁ (HT)	EM)	9	I	Y	i	y	1	0	0	1	0	1	0
	1	0	1	0	10	0	0	10	FE ₂ (LF)	SUB	*	:	J	Z	j	z	1	0	1	0	1	0	1
	1	0	1	1	11	0	0	11	FE ₃ (VT)	ESC	+	;	K	[k	!	1	0	1	1	0	1	1
	1	1	0	0	12	0	0	12	FE ₄ (FF)	IS ₄ (FS)	,	<	L	¥	l	!	1	1	0	1	0	1	0
	1	1	0	1	13	0	0	13	FE ₅ (CR)	IS ₃ (GS)	-	=	M]	m	!	1	1	0	1	0	1	0
	1	1	1	0	14	0	0	14	SO	IS ₂ (RS)	.	>	N	^	n	-	1	1	1	0	1	0	1
	1	1	1	1	15	0	0	15	SI	IS ₁ (US)	/	?	O	-	o	DEL	1	1	1	1	0	1	0

図25 本機で用いる8ビット・コード

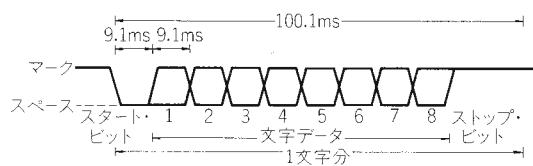


リティ・ビット)を無視することで、完全に ASCII コードとして用いることができるのです。

スピード(バー・レイト)

ASR-33では、ほとんどすべてが 110 ボーを用いています。バー(Baud)とは、通信速度の単位で、ビット / 秒と同じで 1 秒間に最短パルスが何個送れるかということを表わしています。したがって 110 ボーというの は、最短パルスの幅が、 $1/110\text{秒} \approx 9.1\text{ms}$ ということになります(図26)。本機の場合、すべてが電子素子であり、通信速度はかなり速いものまで扱えるのですが、ASR-33 コンパチブルということから、110 ボーとし てあります。また 330 ボーにも簡単に切り替えができ

図26 8ビット・コードの構成(110バーの場合)



るようになりました。

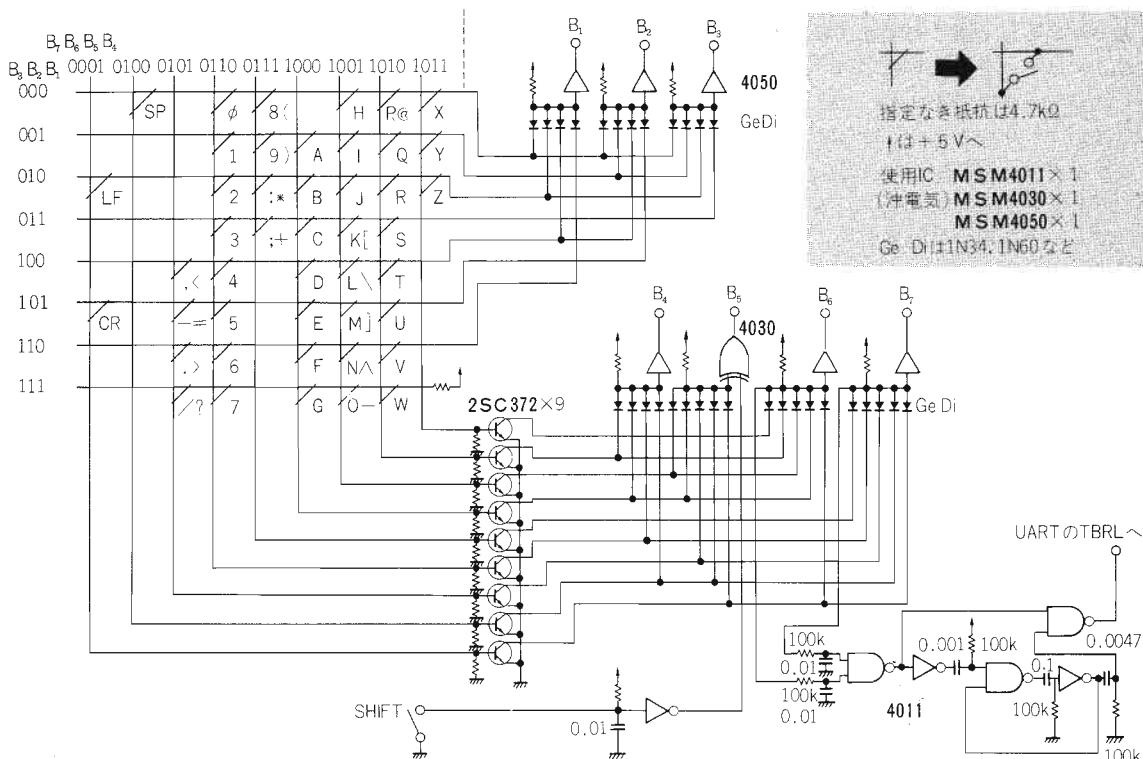
キーボード・エンコーダ

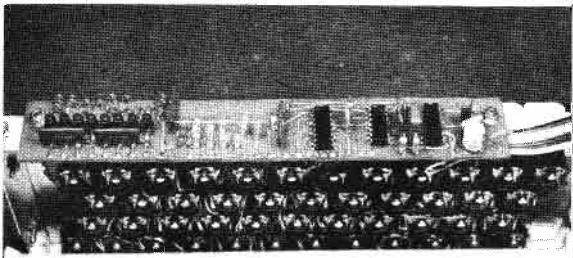
エンコーダにはダイオード・マトリクスから、専用 LSI による方法までいろいろあります。それぞれ一長一短があり、どれを採用するか迷うところですが、ここでは作りやすさとコストに重点を置いて図27のような回路を使うことにしました。

ASCII の下位 3 ビットが「行」に、上位 4 ビットが「列」になるようにキー・スイッチをマトリクス状に配線します。「行」は電圧で、「列」のほうはトランジスタを使って電流で検出し、ダイオードでエンコードしたのち、C-MOS のバッファを介して出力としています。シフト機構については、5 ビット目のみをシフト時

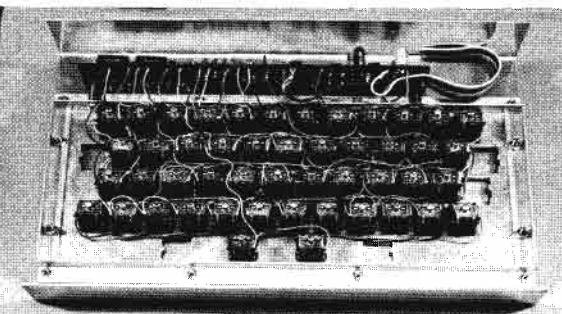
図27 キーボード・エンコーダの回路

キー・マトリクス部 エンコーダ部

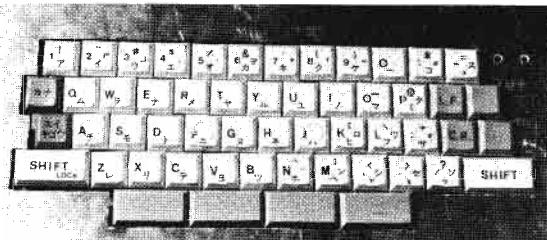




〈写真10〉 キーボード・エンコーダ



〈写真11〉 自作キーボードの裏面



〈写真12〉 自作キーボードの例。いずれも電卓用テンキーを改造し、表面にレタリングを貼っている

に反転させることで簡単にすませています。したがって、シフト時に記号の割り当てのないキーを押すと、とんでもない文字の符号が出力されてしまいます(たとえばシフト・キーを押しながらスペース・キーを押すと“”が出る)。誤動作を禁止する回路は省略しました。

ここではビット6, 7が“11”となるような符号は使っていないので、それぞれのビットの負論理 OR を取って、キーが押されたことを検出しています。そして、チャタリングがなくなったころにUARTのTBRL用パルスを発生させるために、ワンショット・マルチバイブレータを用いています。

図28はキー・スイッチがどうしても入手不可能な場

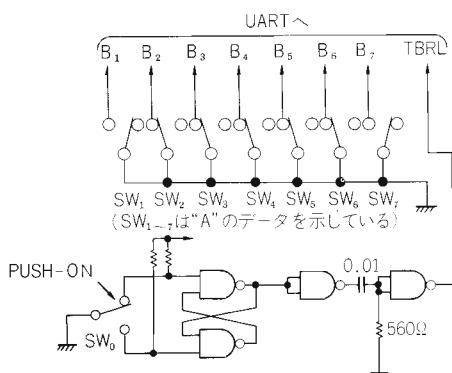
合や、きちんとしたキー・ボードを作る前に回路をテストしてみたいという場合などに使える簡易キーボードです。使い方は簡単で、データどおりスイッチ SW₁～SW₇をセットし、SW₀を押せばUARTのTROからシリアル・データが得られます。

なお、ここで使用したUART TMS6011の入力は、1 TTL ロードですから、ビット5のExclusive OR と TBRL パルスの NAND に使う IC の選択には注意してください。沖電気の MSM4000 (C-MOS) シリーズ以外の IC を使用する場合には、バッファが必要となることがあります。

エンコーダにおける8ビット目の処理

カナ文字を扱わない場合は、8ビット目は何も接続しませんが、カナ文字を扱う場合はエンコーダに8ビット目の出力が出ていないので、図29に示す回路を付加してください。回路図中のSWはノーマリ、オープン型のもの、すなわち押した時だけONするようなものを使用してください。

〈図28〉 簡易キーボード



〈図29〉 エンコーダに付加する8ビット目

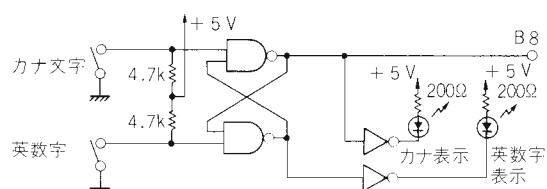
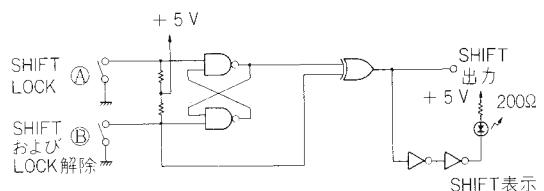


図30 シフト・ロックの回路

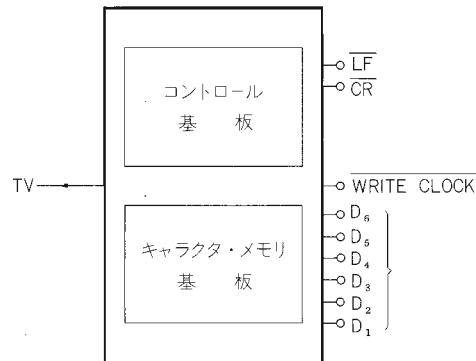


使い方は簡単で、英数字を使いたい場合は英数字の SW を押します。すると英数字を示す LED が点灯して出力が“L”となってビット 8 が“L”，すなわち英数字のコードが output されます。カナ文字の場合も同様に操作することによってビット 8 が“H”，すなわちカナ文字のコードが出力されます。

次にキーボードの SHIFT・LOCK(電子的な)の回路を図 30 に示します。回路図中の SHIFT 出力は、エンコーダの SHIFT・SW があった所に接続します。これも使い方は簡単で、SHIFT・LOCK したいときは ① SW を押します。LOCK を解除したいときは、あるいは押したときだけ SHIFT したいときは ② SW を押します。いずれの場合も SHIFT 時には、SHIFT を示す LED が点灯します。

マイコンの I/O として本機を使う場合は、SHIFT し

図31 CRT ディスプレイ中心部を外から見ると

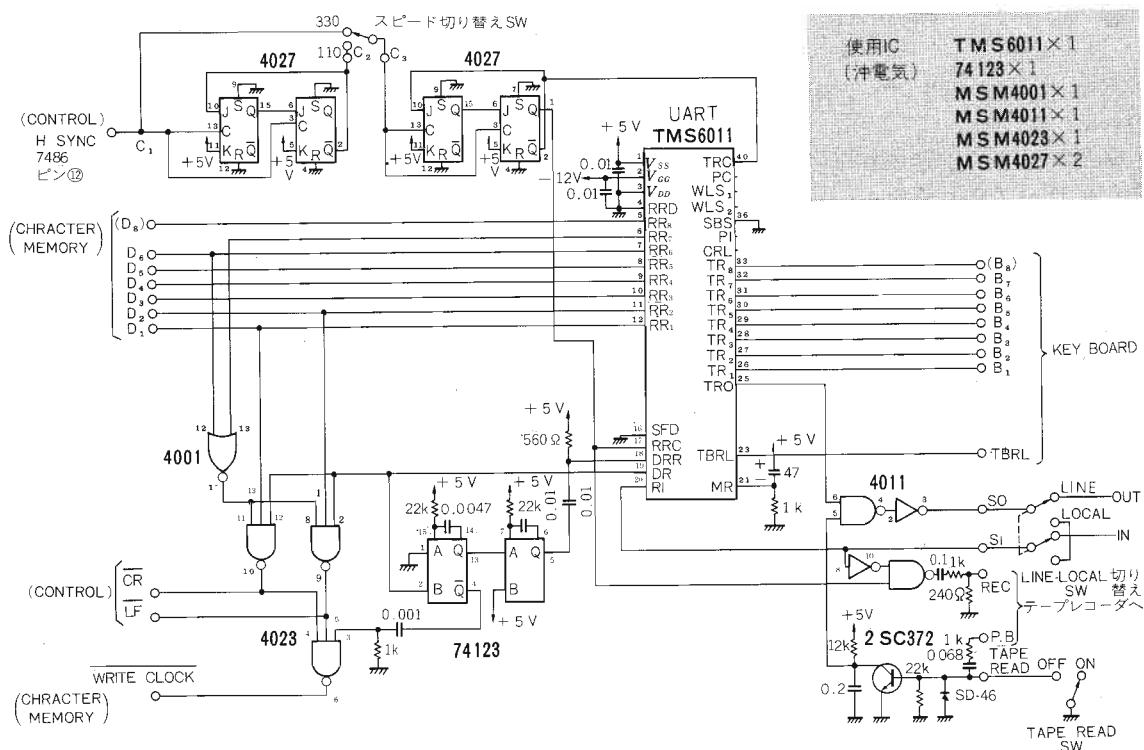


たまま使うということはほとんどありませんが、回路も簡単ですし、付けておくとなにかと便利だと思います。しかし、その場合 SHIFT したことを忘れて使っていると押したキーと異なった文字が出力されるので注意してください。

ACI ユニット

新しく作る ACI 基板の説明に入る前に、コントロール基板とキャラクタ・メモリ基板を動作させるに必要な信号について少し整理しておきたいと思います。図

図32 ACI ユニットの回路

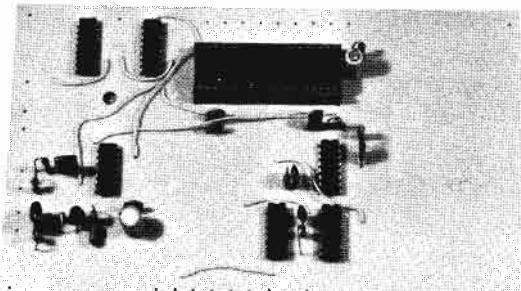


〈写真13〉 ACI ユニット ⇒

(右側のスペースはフリー・エリアになっている)

〈表5〉 CR, LF のコード

	CR B ₇ B ₆ B ₅ B ₄ B ₃ B ₂ B ₁	LF B ₇ B ₆ B ₅ B ₄ B ₃ B ₂ B ₁
ASCII	0 0 0 1 1 0 1	0 0 0 1 0 1 0
図32	0 0 ×××× 1	0 0 ×××× 1 ×



31に示したように、まず文字のデータとして ASCII の下位 6 ビット、D₁～D₆ が必要です。

次に書込み用のパルスです。つまり D₁～D₆ の端子に画面に出したい文字のデータを用意し、WRITE CLOCK 端子に負のパルスを加えれば、文字が TV に映るわけです。

コントロール用信号としては CR と LF がありますが、それぞれの端子に負のパルスを加えれば、復帰と改行が行なわれます。

さて、これで ACI ユニットに要求される機能がはっきりしてきました。つまり、

- ① キーボードからの ASCII(JIS)並列信号を直列信号に変換してラインに出す
- ② ラインからの直列信号を並列信号に直す
- ③ 書込みパルスと CR, LF の信号を作る
- ④ オーディオ・テープ・インターフェースの録音再生回路

大別して以上の四つです。ACI ユニットの回路を図32に示します。

①と②は UART の語長を 8 (WLS1, WLS2 端子とともに "H" にする)、ストップ・ビット長を 1 (SBS 端子を "L")、パリティ・チェックなし (PI 端子を "H") で動作させます。8 ビット目 (B₈, D₈) はカナ文字を表示する時のみ使用し、英数字のみの場合は開放にしておきます。TMS6011 は MOS のデバイスですが、入力端子は内部にプルアップ抵抗を持っているため、開放にすれば "H" レベルを与えたことになります。

ライン・スピードを決めるクロックですが、TMS6011 は使用するポー・レイトの 16 倍の周波数を与えればよいので、110 ポーでは 1760Hz, 330 ポーでは 5280 Hz が必要です。このためにわざわざ別の発振器を作るのは面倒なので、ここでは上記の周波数に近い値を選んで、そこまで、ディスプレイ・システムのメイン・クロックを分周することにしました。こうすることにより、上記の周波数からは少しづれますが、原発振が水晶なので非常に安定したライン・スピードが得られます。

幸いなことに CHARACTER CLOCK として水平同期周波数の 15.625kHz がすでにありますので、それを 1/9 に分周して 1736Hz, 1/3 に分周して 5208Hz を UART のクロックとしています。いずれの周波数も半端ですが、1760Hz, 5280Hz に対して、それぞれ -1.4% 程度の誤差ですから、110 ポーに限りなく近くという具合にはいきませんが、差しつかえのない値でしょう。

分周は同期式の 3 進カウンタで行なっています。110 ポーのときは 2 段、330 ポーのときは 1 段を使います。同期式にした理由は、3 進カウンタの場合非同期式よりも回路が簡単になるためで、他意はありません。なお、TMS6011 のクロック端子 TRC と RRC も、1 TTL ロードですので駆動可能な充電気の C-MOS FF を使い、別々の出力からドライブしています。

③の CR, LF のデコードは表5のように簡略化しています。また書込みパルスは CR, LF のときには禁止されるようになっています。TMS6011 の出力は TTL1 個を駆動する能力がありますが、その全部をキャラクタ・メモリ基板をドライブするのに使ってしまうため、ここでのデコードの回路には TTL は使用できません。

図中の 74123 の 1 段目は、デコード回路の遅れを補なうために設けたもので、2 段目はメモリへの書き込み



〈写真14〉 表示例(2) 左下がカーソル

が終了するまでデータを保持しておくものです。

①の録音回路はゲート IC で作った、スペース・オシリのトーン・バーストで、ハイパス・フィルタを通してテープレコーダの AUX 端子に接続します。

再生回路は、以前トランジスタ技術誌に載っていた

回路で、トーン・バースト信号をトランジスタで検波し、コレクタ回路の CR の時定数で一種のワンショット・マルチバイブレータを構成し、復調するものです。なお、この回路はテープレコーダの周波数特性から、110 ポーのときのみ使用することにします。

<図33>

ACI ユニットのプリント・パターン図

(→ジャンパ線、フリー)
・エリアには DC-DC コンバータなどを組む

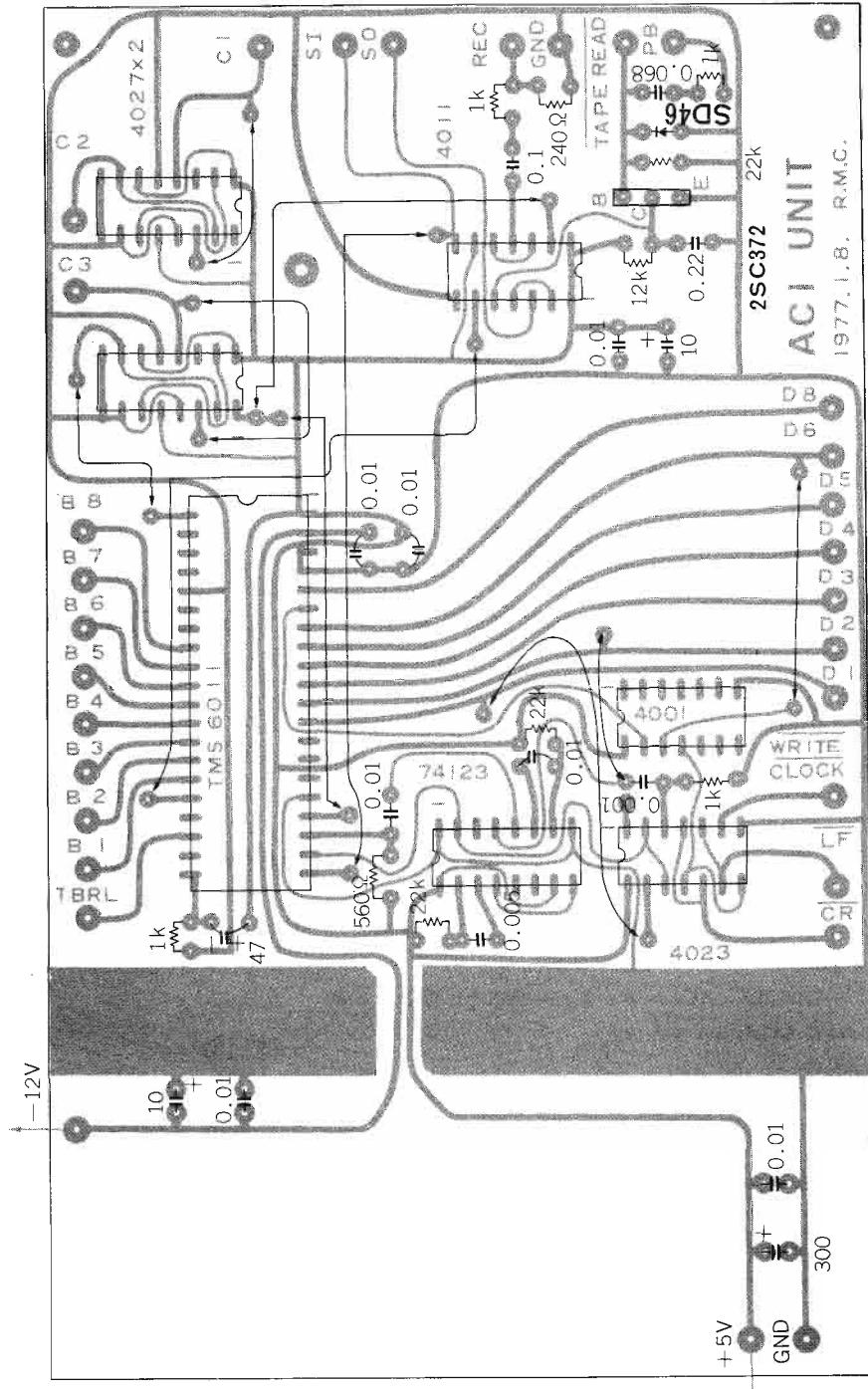
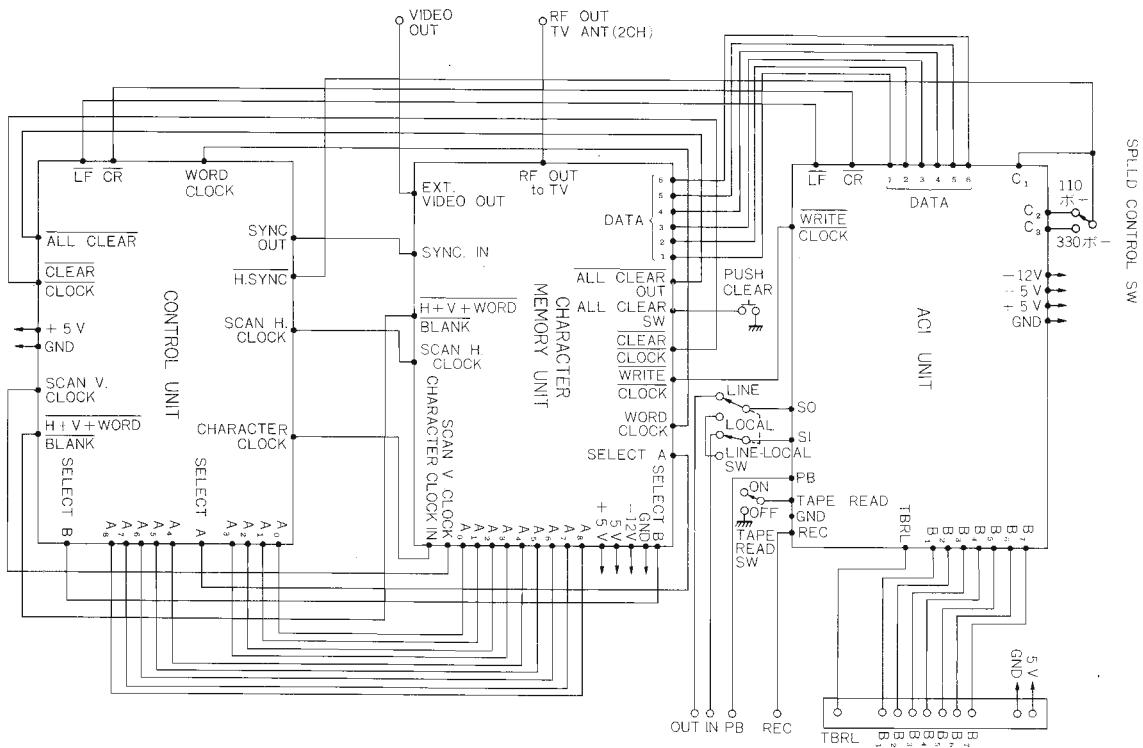


図34 CRT ディスプレイの結線図(ACI ユニット以外は前項)



製作と使い方

キーボードを作る場合、キー・スイッチの入手がネックとなると思いますが、最近では単体のキーが1個250円前後で売られているようですので、ジャンク品がない場合はそれらを購入するのも一つの手でしょう。

図33(前ページ)にACI回路のプリント・パターンを示しておきますので、参考にしてください。片面のパターンとしたため多少ジャンパ線がありますが、片面基板のほうがエッティング液の疲労も少なく、またフォト・エッティングで基板を製作する際にも、作業時間の短縮ができる有利だと思います。

エンコーダ、ACIユニットとも調整を必要とする部分はとくにありません。

使用法

全体の結線図を図34に示します。LINE-LOCAL切り替えSWをLOCAL、TAPE READ SWをOFFにして、キーボードを押すと押された文字が表示されます。この場合、ASR-33における、LOCAL MODEと同じことになります。

テープレコーダにデータを録音させたいときは、テープレコーダを録音状態にしてRUNさせておき、ライン・スピードを110ポーにすれば上述の押された文字が録音されます。録音時のライン・スピードと再生

時のライン・スピードは同じでなければなりませんので注意を要します。

なお、再生はTAPE READ SWをONにし、テープレコーダを再生状態にしてRUNさせれば、録音されていた内容が画面に表示されます。

次にマイコンと接続する場合ですが、その場合はマイコン側にASCIIのシリアル・コードを用いたモニタ・プログラムが用意され、なおかつそのプログラムが実行されていることが必要です。まず、LINE-LOC AL切り替えSWをLINE、TAPE READ SWをOFF、ライン・スピードを110ポーにします。この状態で出入力だけがTTLレベルになったASR-33と同じになります。

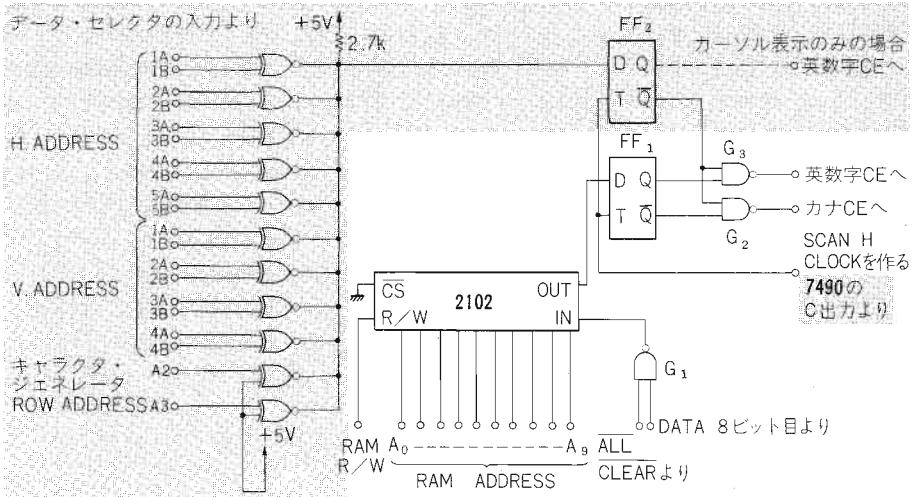
附加回路

さらに本機を使いやすくするために、カーソルやカナ文字を表示する回路を図35に示します。カーソルとは、次に表示する場所を示す線のことで、本機では回路を簡単にするためにカーソルの位置を自由に移動できるようなカーソル・コントロールの機能は持たせていません。

回路の動作ですが、最初にカナ文字表示回路について説明します。図35においてB₈のデータは、カナ文字のときは“1”，英数字の時は“0”となり、G₁のゲー

〈 35〉

カーソルおよびカタカナ表示回路



トは通常生きていますので、データはキャラクタ・ジェネレータの6ビットの文字データと同時に、M（メモリ）に書き込まれます。MはALL CLEARによって最初すべて“1”が書き込まれています。

すなわち、ALL CLEAR によって画面にはすべてスペースが表示されますが、そのスペースは英数字のキャラクタ・ジェネレータのスペースが表示されているわけです。

図35のD FFのTには図9(図13参照)——キャラクタ・メモリ回路)のタイム・チャートに示されるCの波形がクロックとして入力されていますから、Cのポジティブ・エッジによって、D FFのDの入力がQに出力されます。ここでCのポジティブ・エッジというのは ADDRESS が変わってから 720ns のちであり、メモリ(M)のアクセス時間は 650ns(標準)ですから、そのときにはメモリのデータが FF₁ に入力されます。

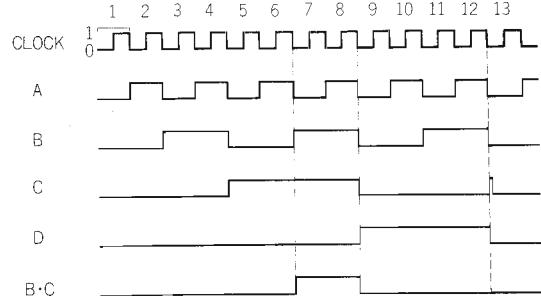
そして G₂, G₃ はカーソルを表示する瞬間以外は生きています。したがって、FF₁ の Q が “1” のときには英数字の CE が “0” に、FF₁ の \bar{Q} 出力が “1” のときにはカナ文字の CE が “0” となり、それぞれのキャラクタ・ジェネレータがセレクトされます。なお、キャラクタ・ジェネレータの IC は、出力がスリー・ステートになっており、CE を “1” にすることによって、出力がハイインピーダンスになるので、英数字とカナ文字のキャラクタ・ジェネレータは CE を除いてすべて共通に接続して使用することができます。

次にカーソルの表示ですが、これは先述のキャラクタ・ジェネレータの CE を“1”にすることによって、出力がハイインピーダンスになることを利用します。つまり、TTL の入力はハイインピーダンスのときは“1”とみなすことができるからです。図 14においてWORD COUNTER のアドレスは、次にエンコードされる

メモリのアドレスを示しています。そこで、WORD COUNTERのアドレスと、SCAN COUNTERのアドレスとが完全に一致したとき、カーソルを表示すればよいことになります。

今回は TTL ファミリの中から、Ex-NOR（オープ

アドレスのタイム・チャート



〈図37〉 カーソルの位置

D	C	B	A						
0	0	0	0	1					
0	0	0	1	2					
0	0	1	0	3					
0	0	1	1	4					
0	1	0	0	5					
0	1	0	1	6					
0	1	1	0	7					
0	1	1	1	8					
1	0	0	0	9					
1	0	0	1	10					
1	0	1	0	11					

ン・コレクタ型) 74LS266 というゲートを見つけたので回路が非常に簡単になり、それぞれのアドレスのEx-NORを取り、出力はすべてワイヤード OR であります。また、このままではカーソルがバー(棒)でなく長方形に白く抜けてしまうので、バーとするためキャラクタ・ジェネレータのROWアドレスのB, Cからの信号も加えています。この様子を図36のタイム・チャートに示します。このタイム・チャートからもわかるように、図37の斜線で示した部分がカーソルとなつて表示されるわけです。なお、カーソルのみを表示させたい場合は、図35の回路図中、影のついている部分のみを付加するだけです。

参考文献

- (1) 谷村：無線通信工学，コロナ社，1960年
- (2) 斎藤：RTTY用AFSK発振回路の考え方と製作，Ham Journal, No4, 1975年, CQ出版社
- (3) 関根：エレクトロニクス RTTYシリーズ(4), Ham Journal, No4, 1975年, CQ出版社
- (4) その他各社のカタログ・データ、シート・カタログを参考にしました。
- (5) 松本：誌上マイクロコンピュータ・セミナー、トランジスタ技術, 1976年, 12月号, CQ出版社
- (6) 富永：データ・エントリー キーボードの試作, CQ ham radio, 1976年, 7月号, CQ出版社