PREVENTION OF PIRATE TELEVISION/PAY TV

A thesis, in part fulfillment of the

Masters Diploma in Electrical Engineering(L/C).

By **Trevor Charles**

DECLARATION

I, Trevor Charles, hereby declare that the following thesis is my own work and has not been submitted before at the Cape Technicon or at any other institution for a Masters Diploma in Electrical Engineering (L/C).

....

94 - 03 - 02

ABSTRACT

This thesis," Prevention of Pirate TV/Pay TV ", implements a method whereby television viewing could possibly be controlled by payment. Each television set would have a microcontroller board with a unique preprogrammed security code and a display installed and a remote control (ram card). The remote control is programmed with the desired viewing dates and security codes by the authority in charge by using the programming unit. A cash transaction then takes place.

The client has 24 hours to download this information into the television's microcontroller before it becomes useless. Software checks occur to determine downloading. With a successful download, the Tv set is switched on. When the end date occurs, the Tv set switches off and the displayed dates are cleared from the display. The 'Licence Due' light emitting diode comes on. For the television to be switched on, the above procedure must be repeated.

Any calender duration maybe programmed for the viewing dates.

ACKNOWLEDGEMENTS

I hereby thank the staff of RDC, Barrack Street, Cape Town for their help and assistance. A word of thanks to National Panasonic, Parow for repairing the television and to the library staff of the Argus for their assistance.

CONTENTS

1. INTRODUCTION

This thesis involves an investigation into a suitable design and implementation of a system whereby pirate viewing of television could possibly be prevented and is based upon a payment/viewing principle.

This was done due to the fact that the SABC TV has to some extent no control over the payment of TV licences, thereby incurring revenue losses and subsequently increasing the TV licence periodically. A graph, FIG 1.0, shows the increase in Tv licence since the introduction of TV in 1974. FIG 1.1 shows the losses incurred as reported by the Argus newspaper. The only method of control implemented was to draw up a database, over a period of time, of its clients. SABC TV contracted the SAPO to collect the licence fee on its behalf and to send 'TV LICENCE RENEWAL' reminders. The SAPO Inspectors also did a house to house inspection of TV licences of areas which were picked at random. At first a fine would be imposed, later the TV would be sealed if the licence had not been renewed. These methods proved not to be very successful as the database was not complete and not every household could be inspected. Thus, the SABC TV lost revenue due to unpaid licences. The following design criteria were considered to prevent pirate viewing.

(i) to design a system which is 'hands/finger' free at the client's end.

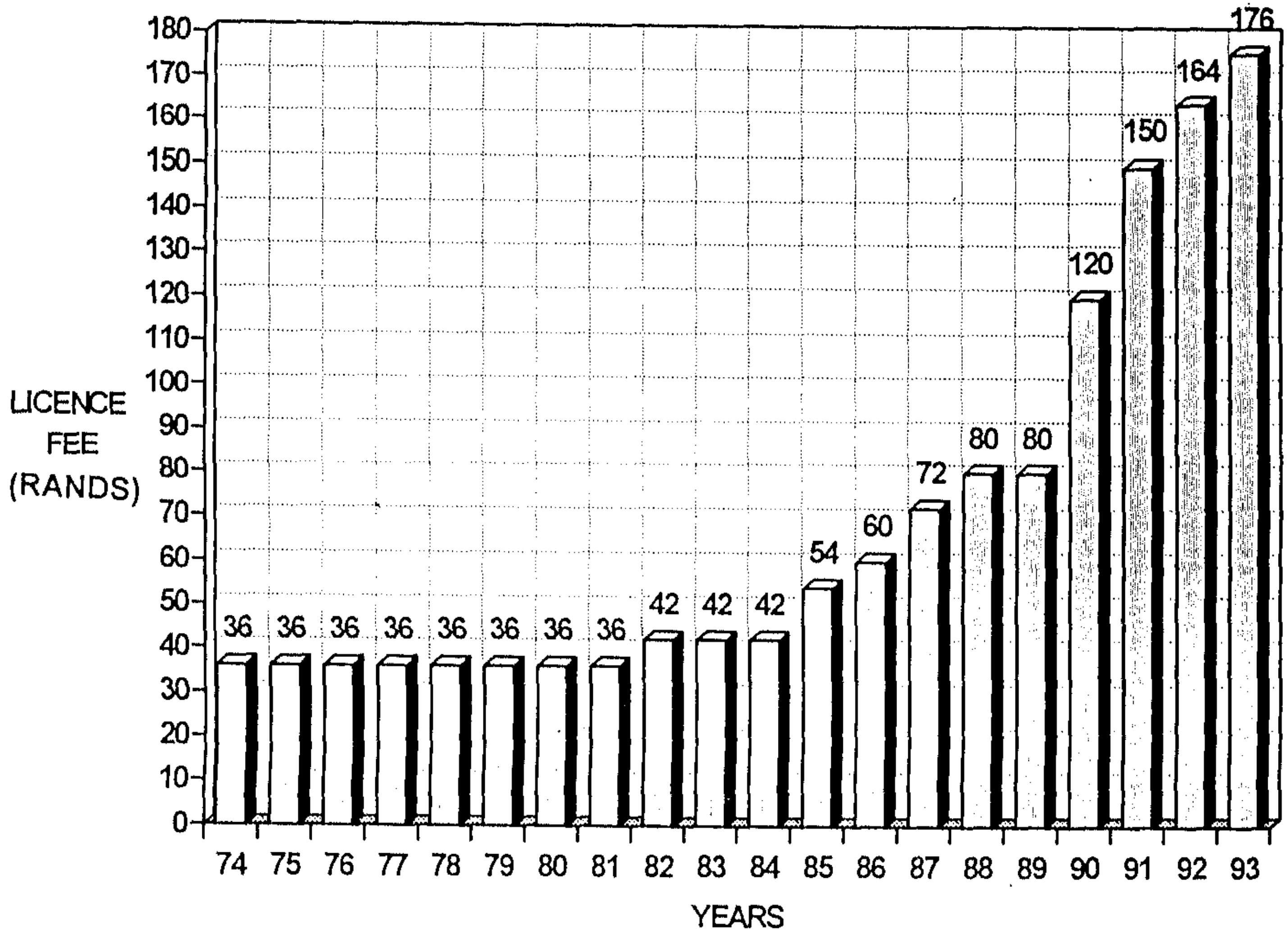

LICENCE
FEE
(RANDS)
0
10
20
30
40
50
60
70
80
90
100
110
120
130
140
150
160
170
180
36
36
36
36
36
36
36
36
42
42
42
54
60
72
80
80
120
150
164
176
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
YEARS

FIGURE 1.0

(ii) to be simple, fully functional, yet cost effective with minimum client participation.

These objectives were obtained by consulting various technical books and from in-depth discussions with knowledgeable engineers.

YEAR	LOSS R MILLIONS
92	100
87	30
86	27

Fig.1-1

Based upon the above systems that were investigated, with cost, size and also minimum device count being a major factor, it was decided that the design would be based upon the 8031/8051 architecture. This design would comprise of three circuit boards i.e.

(i) A 80C31 microcontroller based programming unit which would program the remote control. A data base would be incorporated into the programming unit. The p.u. would have the facility of downloading this data.

(ii) The Remote Control would contain the viewing data and security codes.

(iii) A 80C31 microcontroller supported by a Real Time Clock(RTC) would be installed and interfaced to the TV. The main function of this board would be to download the data from the remote control and to control the viewing times accordingly.

The database of the p.u. consists of subscribers who renewed their

TV licence. This data could be downloaded from the p.u., analysed and a statistical report could be drawn as to how many licences had been issued. From this report, the cost of a licence could be determined.
All software programming would use a low level computer language i.e. 8051 ASSEMBLER.

Thus, this thesis would provide:

(i) Background information of various systems investigated.

(ii) A detailed report on the proposed design/system.

The report commences with Chapter 2 by providing background information about the various systems investigated in the literature survey. This chapter would also contain a basic description of the proposed design/system. A detailed description of the complete design is given in Chapters 3, 4 and 5. Detailed analysis of software programming would also be given in Chapter 7. Chapter 10 would describe problems encountered with the design and solutions.

2. SYSTEMS INVESTIGATED

This chapter outlines the implementation, cost, advantages and disadvantages of the various systems investigated in the previous chapter. These were obtained through a literature survey. A more detailed explanation is also included of the proposed design i.e.. prevention of pirate tv/pay tv.

2.1 CABLE TV

2.11 Introduction

Cable television (CATV) started as a means of providing signals to communities that could not receive broadcast stations, either because of distance or shadow areas in which the signal was to weak. A community antenna was used at a remote location to feed TV signals to receivers in the area. Today, cable TV has developed far beyond that into huge systems that cover large areas, even for locations having good receptions. The reason is that cable TV does not have the restriction of channel allocations for broadcasting. The cable systems offer up to 36 channels. A cable converter box permits selection of the desired channel. Premium pay services such as Home Box Office, Spotlight, Prism, Cinemax, and others also offer current movies and sports events not available on broadcast television. These programs reach the cable operator via satellite transmission. Details of the cable channels,distribution systems, are described as follows:

2.12 CABLE FREQUENCIES

Many older cable systems distribute TV signals on the same VHF channel frequencies that are used for broadcasting. The UHF channels are converted to VHF channels for distribution because cable losses are too high in the UHF band. This method is a 12-channel system, including the lowband and the highband VHF channels 2 to 13. Direct cable connections are made to the TV receiver, where the RF tuner can be used to select the desired channel.

2.12.1 ADJACENT CABLE CHANNELS

With a 12-channel system, some receivers may have adjacent-channel interference. This interference produces a windshield wiper or venetian-blind effect in the picture. In the cable system , interference is minimized by balancing the signals for all channels at a common level.

2.12.2 MIDBAND AND SUPERBAND CABLE CHANNELS

Since the cable signal is not radiated, at least not intentionally, the cable system can use frequencies that are assigned to other radio services without interference. The midband cable channels are used in the gap between VHF channels 6 and 7. These frequencies, from 88 MHz to 174 MHz include 88 to 108 Mhz for the FM radio broadcast band plus various marine and aircraft communications services.

Table 2-0 lists the midband cable TV channels. Although not listed, the sound carrier frequency is 4.5 MHz higher.

TABLE 2.0
CABLE TV CHANNELS

LETTER DESIGNATION	NUMBER	VIDEO CARRIER, MHz	NUMBER	VIDEO CARRIER, MHz
Midband channels			Superband channels without letters	
A	14	121.25	40	319.25
B	15	127.25	41	325.25
C	16	133.25	42	331.25
D	17	139.25	43	337.25
E	18	145.25	44	343.25
F	19	151.25	45	349.25
G	20	157.25	46	355.25
H	21	163.25	47	361.25
I	22	169.25	48	367.25
			49	373.25
Superband channels			50	379.25
			51	385.25
J	23	217.25	52	391.25
K	24	223.25	53	397.25
L	25	229.25	Additional midband assignments	
M	26	235.25		
N	27	241.25		
O	28	247.25	54	89.25
P	29	253.25	55	95.25
Q	30	259.25	56	101.25
R	31	265.25	57	107.25
S	32	271.25	58	97.25
T	33	277.25	59	103.25
U	34	283.25	Nominal channel numbers for use with digital *readout converters*	
V	35	289.25		
W	36	295.25		
X	37	301.25		
Y	38	307.25	A-2 or 00	109.25
Z	39	313.25	A-1 or 01	115.25

Double digits are used for all cable channel numbers to allow for a digital control board for tuning.

SUPERBAND means cable TV channels above the VHF broadcast channel 13. The use of VHF broadcast channels provides 36 channels in a typical large cable TV system. These frequencies are up to approximately 300 MHz.

2.12.3 TUNING TO THE CABLE CHANNELS

In conventional TV receivers, the RF tuner usually is not made to select the midband and the superband cable channels. The cable operator provides a separate converter unit to convert all cable frequencies to a designated VHF channel, such as channel 2, 3, or 4. The subscriber keeps the receiver tuned to the specified channel, and all channel selection is done at the converter.

2.12.4 CABLE-READY TV RECEIVERS

Late model receivers offer a tuner that can select the midband and superband cable channels directly without the need for a converter. However , there is another practical problem. The premium pay services have a signal that is scrambled electronically. The system-oriented converter would be needed to watch the scrambled-signal premium channels.

2.12.5 CABLE RADIATION

Radiation occurs if cables are open, short-circuited, or partially mismatched at their termination.

To detect radiation, a selected midband channel may be used for an FM tone modulated indicator signal. A portable FM radio is used as a "sniffer" to locate any radiation, by riding along the cable route.

3.0 COAXIAL CABLE FOR CATV

The type of cable generally used in a main signal route, is called a TRUNK LINE, as shown in Fig. 2-1. It consists of a heavy central aluminium conductor that is copper-clad. The outer conductor or shield is also aluminium and is shaped in a solid tube. A polyethylene foam fills the internal space and supports the inner conductor exactly at the centre. The cable diameter is about 19.1mm.

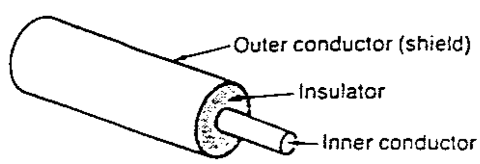


Fig. 2-1 Coaxial cable consists of inner conductor at center of outer tubular conductor.

Type of coaxial cable used for trunk lines.

The line from a branch to the subscriber is called a DROP LINE. The drop line is generally RG-59U coaxial cable. Its diameter is 6.35 mm.

4.0 CHARACTERISTIC IMPEDANCE

Coaxial cable is a type of transmission line, which means a line with uniform distance between the two conductors, separated by a dielectric material.

The theory of operation can be explained as follows:

A coaxial transmission line consists of concentric centre and outer conductors that are separated by a dielectric material. When current flows along the centre conductor, it establishes an electric field. The electric flux density and the electric field intensity are determined by the dielectric constant of the dielectric material. The dielectric material becomes polarized with positive charges on one side and negative charges on the opposite side. The dielectric acts as a capacitor with a given capacitance per unit length of line. Properties of the field also establish a given inductance per unit length , and a given series resistance per unit length. If the transmission line resistance is negligible and the line is terminated properly, the following formula describes the characteristic impedance (Zo) of the cable.

$$Zo = \sqrt{L/C}$$

Where:

L = inductance in H/ft

C = capacitance in F/ft

The coaxial cable used for CATV has a characteristic impedance of 72 to 75Ω. Generally 75Ω is accepted as the nominal value.

The Zo is resistive, without a reactive component, but it is an ac value that cannot be measured with the ohmmeter.

For example, assume for a 1-ft length that L = 0.12μH and C = 21pF, then

$$Zo = \sqrt{0.12x10-6 /21x10-12}$$

$$= \sqrt{0.0057x10-6}$$

$$= 0.076x10-6$$

$$= 76\Omega$$

Zo stated in terms of L and C are determined by the physical

characteristics of conductor size, spacing and the dielectric.

For coaxial line with an air dielectric between the two conductors,

$$Zo = 138 \log D/d \quad \Omega$$

where

d is the diameter of the inside conductor

D is diameter of the outside conductor, which really indicates the spacing between the conductors.

For example, consider No 18 gauge wire with d = 0.04in. for the inner conductor. Let D = 0.25in. for the outer conductor for 1/4in. RG-59U cable. Then,

$$Zo = 138 \log 0.25/0.04$$

$$= 138 \log 6.25$$

$$= 138(0.796)$$

$$= 110 \ \Omega$$

The 110 Ω value is for an air dielectric. Use of a foam or plastic insulator between the conductors reduces Zo by about 66 percent, to a value of 110 x 0.66 = 72.6 Ω.

The 0.66 is the velocity factor of the line which is given by:

$$Vp = \sqrt{1/LxC}.$$

Vp, is the amount the signal is slowed and is represented as a percentage of the free space velocity.

4.1 MISMATCH WITH AN OPEN OR SHORT-CIRCUITED LINE

When the line is not terminated in Zo an impedance mismatch results. One extreme case is a line open at the end, as shown in figure 2-2a. The driving source here is an ac generator to supply signal on the line. C3 has a signal voltage of the same polarity as the source and acts as a source. C3 sends back a wave of discharge current, in the opposite direction from the incident charging current. These effects are illustrated by the standing-wave patterns of V and I in Fig. 2-2b.

In the line with the open end, standing waves of V and I are set up along the line. A maximum on the standing wave occurs where the incident and the reflected values have the same phase, with respect to distance along the line. A node occurs where the incident and reflected values are out of phase and so cancel. With respect to time, the ac values of V and I are continuously changing.

For a specific frequency of ac signal on the line, the distances can be considered in terms of the wavelength. One-quarter wave back from the open end, the voltage waves are 180deg. out of phase, and a voltage null results. The current is additive here, which makes I double the value of a matched line. At this point on line, it is equivalent to a series resonant circuit. The reason is

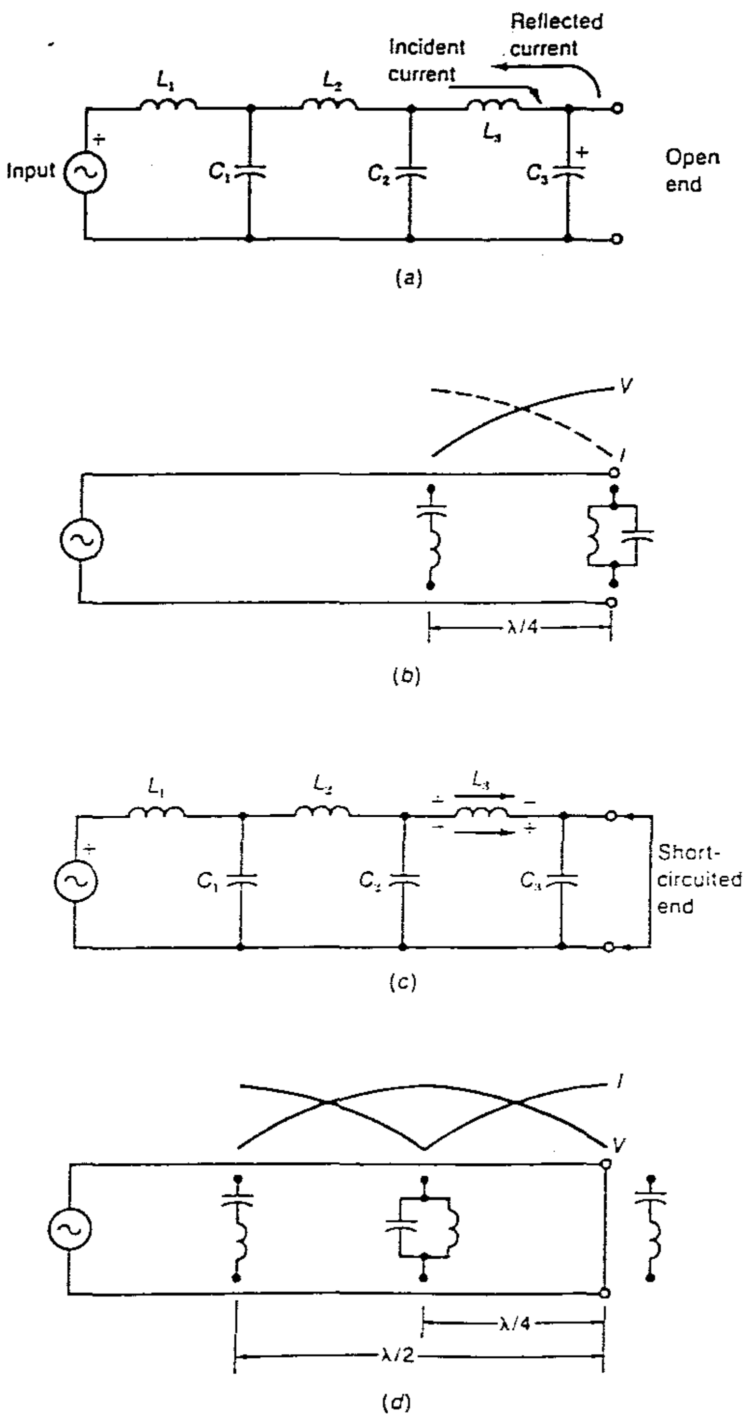


Fig. 2-2 Effects of an open circuit and short circuit at end of transmission line. (*a*) Model line terminated in an open circuit. (*b*) Effect a quarter-wave back from the open end. (*c*) Model line terminated in a short circuit. (*d*) Effect a quarter-wave back from short-circuited end.

that I is high and V is low for a specific frequency. At the end of the open line, the current is nulled and the voltage is doubled. The open end corresponds to a parallel resonant circuit, with a high Z at a specific frequency. The relative values of V and I reverse at quarter-wave distances back to the source. At each peak or null on the standing wave, the V/I ratio for Z is resistive and has a maximum or minimum value. At intermediate points, the line is reactive and has intermediate values of impedance.

The conclusion is that an open coaxial cable acts as a tuned circuit. All the energy supplied to a lossless line would be reflected to the source at the input. When long run lines are used, the resulting time delay in the reflected signals can produce ghosts in the TV picture. If the driving source does not have a 75Ω impedance, then the signal reflected from the open end is also reflected again from the source, which creates multiple ghosts.

The example of a line short-circuited at the end is shown in Fig.2-2c. After a current peak has been produced by the driving source, the inductance tends to keep the current flowing. Energy is reflected from the end of the line in the same direction as the incident current. The polarity of the self-induced voltage is reversed. The resulting standing waves of V and I are shown in Fig.2-2d. They respond to the standing waves in Fig. 2-2b for the open line, but the V and I patterns are reversed in terms of peaks and nulls.

4.2 VOLTAGE STANDING-WAVE RATIO

Reflection occurs with a poor termination not equal to the characteristic impedance Zo. As a result, the voltage is less than double at the peaks in the standing-wave pattern, and the nulls are more than zero. Thus the degree of cable match can be expressed as the ratio of the voltage at the peaks to the voltage at the nulls. The proportion is the voltage standing-wave ratio (VSWR).

$$VSWR = V_{max}/V_{min}$$

When the line is perfectly matched with a termination equal to its own Zo, there is no standing waves, then VSWR = 1.

5.0 CABLE LOSSES

When a signal is applied to the line some energy is dissipated in the line itself. The result is attenuation of the signal. There are three causes of attenuation:

(1) I^2R losses produced by current in the conductors.

(2) Dielectric losses in the insulator between conductors.

(3) Skin effect. The RF current flows more on the circumference of the conductor than its centre. Because of the smaller area for current, the ac resistance of the conductor increases. The aluminium cable in Fig. 2-1 has a copper coating around the inside conductor to reduce losses from the skin effect.

5.1 LOSSES INCREASE WITH FREQUENCY AND CABLE DISTANCES

The losses increase in proportion to the square root of frequency f. For a practical case, refer to Table 2-1, compare channel 13 at 210 to 216 MHz which is about four times higher in frequency than channel 2 at 54 to 60 MHz. At 4f, the line losses for channel 13 equal $\sqrt{4}$, or double the losses for channel 2.

With respect to the signal, the attenuation is measured by per unit distance of the line i.e.

6dB loss per 183m.

A 6dB loss in voltage means one-half the power of the signal.

CHANNEL NUMBER	VIDEO CARRIER, MHz	CHANNEL NUMBER	VIDEO CARRIER, MHz
00	108.00	30	258.00
01	114.00	31	264.00
02	54.00	32	270.00
03	60.00	33	276.00
04	66.00	34	282.00
05	78.00	35	288.00
06	84.00	36	294.00
07	174.00	37	300.00
08	180.00	38	306.00
09	186.00	39	312.00
10	192.00	40	318.00
11	198.00	41	324.00
12	204.00	42	330.00
13	210.00	43	336.00
14	120.00	44	342.00
15	126.00	45	348.00
16	132.00	46	354.00
17	138.00	47	360.00
18	144.00	48	366.00
19	150.00	49	372.00
20	156.00	50	378.00
21	162.00	51	384.00
22	168.00	52	390.00
23	216.00	53	396.00
24	222.00	54	72
25	228.00	55	90
26	234.00	56	96
27	240.00	57	102
28	246.00	58	402
29	252.00	59	408

TABLE 2-1

6.0 CABLE DISTRIBUTION SYSTEM

Refer to Fig. 2-3. The starting point for cable signals is called the Head End. Here the broadcast signals picked up by the antenna are amplified, adjusted for level, and fed into the trunk lines. The UHF channels are converted to VHF channels. Also included are local-origination signals from a studio. The video and audio signals modulate separate carriers in a VHF channel not being used. The main routes of signal from the head end are the trunk lines.

6.1 TRUNK AMPLIFIERS

The trunk amplifiers are inserted at regular intervals along the trunk route to make up for cable losses and keep the signal up to the standard level of 1 to 3 mV.

Figure 2-4 shows a weatherproof housing for a trunk amplifier in an aerial system, which is pole mounted. Power is obtained from a tap on electric service lines on the same pole.

6.2 BRIDGING AMPLIFIERS

This type of amplifier is for a branch from the main trunk to feed a particular neighbourhood in the cable system. The typical gain is 20 to 40 dB. The output is for the branch lines to individual subscribers.

An attenuator may be used at the input to the bridging amplifier to balance the signal levels.

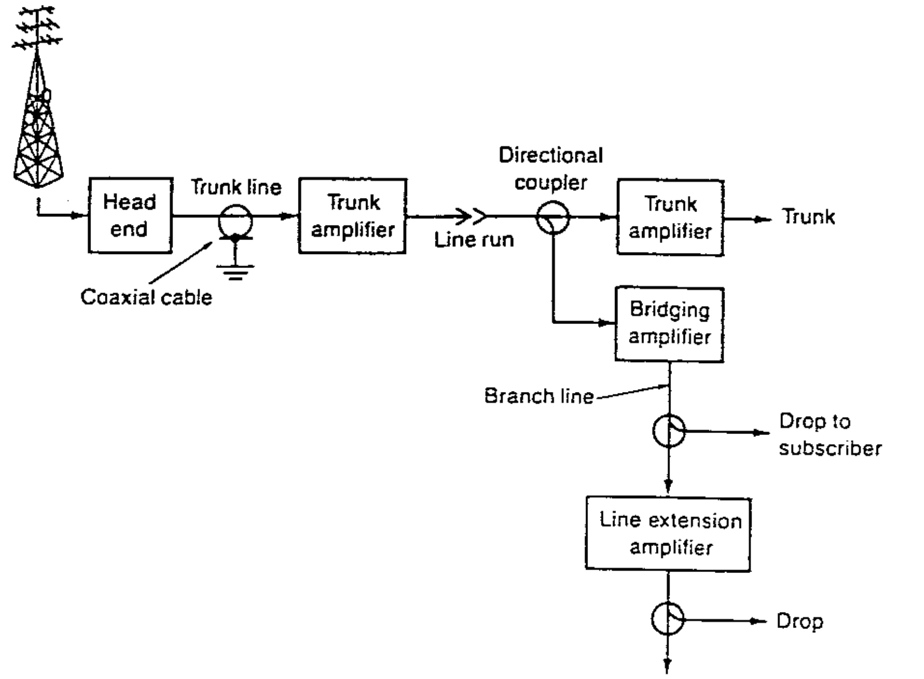


Fig. 2-3 Basic distribution system for cable television. Note symbol for directional coupler.

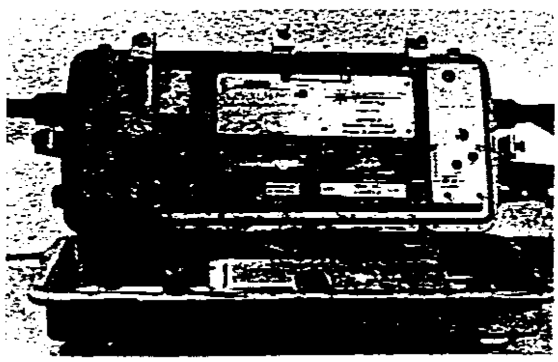


Fig. 2-4 Typical trunk amplifier unit, for outdoor pole mounting. Front cover removed for internal view. Width is 14 in. [355.6 mm].

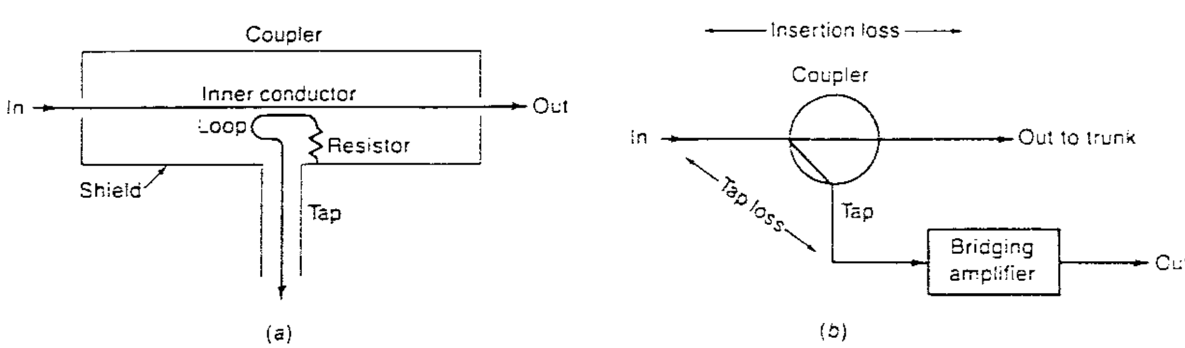


Fig. 2-5 Directional coupler tapping off signal from trunk line. (*a*) Schematic diagram. (*b*) Symbol.

6.3 LINE AMPLIFIERS

Long line runs from the bridging amplifier may require that line-extender amplifiers be inserted in the branch line to make up for the cable losses in that branch. This amplifier extends the number of drop lines that can be used on a branch line. The typical gain for a line amplifier is 20 to 40 dB.

6.4 DIRECTIONAL COUPLERS

Signal power taken from the trunk must be kept very small so that the line is not loaded by all the branches. Its construction and symbol is illustrated in Figs.2-5a and 5b.

It is a three-terminal device. One terminal is for signal input, another carries the signal through the trunk line and the third terminal has tapped output signal for a branch.

The directional coupler is so named because it feeds a sample of the direct, downstream signal out at the tap but ignores reflected energy in the trunk line. This is accomplished by a 75Ω loop placed in the wall of the coaxial assembly, as shown in fig.2-5a.

The loop acts as both a capacitor and an inductor. Its capacitance charges to the potential difference between the inner and outer conductors at that point on the line. As a one-turn coil, the loop is magnetically coupled to the centre conductor to tap off the signal.

Directional couplers have a very small insertion loss, 1dB at 300 MHz, between the input and the output signals on the trunk line. The tap loss from input to output at the tap is 13dB, but this loss is made up in the bridging amplifier.

6.5 POWER SUPPLIES

Input for the power supply is 120 V ac tapped from the power line on the same pole in an aerial system. The dc supply voltage, 24V, powers the cable amplifiers and has a back up battery supply. The power supply module often is located in the same weatherproof housing that encloses the trunk and bridging amplifiers.

6.6 LINE TAPS

The final tap on the system feeds the drop line for the subscriber, usually with RG59U coaxial cable. Fig.2-6 shows a multitap with six taps for six houses. The line tap has a low insertion loss but a high tap loss.The tap-to-tap loss is made high to provide isolation between the individual subscriber lines. Isolation is necessary so that a misterminated cable at the subscriber's TV receiver will not set up reflections in the cable system. Two possible terminations are: a cable that is not connected or leads that are short circuited. The tap units are available with various values of tap loss, so the signal levels can be balanced for different subscriber drop lines along the branch.

6.7 BALUN UNITS

A typical unit is shown in Fig.2-7a. The balun is used to match the 75Ω coaxial cable to the 300Ω TV receiver input. The balun can match the impedances in either direction.

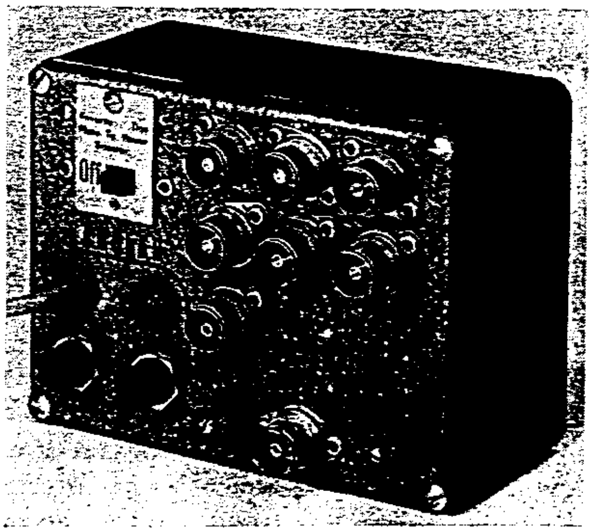
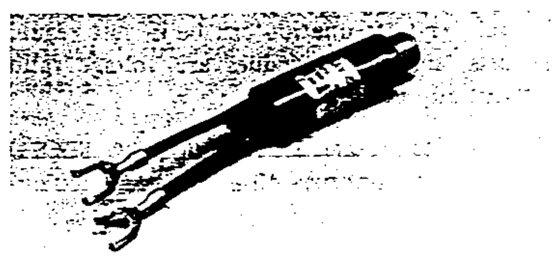
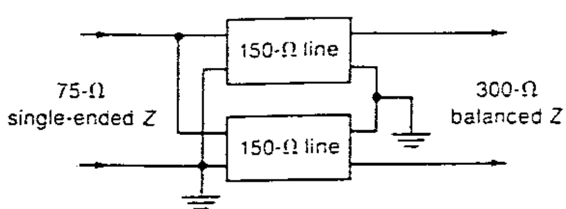


Fig. 2-6 *A typical u.h.f. distribution amplifier with take-offs for six points (Lagbear Ltd).*



(a)



(b)

Fig. 2-7 The balun matches 75-Ω single-ended Z of cable to 300-Ω balanced Z of antenna input circuit of receiver. (a) Typical unit. Size is 2 × ½ in. [50.8 × 12.7 mm] without leads. (b) Schematic diagram.

6.8 F CONNECTOR

The coaxial connector for the 75Ω line in FIG.2-7a is the standard F connector used in cable work. Its advantage is that no soldering is required. However, most receivers are manufactured with this connector, so the coaxial cable may be directly plugged into the receiver.

7.0 DISTORTION IN THE CABLE SIGNAL

(1) HARMONIC DISTORTION

When the amplifier is overdriven, harmonic distortion occurs as a result of amplitude distortion. These changes in the waveform generate new frequencies. Second-order harmonics are not serious in a 12-channel system because the second harmonic of channels 2 to 6 falls in the range of 110 to 176 MHz which is in the midband range between channels 6 and 7. These harmonics cause interference problems, in cable systems using midband and superband channels. The cable amplifiers use push-pull operation. Second harmonics generated in the push-pull circuit are cancelled in the output.

(2) CROSS MODULATION

Third-order harmonics and the effects of cross modulation are serious results of overload distortion. The nonlinear amplifier characteristic causes an overloaded amplifier stage to operate as a mixer stage, the input signals are detected, and the modulation is extracted. A transfer of the modulation to another carrier frequency can then occur.

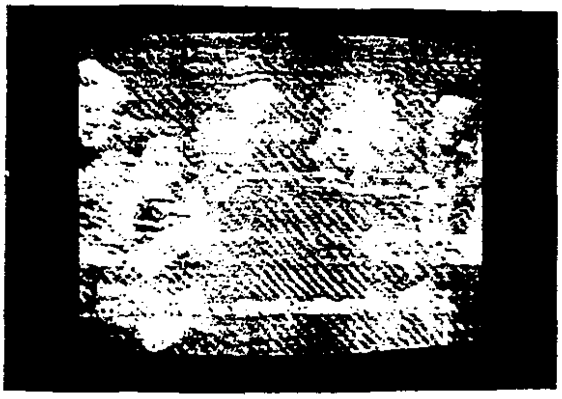


Fig. 2-8 Symptoms of cross-modulation distortion in cable amplifier.

The effect of cross modulation on the picture is a jumble of lines and the appearance of video from another channel.
The easiest symptom of a cross modulation to recognise, is the maximum modulation of the interfering signal, which is synchronization. This appears as vertical bars in the picture as shown in Fig.2-8.

(3) SPURIOUS SIGNALS

Another effect of amplifier overload is the production of spurious frequencies that are not harmonically related to the desired signal. Harmonics of all the signal frequencies are produced, caused by the mixer action with nonlinear operation. These frequencies can beat with each other, resulting in a wide range of unwanted signal frequencies.

(4) TEMPERATURE EFFECTS

The signal level changes at an approximate rate of 1% per 10deg.F due to the effect of temperature on the cables. In very long cable runs the variation in signal level could be 100 dB. The double trouble is a reduction in signal-to-noise ratio in hot weather and the probability of amplifier overload when it snows.

7.1 AUTOMATIC GAIN CONTROL

AGC in the trunk amplifiers compensate for temperature changes and other variable factors. A typical arrangement is shown in Fig.2-9. The system uses pilot carrier inserted at a frequency selected just for the AGC system. A pilot frequency of 73 MHz is indicated here. The narrowband amplifier is tuned to the pilot frequency. The AGC detector produces a dc control voltage proportional to the amplitude of the pilot signal. The dc control voltage varies the attenuation of the cable signal into the preamplifier. A manual control is also provided by R1 for the initial adjustment.

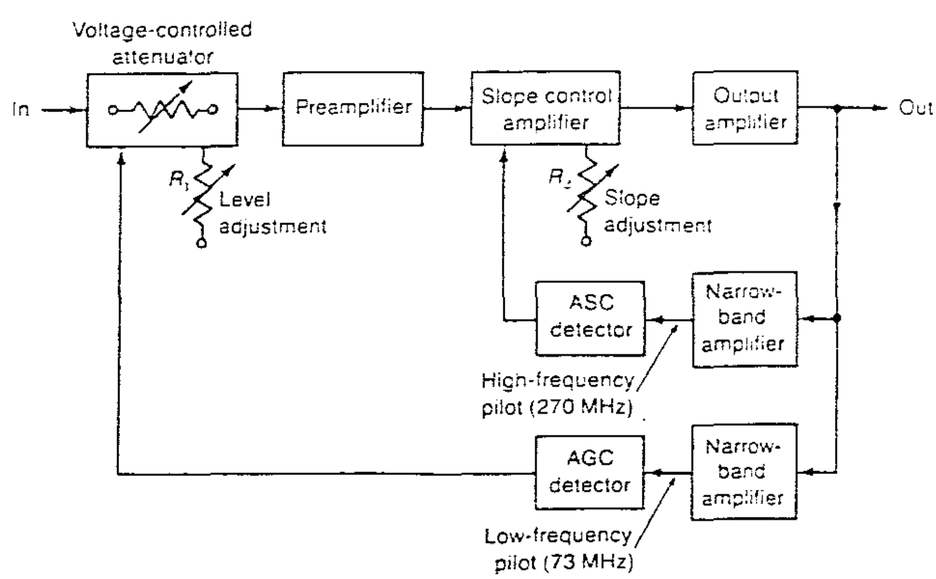


Fig. 2-9 Block diagram of trunk amplifier. ASC is automatic slope control.

7.2 AUTOMATIC SLOPE CONTROL

The cable attenuation increases for higher frequencies. Therefore, the overall frequency response of the cable tilts down for higher-frequency channels. To compensate for this effect, the frequency response of the amplifier is made to slope upward for the higher channel frequencies. the desired result is a uniform or flat frequency response for signals in all the channels. The automatic slope control (ASC) is an AGC system designed to increase the amplifier gain just for higher channel frequencies. In Fig 2-9 the ASC pilot frequency is 270 MHz. The narrowband amplifier and ASC detector are tuned to this frequency to provide a dc control voltage that varies the gain of slope-control amplifier. The R2 control for ASC and R1 for AGC are set to balance the system to achieve a uniform signal on all channels. Automatic control circuits then maintain the balance.

8.0 TWO-WAY CABLE SYSTEMS

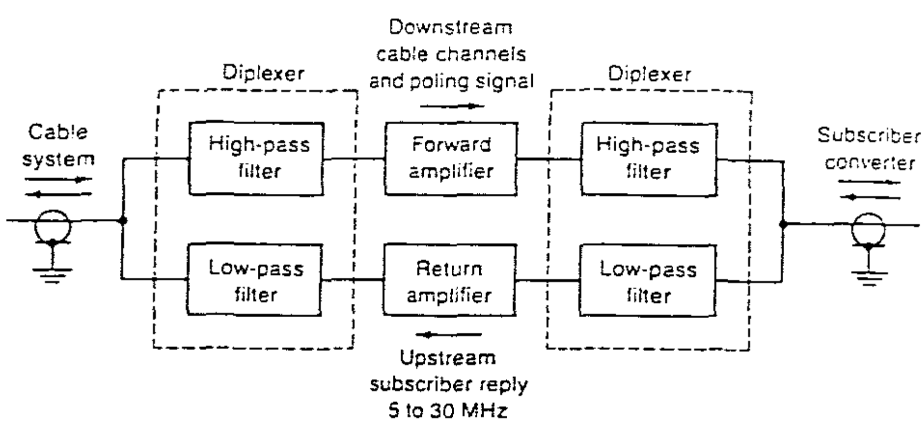


Fig. 2-10 Two-way cable amplifier system.

The cable systems are also designed for upstream service from subscribers to the head end. Separate amplifiers are need for the

upstream signal, as shown in Fig. 2-10. Upstream communications is in the band of frequencies between 5 and 30 MHz.
The converter unit at the subscriber end supplies the upstream signal.
The system provides two-way communications with subscribers for billing and pay per view for special programs.

8.1 POLING SIGNAL

The pay-per-view system requires that a poling signal be sent downstream to communicate with only the appropriate converter units. The frequency of the poling signal is in the 107 to 119MHz range. Frequency-shift keying (FSK) is used with a 16bit word as modulation for the particular address. When poled, the converter replies with data set up at the keypad. For example, punching in 55 can indicate that the subscriber wants to view a special program that evening. The reply to the poling signal is sent back to the head end by means of a modulator operating in the 5 - 30MHz range for the upstream signal.
An affirmative reply to the head end sends a coded signal back downstream to operate the descrambler in the subscriber's converter unit, on the correct channel at the specified time.
Also data is entered into the cable operator's computer for billing purposes.
A digital address for each subscriber is set in ROM in each converter unit upon installation. If the convertor is stolen, the electronic address will show up at the wrong location.
The main problem with two-way cable systems is interference from

the upstream signal in the 5 to 30 MHz range. The entire cable system acts as an antenna for the return signal. Any part of the system that radiate signals also pick up interfering frequencies. Digital switching methods are used to isolate any branches that pick up interference. A particular problem is that the subscriber's modulator may be stuck in the transmit mode, which can shut down the entire poling system. Protection Switch provisions are made to switch to another poling frequency when this occurs.

9.0 CABLE TV CONVERTERS

The converter on top of the TV set is an RF tuner used to select the desired channels. All the cable channels are heterodyned to a specific frequency band chosen to be one of the lowband VHF channels. Usually, channel 3 or 4 is the choice, depending on which is not an active broadcast channel in the area. A simple frequency conversion to channel 3 or 4 presents some problems. For channel 3 conversion, the local oscillator (LO) in the converter would operate 61.25MHz above the selected channel. 61.25MHz is the picture carrier frequency for channel 3. Assume that channel 4 is selected by the converter. Then the LO operates at: 61.25+67.25 = 128.5MHz. The LO frequency is inside the cable midband for channel B or 15 at 126 to 132 MHz. There are several other frequency combinations in which the LO signal can be a source of interference, since the broadband preselector in the converter is not very selective for RF tuning owing to the wide range of cable channels.

The solution to this problem is to use a double-superheterodyne circuit for the cable convertor, as shown in Fig 2-11.

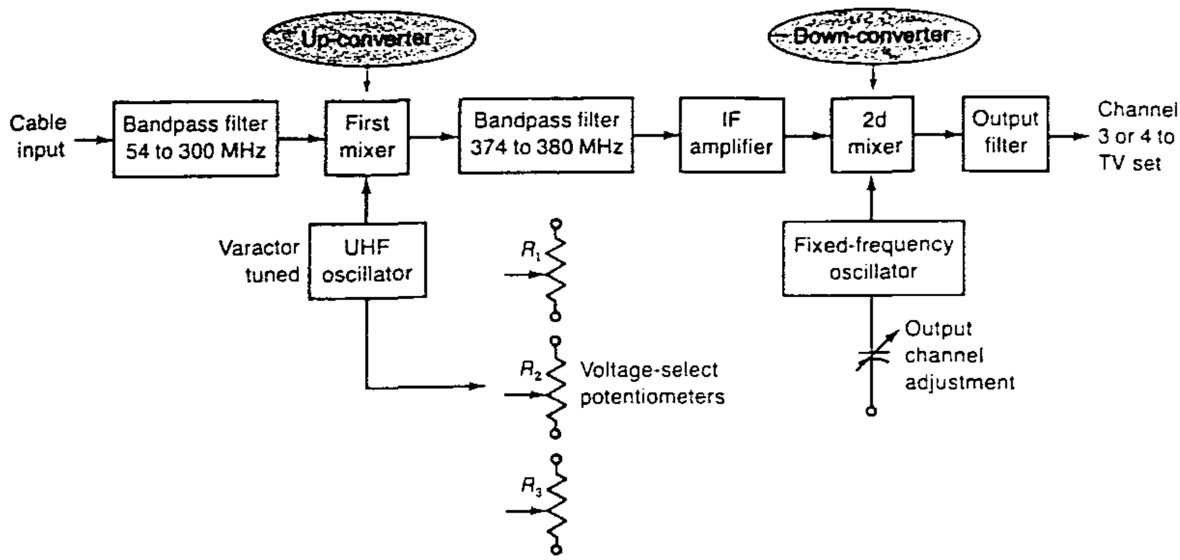


Fig. 2-11 Basic block diagram of up-down cable TV converter.

9.1 UP-DOWN CONVERTERS

Refer to the block diagram in Fig. 2-11. In this circuit, the cable channels are heterodyned up to the IF value of 374 to 380 MHz in the UHF band.

The IF signal is converted down to the frequency for either channel 3 or 4. Two mixer stages are used. The first mixer with the varactor-tuned local oscillator converts all incoming channels to IF of the convertor. The IF band is 374 to 380 MHz, but some convertors use 608 to 614 MHz for the IF signal. A bandpass filter in the output circuit of the up convertor selects only the IF signal, which is the desired channel tuned in by the UHF local oscillator. The second mixer, with a fixed frequency oscillator, is the down converter. It heterodynes the IF signal at its UHF values down to either channel 3 or 4 for the TV receiver. An adjustment in this oscillator is set for either output channel. The frequency that is used in the second LO for down conversion stays for all channels. The IF signal is always in the fixed IF passband of the converter. Each of the selected channels has been selected by tuning the first LO frequency in the up converted, which heterodynes all the cable channels up to the IF band.

9.2 CHANNEL SELECTION

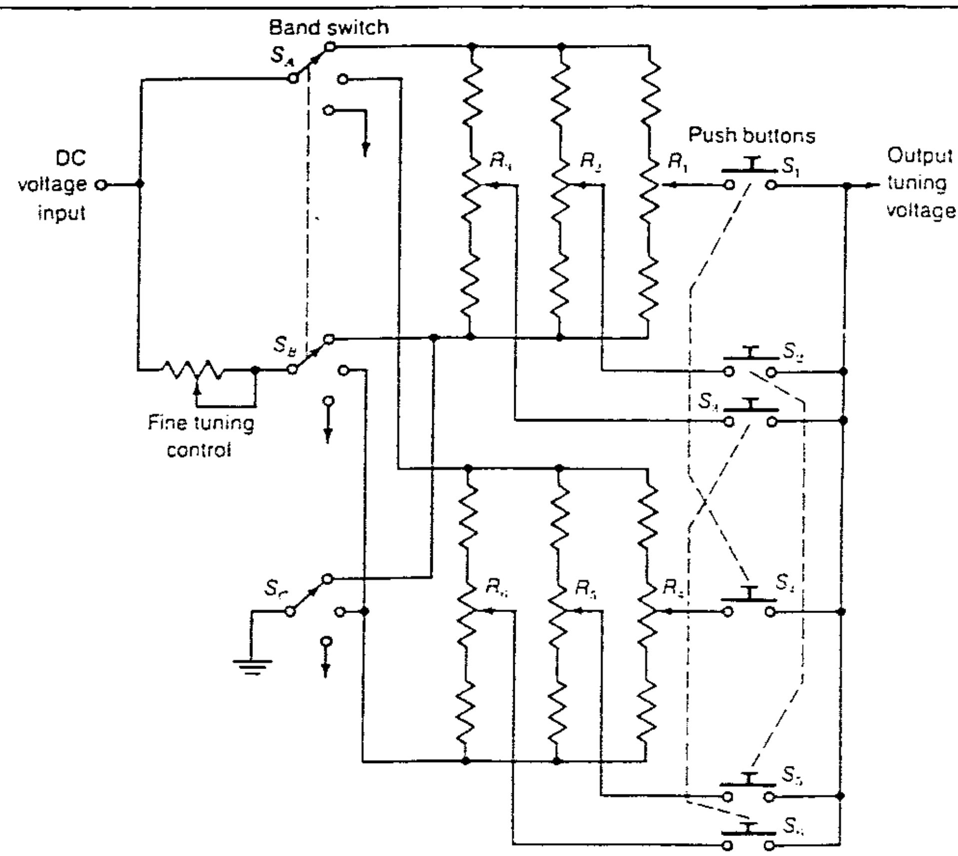


Fig. 2-12 Voltage-control for varactor tuning with variable-R potentiometers. System uses 36 potentiometers, 12 pushbuttons, and three-position band selector for 36 channels. Only six channels are shown here.

Each channel is tuned by varactor control of the local oscillator frequency in the up convertor. A typical arrangement for the cable convertor is shown in Fig. 2-12. Only six pushbutton switches are shown for simplicity. The pushbutton switches S1 to S6, are identified with the channel numbers. When they are pushed in, the switches insert the variable resistors. R1 to R6 into the circuit. The dc bias voltage is supplied through the three-position band switch with sections SA, SC. Each potentiometer is preset to provide the dc voltage that makes the VCO operate at the frequency needed to tune in a specific channel. The three-position band switch, with 12 pushbutton switches and 12 potentiometers, allows tuning for 36 channels.

9.3 FREQUENCY SYNTHESIZER

Fig. 2-13 shows the complete block diagram for an up-down converter using frequency synthesis. The synthesizer section illustrates how the frequency is set for the VCO as the first local oscillator for the up converter. An 8MHz crystal-controlled oscillator is the reference.

The 8Mhz oscillator in the synthesizer is divided by 1024 to produce a 7.8125kHz signal into the phase detector or comparator. The other input to the comparator is a sample of the signal from the VCO in the up converter. This sample is divided first by 256 and then by a factor N that is set in the programmable counter. The idea is to divide by whatever N is needed to provide 7.8125 kHz for comparison with the reference signal in the phase detector.

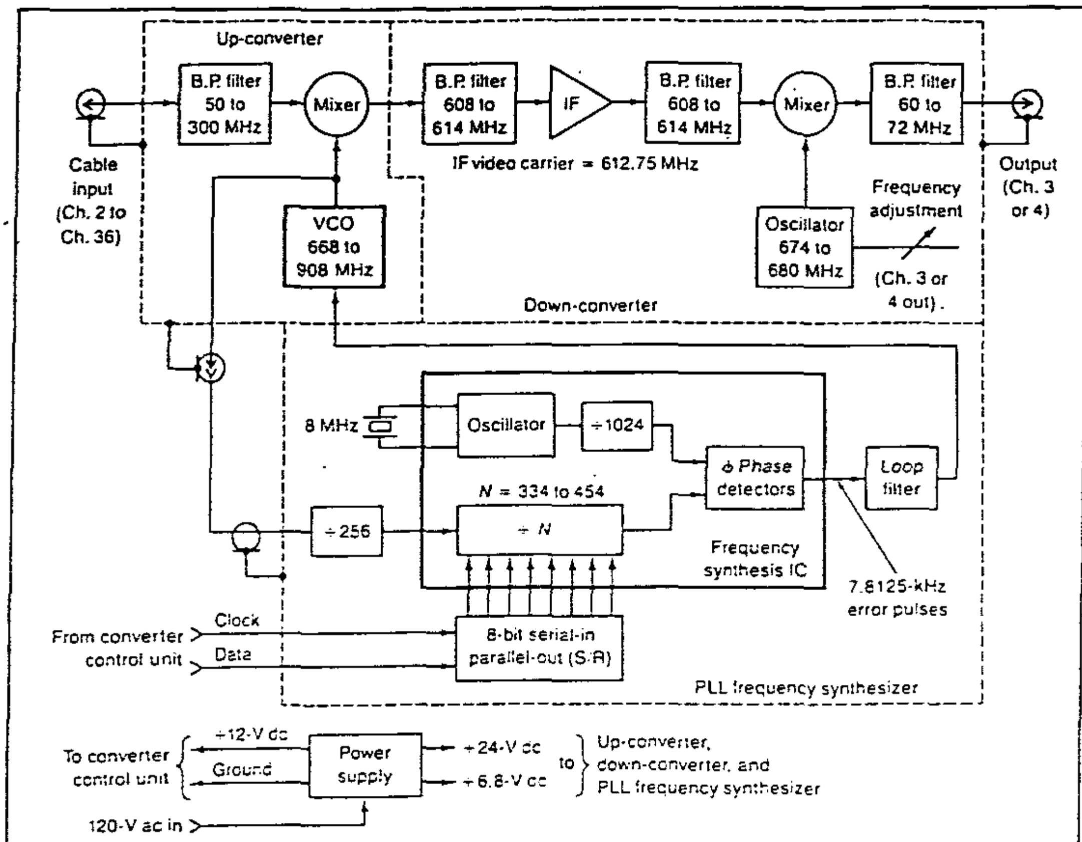


Fig. 2-13 Cable converter unit using frequency synthesis for first local oscillator in up converter. B.P. indicates bandpass filters used to remove spurious signals.

The factor N can take any valuable between 334 and 454. The actual count is determined by an 8bit binary code set with the pushbutton switches, on the counter or remote unit. For example, Superband channel 36 has a video carrier frequency of 295.25 MHz.

The video IF carrier frequency is 612.75 MHz for the convertor in Fig. 2-13. The required VCO frequency is

$$295.25 + 612.75 = 908 \text{ MHz}$$

The 908MHz oscillator output is supplied to the first mixer, but a sample is also taken for the synthesizer IC unit. The 908MHz LO frequency is divided by 256, which gives 3,546875 MHz. Then the programmable counter is set to divide by 454:

$$3546.875 \text{ kHz}/454 \quad = 7.8125 \text{ kHz}$$

The divided VCO signal and reference oscillator signal are compared in the phase detector. Its output is the dc control voltage to correct the VCO frequency. The feed to the phase lock loop (PLL) VCO is through a filter to eliminate noise and it is used for frequency control. If there is any error in frequency, the control voltage corrects the VCO to make its divided value at 7.8125 kHz exactly the same as the reference from the crystal oscillator. As a result, the VCO in the up converter has the same accuracy as the standard crystal oscillator for any selected channel.

The cable converter unit has another frequency inversion in the down converter. As a result of the double inversion, in the output signal on channel 3 or 4 the picture and sound carriers are at the same frequencies as in a regular TV broadcast channel.

10.0 SCRAMBLING/DESCRAMBLING METHODS

Cable systems offer for a minimum fee the so-called basic service, plus the premuim services. These premuim services feature special sports events and movies, uncut and without commercial interruptions. These premuim channels require a fee to be paid that is added to the basic charge.To serve only those subscribers who pay for the extra service, a scrambling technique is used.

10.1 SCRAMBLING

The most common method of scrambling the signal is SYNC SUPPRESSION. Sync is compressed only in the RF modulation envelope of the video carrier in the cable channel.The receiver cannot lock in with the sync-suppressed signal. The picture is out of sync, both vertically and horizontally. The loss of sync interferes with the receiver AGC overload distortion and results in the picture being dark, possibly reversed in white and black values, like a negative, and out of sync.

10.2 DESCRAMBLING

The descrambler unit reverses the effect of the scrambler at the head end by restoring sync pulses to the RF signal. Sync is restored by means of a keyed RF attenuator bypassed with a diode switch, indicated as RA and S in Fig. 2-14.

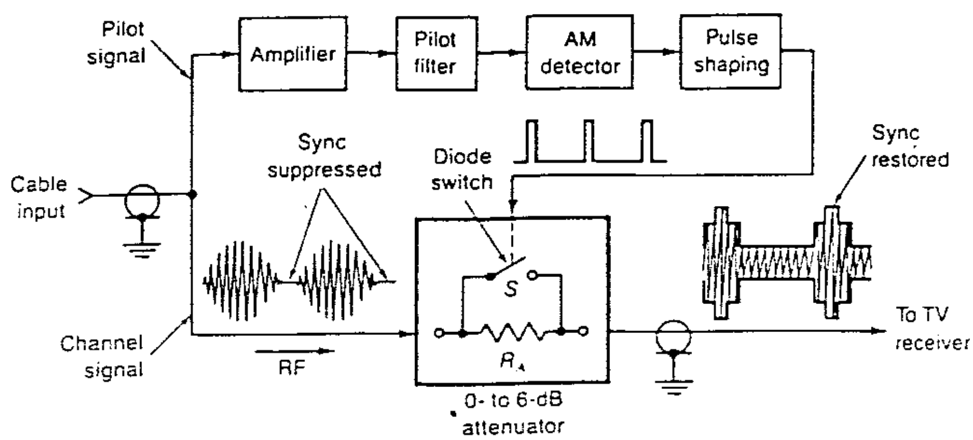


Fig. 2-14 Block diagram of basic descrambler.

In this method, the pulses needed for the switched attenuator are sent to the descrambler unit by a separate route. A pilot carrier signal having a frequency below that of the channel is used. An example of the pilot carrier frequency is 114 MHz for midband channel A at 120 to 126 MHz. Cable operators choose their own pilot frequencies for security reasons. In Fig.2-14, the descrambler contains a narrowband receiver tuned to the assigned pilot frequency. The receiver has an amplitude detector and pulse-shaping circuits to drive the diode switch. The decoding pulse in the pilot signal is the sync needed for descrambling. As a result, sync is stored in the RF signal for TV receiver.

10.21 IN-BAND DESCRAMBLERS

In-band descramblers uses the same idea of a pilot signal for descrambling, but the decoding pulses are sent inside the passband of the scrambled channel. The descrambler units are used with cable converter, connected as shown in Fig. 2-15.

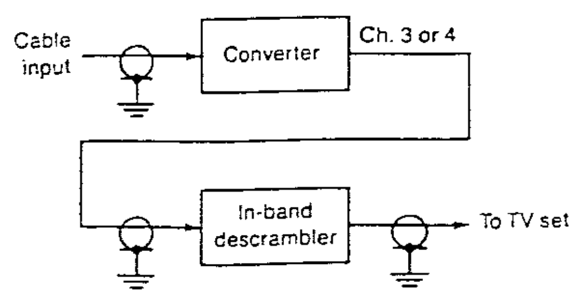


Fig. 2-15 Connections for in-band descrambler.

For an in-band system, the FM sound carrier signal in a scrambled channel is amplitude-modulated with the decoding pulses.
Since the converter changes all cable channels to a single channel such as 3 or 4, the descrambler unit in Fig. 15-23 contains a narrowband receiver tuned just for the sound carrier frequency of the designated channel in the converter output. The descrambler operates just as the pilot carrier system illustrated in Fig. 2-14.
The advantage of the in-band method is that it allows as many scrambled channels as the cable operator wants to offer.

11.0 LONG-DISTANCE LINKS

Large cable systems often cover long distances which could result in prohibitive cable attenuation. The methods used by the cable operator to reduce losses for long-distance links include super-trunks, microwave links, and fibre-optic links.
Larger cables are used and the cable channels are heterodyned down to lower frequencies. Both techniques reduce the cable losses..

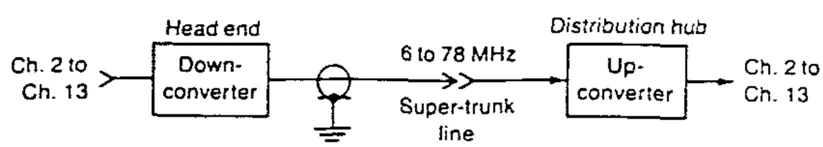


Fig. 2-16

As shown in Figure 2-16, the cable channels are changed to 6 to 78 MHz in the down converter.This band has space for 12 channels. Cable losses in the supertrunk are reduced for the lower frequencies, in proportion to the square root of the frequency change. A second heterodyne circuit is needed as an up converter in Fig 2-16, to provide the cable channels at their standard frequencies. From the hub, channels 2 to 12 are fed to the cable distribution system. In the supertrunk line, special low-loss cable is used. For example, a 25.4mm coaxial cable with fused insulator disks has an attenuation of only 0.32 dB per 30m at 78 MHz allowing wider spacing between amplifiers on the trunk.

11.1 MICROWAVE LINKS

Frequency allocations by the FCC permit operation in the band of 12.7 to 13.2 GHz. Relay stations of this type are called community antenna relay services(CARS).

The advantage of microwave transmission is that parabolic reflector dish antennas can be used to provide very high gain with a directive narrow beam to cover large distances.

11.2 FIBRE OPTICS

The latest type of communications link uses a cable made with thin glass fibres that serve as a conduit for light over long distances with little losses. The full cable-channel bandwidth can be used for amplitude modulation of the light source.

The advantages of using fiber-optic cable as a long-distance link are:

(1) The cable is not as heavy as copper conductors, making it convenient for installation.

(2) Attenuation of the light is much less than the losses with conduction or radiation of an RF carrier wave.

12.0 Subscription Payment

Each cable company sets its own scale of charges to subscribers ussually in several 'tiers' or groups of channels ranging from a few dollars a month for the four so called 'must carries' upto 30 dollars.

'CHURN' - subscribers who drop out every year, is of major concern to cable operators. This 'churn' could be attributed to:

(i) programme tiers maybe badly arranged

(ii) subscriptions to high

(iii)service and maitenance maybe to bad

This 'churn' could also indirectly influence subscriptions.

13 0 Disadvantages of Cable Tv

(i) The biggest hurdle to starting a network is costs. A cable system is fiendishly expensive, it costs about $15-20 million to install a system in a area.

(ii)Satellite channels have to be picked up with large parabolic antenna, approximately 35m diametre. The reason is for the antenna to receive maximum energy because meteorological conditions, such as rain and fog causes a detrimental attenuation of the signal.

(iii)Installation and labour costs.

Cabling ducts ussually have to be provided at extremely high costs.

13.1 Advantages of Cable Tv

(i) The major advantage is that the viewer will pay for what he/she wants to see.

(ii)Viewers are offered the chance to receive a completely ad-free movie channel.

(iii)The viewer determines how much he/she pays by simply entering the appropriate code for that particular channel/movie into the converter.

(iv)Viewers are offered multiple viewings, so he/she determines which viewing period is suitable.

(v) Viewers are offered many channels and the latest in entertainment around the clock.

14.0 M-NET

Basic principle of M-NET

M-Net uses the principle whereby it encodes both the video and audio signals.

The audio signal is digitised, encrypted and superimposed onto a sinewave to attain enhanced scrambling. At the receiving end ie, decoder/tv side, the digitised signal is converted to an analogue signal and under processor control various bits are inserted or deleted to obtain the original audio signal.

With the video signal, all the sync pulses are eliminated. All the available timeslots in both the vertical blanking interval (VBI) and horozontal blanking interval (HBI) are used for data transfer. During the VBI all lines are filled with a string of phony bits, 10101010...., at half the clock rate. In line 4, just before line 5 a reference pattern of 24 bits is inserted such that the last referance bit and the first preamble bit in the next HBI differ 79 cloc periods. The reference pattern, 1010111001100110011001010, follows the inserted phony bits.

The reference pattern will be recognised by the decoder to indicate that the signal is scrambled and the resulting field index pulse (FI) is used for reinserting the sync pulses.

14.1 General Description

Basic features of the system are:

(a) Elimination of all sync pulses.

(b) Shifting the back porch level.

(c) Using the available timeslots in the VBI and HBI for data transfer.

(d) Digitised encrypted sound in the HBI.

(e) Superimposed sinewvae on the audio and VBI data in order to attain enhanced scrambling.

(f) Inserting control and operational data in the VBI.

(g) Changing the video polarity under control of scene changes detected in the incoming video.

14.2 Disadvantages/Advantages of M-NET

The major disadvantage is that of subscription cost as the subscriber cannot determine his/her payment as in the USA. A fixed amount is set, increasing periodically.

The advantage of M-NET is that the viewer is offered an ad-free movie and 'repeats' so that the viewer may determine a suitable viewing time. The latest in entertainment, sport and news items presented virtually around the clock.

15.0 PROPOSED DESIGN/SYSTEM

This section deals with the description of the proposed design. A basic arrangement as shown in Fig 2-17 is outlined.

As previously mentioned the system consists of three circuit boards i.e.

(i) Programming Unit

(ii) Remote Control

(iii)Viewing Controller

Each board is explained separately.

15.1 Programming Unit

This unit is a 80C31 micro-controller based board supported by 8k RAM, eprom and keyboard controller. In addition it has a parallel port taken from the address and data bus, a 32x2 lcd and a keypad. The keypad consists of a 8x5 matrix of keys that is scanned every 1.25 microseconds by the 8279 keyboard controller to determine if one of the keys has been pressed. This relieves the micro-processor of scanning the keyboard and leaves it free to continue some other task.

The operating frequency of the keyboard is controlled by a 2MHz crystal referenced oscillator. The scanned keypad is used in the encode mode i.e. the counter provides a binary count that is externally decoded to provide the scan lines for the keypad. Debouncing of the keys is also a function of the keyboard/pad controller. Each time the debounce circuit detects a closed key, it waits 10 milliseconds to check if the key remains close.

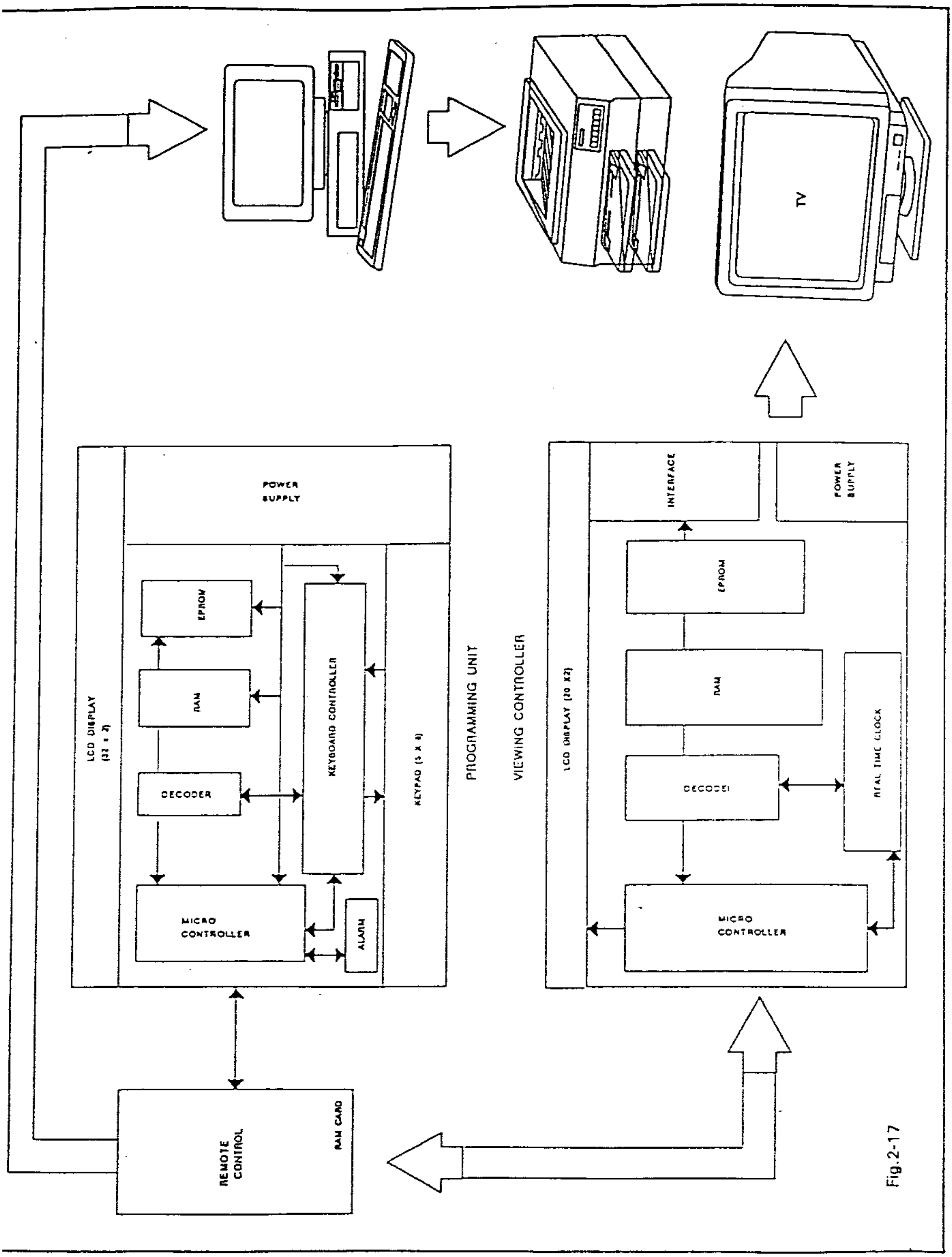

TV
POWER SUPPLY
EPROM
RAM
DECODER
KEYBOARD CONTROLLER
MICRO CONTROLLER
ALARM
LCD DISPLAY
PROGRAMMING UNIT
VIEWING CONTROLLER
INTERFACE
POWER SUPPLY
EPROM
RAM
LCD DISPLAY (20 X2)
REAL TIME CLOCK
MICRO CONTROLLER
REMOTE CONTROL
RAM CARD

Fig.2-17

If it does, the address of the key in the matrix is transferred to the 8x8 ram, FIFO in which it is stored. Each new entry is written into successive ram positions and each is then read in order of entry. Each time the ram is read, the keyboard controller sends an interrupt to the processor and displays the corrosponding letter/number. The keyboard controller would be used in the N-Key Rollover mode because each key depression would be treated independantly. If two depressions occur, the keys are recognised according to the order the keyboard scan found them.
A data base is kept of subscriber's names and addresses in the 8k RAM. The reason for using an 8k RAM is because the P.O. is not fully computerised yet. Downloading of the data base would have to be done manually to the mainframe. When the P.O. is fully computerised, the RAM would be upgraded and downloading of the data base would be directly into the computer.
This 8k RAM is mated to a Smart Socket making the memory non volatile. An audible and visual alarm is triggered when a certain memory location in RAM is reached, thus initiating a downloading process to the r.c. All external program memory would be executed from the eprom as the 80C31 mcro-controller has no on board ROM.
The remote control unit is powered from the parallel port. All applicable data would be entered into it via the keypad. This data consists of security codes of 27 bits, and the viewing dates i.e. the start and end dates.

15.2 Remote Control

This unit would contain the data for the security codes and viewing dates. The data is programmed into the non volatile SRAM using the programming unit. While data is being programmed, power is drawn from the programming unit.

15.3 Viewing Controller

This board is similar to the programming unit except that it has a real time clock(RTC) and TV interface instead of a keyboard controller. This SRAM is preprogrammed with data, i.e. time, date, rebooting information and security codes, by using the programming unit. Upon initial powering up of the viewing controller, the RTC is programmed from RAM, setting the time and date. A 20x2 LCD is used for displaying the time and dates. With the above method of setting the RTC, it makes it 'hands free' and thus no external interference occurs as this is very critical because the switching of the television is date dependant. The RTC has a dedicated nicad battery, for power backup during mains failure, that is trickle charged.

LED indicators for 'Licence Due' and 'Request Dates' are on and flashing respectively during the 'Tv Off' period. While the 'Request Dates' led flashes for approximately 20 seconds, the remote control should be plugged into the parallel port. If one misses the 'plug in' period, a wait delay period of approximately 5 seconds occurs before a new 'plug in' period becomes available. Power to the remote control is drawn from the viewing controller which is processor controlled to prevent any external interference

from the user.
Software compares the remote control security bits to the preprogrammed SRAM bits. Should this compare not pass, the 'Request Date' LED continues flashing. If this compare passes, the processor then compares the programmed 'START DATE' of the remote control to the RTC's present date. Should this compare not pass, the 'Request Dates' led continues flashing and the TV remains off. If this compare passes, the dates are downloaded from the remote to the RAM and displayed on the LCD. Both led indicators will be permanently on for approximately 20 seconds during which period the remote control must be removed. After the LEDS extinguish, the processor switches the Tv 'on' via the interfacing. A continous checking of the 'END DATE', which is stored in SRAM, to the RTC's date is done by the processor through software. When the dates are equal, the processor switches the Tv off and clears the programmed dates from SRAM and display. The 'Request Dates' and 'Licence Due' LED indicators come up again.
For the Tv to be switched 'on' again, a licence fee must be paid to the authority in charge, who will programme the remote control and the above procedure repeated again.

16.0 Cost Analysis

A survey was conducted in the Cape Town metropolitan area on the pricing of an M-NET decoder. The results of the survey showed that the price ranged from R499-R599 for the standard decoder and R699-R799 for the 9000+ stereo model.The cost setting up the M-NET network and its installation was unavailable. The cost to set up a cable network in a city, as mentioned earlier, is $15-20 million. In addition, a converter must also be purchased for approximately $100.

With respect to the design, Prevention of Pirate Tv, only the viewing controller and remote control will be considered as it will directly influence the price of the TV. An analytical analysis is shown in Table 2-3 on the pricing of the viewing controller and remote control.

Ref	Part	Description	Source	Unit	Price/unit	Total(R)
U1	74HC00	nand gate inverter	S&S Electronics Salt River	1	0.71	0.71
U2	4066	cmos switch	Hamrads Cape Town	1	1.55	1.55
U3	74HC573	latch	S&S Electronics Salt River	1	3.74	3.74
U4	80C31	microcontroller	EBE Observatory	1	10.55	10.55
U5	74HC138	3-8 decoder	S&S Electronics Salt River	1	1.71	1.71
U6	MM58274	real time clock	Electrolink Foreshore Cape Town	1	31.85	31.85
U7	6264	8k ram	Tarsus Technology Cape Town	2	8.95	17.9
U8	2764	8k eprom	Tarsus Technology Cape Town	1	11.95	11.95
U9	44-92	Gunther relay	Kopp Electronics Rondebosch	1	7.65	7.65
U10	LM7805	5v regulator	S&S Electronics Salt River	1	1.96	1.96
Q1,Q6,Q7	2N2907	transistor,pnp	Capetronics Bellville	3	0.72	2.16
Q2,Q4	VN2222	transistor,pnp	Capetronics Bellville	2	5.31	10.62
Q3,Q5	2N2222	transistor,mosfet	Capetronics Bellville	2	1.33	2.66
Y1	32Mhz	crystal	Microsource Edgemead	1	1.53	1.53
Y2	6Mhz	crystal	Microsource Edgemead	1	1.53	1.53
R1	5K	potentiometer	Microsource Edgemead	1	7.55	7.55
R2	2K2	resistor,1/4W	Hamrads Cape Town	1	0.31	0.31
R3	100K	"	Hamrads Cape Town	1	0.31	0.31
R4,11,12,13	10K	"	Hamrads Cape Town	3	0.31	0.93
R5,R9	2M	"	Hamrads Cape Town	2	0.42	0.84
R6	4K7	"	Hamrads Cape Town	1	0.31	0.31
R7,R8	2K	"	Hamrads Cape Town	2	0.31	0.62
R14	10R	"	Hamrads Cape Town	1	0.31	0.31
D1	1N4148	diode	Hamrads Cape Town	1	0.12	0.12
D2	Din746	3.6v zener diode	Hamrads Cape Town	1	0.39	0.39
D3,D4	LED	LED,red	Hamrads Cape Town	2	0.45	0.9
C1	100uf	capacitor,elec	Electronic Supermarket Cape Town	1	0.37	0.37
C2	6-36uf	varicap	Microsource Edgemead	1	1.95	1.95
C3	22pf	ceramic cap	Electronic Supermarket Cape Town	1	0.31	0.31
C4,C7	33pf	ceramic cap	Electronic Supermarket Cape Town	2	0.31	0.62
C5,C6,C9	0.1uf	poly cap	Electronic Supermarket Cape Town	3	0.31	0.93
C11,C12,C13	0.1uf	poly cap	Electronic Supermarket Cape Town	3	0.31	0.93
C8	1000uf	cap,eletrolytic	Electronic Supermarket Cape Town	1	1.42	1.42
C10	1uf	cap,electrolytic	Electronic Supermarket Cape Town	1	0.31	0.31
P1	conn 6way	wafer connector	Microsource Edgemead	1	0.36	0.36
P2,P3	14way	"	Microsource Edgemead	3	1.25	3.75
P4	2way	"	Microsource Edgemead	1	0.16	0.16
Miscellaneous						
VC board		printed cct board	Telkom RDC Cape Town	1	18.75	18.75
RC board		printed cct board	Telkom RDC Cape Town	1	3.12	3.12
RC box		RC housing	Electronic Supermarket Cape Town	1	6.53	6.53
S/socket		smart socket	Tarsus Technology Cape Town	2	29.85	59.7
rectifier		1.5A/5v	Electronic Supermarket Cape Town	1	0.99	0.99
transformer		220/9v	ERL Parow Industria	1	24.95	24.95
dipswitch		2 way	Kopp Electronics Rondebosch	1	2.74	2.74
header pins		pins	Microsource Edgemead	1	5	5
ribbon cable		26 way	Microsource Edgemead	1m	1.25	1.25
nicad battery		3.6v cells	Capetronics Bellville	1	38.05	38.05
pins d.i.l		40 way	Microsource Edgemead	2	0.82	1.64
fuse/holder		100mA	Microsource Edgemead	1	3.65	3.65
display	DMC2026	24chars x 2lines	Kopp Electronics Rondebosch	1	78.65	79.65
						377.79

TABLE 2-3

3 PROGRAMMING UNIT

Refer to Appendix A, drawing No A1 for a complete schematic diagram of the Programming Unit.

3.1 HARDWARE DESIGN

3.11 Processor(U5), Latch(U2), Eprom(U7)

The Programmer is comprised of a CMOS type microcontroller, 80C31, designated by 'C'. The 80C31 has four ports and for this application the ports will be used as follows:

(i) P0 is an 8bit open drain bidirectional I/O port. It is also the multiplexed low-order address and data bus during access to the keyboard controller, Display, SRAM and Remote Control port.

(ii) Port 1 is an 8bit bidirectional port with internal pullups. It is used as follows:

P1.0	RS for LCD unit
P1.1	R/W with 74HC00 for lcd
p1.2	Led indicator - on/off, alarm
p1.3	Buzzer, audible alarm
p1.4	Switches Vcc to parallel port for Remote Control

(iii) Port 2 is an 8bit bidirectional I/O port with internal pullups. This port emits the high order address byte during access to the RAM, remote control, keyboard controller and fetches code/instructions from external program memory.

(iv) Port 3 is used as follows:

P3.2	¯INT0 (external interupt 0)
P3.6	¯WR (external data memory write strobe)

P3.7 ¯RD (external data memory read strobe)

For this application the 80C31 micro-controller is using the on chip oscillator with a 12Mhz crystal. The crystal is connected between pins 19 (osc 1) and 18 (osc 2).
The reset input(RST) will reset the device if the RST pin is kept high for two machine cycles. The RST pin is connected to Vcc via a 10uF capacitor and to ground via a 8.2K resistor to reset the device automatically when power is applied. The reset of any peripheral, such as the keyboard controller, must be much quicker than the processor on power up as the peripheral is programmed and controlled by the processor.

Because the 80C31 has no on board ROM, ¯EA must be strapped to ground to enable the device to fetch code from external program memory locations 0000H to 0FFFFH. ¯PSEN strobes the EPROM and the code is read into the processor. The address latch enable(¯ALE) output is connected to pin 11 of the 74HC573 latch. The output pulse from the ¯ALE is used for latching the low byte of the address during accesses to external memory.
FIG 3.0 Shows the layout for external memory used.

3.12 Line Decoder(U4), 3 - 8

To select the various devices i.e., RAM, keyboard controller, LCD etc, the processor board is memory mapped and this is done by the 74HC138 under processor control. Three inputs(A, B and C) provide a selection of eight outputs. Only four outputs are used in this application.

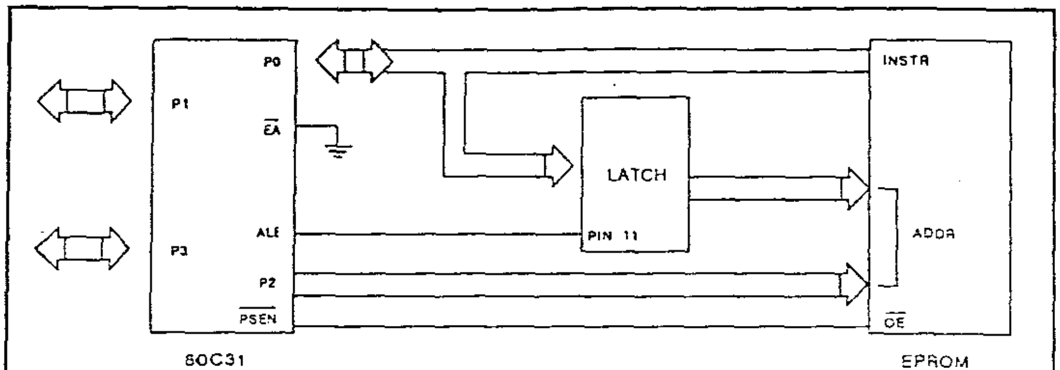


FIG 3.0

3.13 Keyboard Controller(U10), Line Decoder(U4), Oscillator

FIG 3.1, shows the layout of the 8279, keyboard controller, and its application.

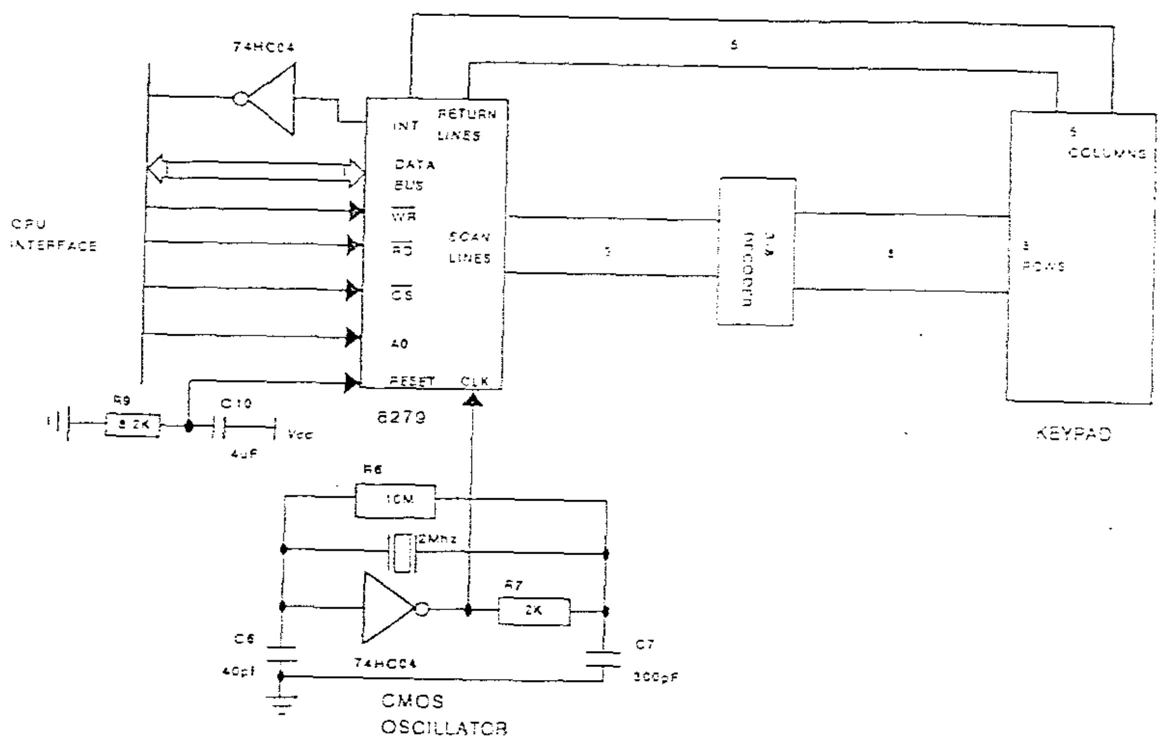


FIG 3.1

DATA — Bidirectional data bus - all data and commands between the processor and 8279 are transmitted on these lines.

CLK — Derived from a CMOS 2Mhz crystal oscillator to generate internal timing

RST — Reset input, a high on this pin will reset the device. The RST pin is connected to Vcc via a 4uF capacitor and to ground via a 8.2K resistor.

The four scan lines of which three are used is connected to the 3-8 line decoder. These lines are used to scan the keypad.
There are eight return lines of which five are used because of 8x5 matrix (40 keys). These lines are connected to the scan lines via the keys. Internal pullup resistors keep them high until a key closure pulls one low. A lookup table is used to scan for the correct key pressed.
The Interrupt Request(IRQ) goes high when data, the address of the key pressed in the matrix, is available in the FIFO. An inverter, 74HC04, brings this line low to strobe the processor on interrupt. After the FIFO ram has been read or is empty, A0 goes high indicating an end of interrupt to the micro-controller and IRQ goes low . The 74HC04 keeps INT0 high to the micro-controller to indicate no further interrupt has taken place.

3.14 Display

The DMC 32216 is a 5x8 dot, 32 character, 2 line dot matrix liquid crystal display module with three drivers and LSI controller mounted on a single printed circuit board. Contrast is adjusted by varying the voltage drop across R1 between 0v and 5v. The supply voltage must be between 4,75v and 5.25v.

3.15 Power Supply

Refer to Appendix A, drawing No A2 for the schematic diagram of the primary power supply of the Programming Unit. The power supply is a very basic yet efficient supply. It is a simple rectifier with a regulator, LM7805, that converts 220 AC voltage to +5v ± 0.04% DC.

The total current consumption is 140 mA. C9 is used for smoothing DC voltage ripple. C1 and C2 are used to filter any noise and for output stability.

3.2 PRINCIPLES OF OPERATION

3.21 Keyboard Controller

The following is a description of the major elements of the 8279 programmable keyboard controller. Refer to fig 3.1 and drawing No A1, Appendix A.

I/O Control and Data Buffers

The I/O control section uses the ¯CS, A0, ¯RD and ¯WR lines to control data flow to and from the internal registers and buffers. All data to and from the 8279 is enabled by ¯CS. A0 determines the character of information given by the processor. A logic one means the information is a command, logic zero means it is data. ¯RD and ¯WR determine the direction of data flow through the bidirectional data buffers, which connects the internal bus to the external bus. The devices are high impedance when CS=1.

Control and Timing Registers and Timing Control

These registers store the keyboard modes as programmed by the processor. The modes are programmed by setting A0=1 on the data lines, then sending ¯WR. The command is latched on the rising edge of ¯WR, decoded and the appropriate function set. The timing controls the basic timing chain. This is set by a ÷20 prescaler to yield an internal 100KHz operating frequency when pin 3 is clocked by a 2MHz signal. Other counters divide down the basic internal frequency to provide proper key scan, row scan and keyboard matrix scan lines.

Return Buffers and Keyboard Debounce

The five return lines are buffered and latched by return buffers. These lines are scanned and if the debounce circuit detects a closed switch, it waits 10 milliseconds to check if the switch remains closed. If it does, the address of the key in the matrix is transferred to the FIFO.

FIFO RAM

This is a 8x8 RAM. Each new entry is written in successive RAM positions and each is read in order of entry. FIFO status also keeps track of the number of characters the FIFO and whether it is full or empty. Status can be read by a ¯RD with ¯CS low and A0 high.

Software operation sets the keyboard mode, program clock, read FIFO and end interrupt mode.

4 REMOTE CONTROL (RAM card)

Refer to Appendix A, drawing No A3 for the schematic diagram of the Remote Control

4.1 HARDWARE DESIGN

The 6264, 8x8k, CMOS Static RAM is used. The 6264 has three control inputs i.e., chip select(¯CS), output enable(¯OE) and write enable(¯WR). These inputs must be logically active in order to write data to the device or to obtain the ¯RD and ¯WR outputs from the device. The ¯OE and ¯WR is directly connected to the ¯RD and ¯WR outputs of the 80C31. R1 is used as a pullup resistor on ¯WR line to keep ¯WR permanently high during 'plug out' period to prevent data from being lost or corrupted. Pin 20 is used to select the RAM. The RAM can be interfaced with a link card via an edge connector for downloading purposes.

The RAM is permanently mounted onto the DS1213B smart socket to provide a compatible solution to problems associated with memory volatility. The smart socket has a built in CMOS controller circuit and an internal lithium battery. The smart socket monitors the incoming Vcc for an out of tolerance condition i.e. between 4.75v and 4.5v. When such a condition ocurrs, the internal battery is automatically switched on and write protection is unconditionally enabled to prevent data from being corrupted .

By using the smart socket, battery backup design is eliminated and printed circuit board space is saved since the combination of the smart socket and the RAM uses no more area than the memory alone. Data retention is estimated to be approximately ten years.

5 VIEWING CONTROLLER

5.1 Introduction

When a RTC is used in a microprocessor a precise oscillator is required to control real time counting, a backup power source to maintain time keeping and some form of write protection when the main system power fails to ensure a reliable and trouble-free system. Refer to Appendix A, drawing No A4 for the schematic diagram of the viewing controller.

5.2 Hardware Design

As with the programming unit, a 80C31 microprocessor working at 6MHz is used to form the heart of the viewing controller.

Port 1 is used as follows:

P1.0	Control of TV IF
P1.1	'Licence Due' LED indicator
P1.2	'Request Due' LED indicator
P1.3	Power control to Remote
P1.4	Power control to TV
P1.6	RS for display
P1.7	¯RD_WR for display

Ports 3, 2 and 0 are used in the same manner as with the programming unit, thus no further explanation is required as the operation is similar.

5.21 Display

The display, DMC 20216 is similar to that used in the programming unit except that it is a 20 character, 2 row display. As the operation is similar no further explanation will be given.

5.22 Power supply

The design and operation of the power supply is similar to that of the programming unit, thus no further explanation will be given. The total current consumption for the viewing controller board is 40mA.

5.23 Real Time Clock

The MM58274 microprocessor compatible RTC is used. The reason for using a RTC is to monitor the dates for switching the TV on and off. The RTC also shares part of the address (A0 - A3) and data bus (AD0 - AD3) with the RAM. The RTC has write (~WR), read (~RD) and chip select (~CS) as its control inputs. The RTC has 15 internal registers to read from or to write to. The ~WR input is used to initialize the RTC internal registers, while the ~RD input is used to read the date and time from these registers. The RTC uses the on chip oscillator with a 32,728Khz crystal, with a trimmer capacitor, C2, for fine tuning.
The RTC uses 3 x 1.2v nickle caduim batteries as backup power supply that is trickle-charged from the system power supply via resistor, R3, as shown in drawing A4, Appendix A.
To prevent corrupt data on power failure, some form of write protection is required. This is done by using dip switch, S1, keeping the ~WR line 'locked out' after the RTC has been programmed. The ~WR line of the RTC is then pulled high by R13. The switching of S1 requires the intervention of the operator to alter time data, but this application is quite adequate as the time data is programmed only once.

5.3 Principles of RTC Isolation and Backup Power Operation

When the mains power supply fails, there are two reasons for disconnecting the RTC circuit from the rest of the system i.e.

(i) To prevent the backup battery powering the rest of the system.

(ii) To minimize battery current (and extend battery life) by preventing current leakage out of the RTC pins.

5.31 Isolation Technique

Refer to FIG 5.0 for the Isolation Circuit.

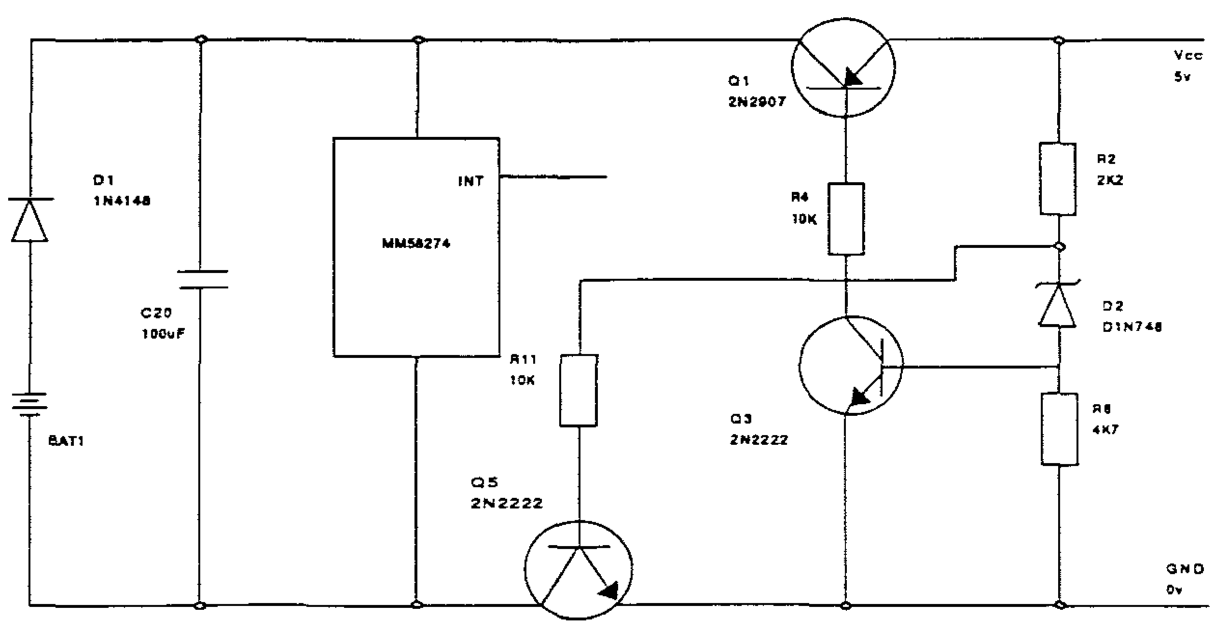


FIG 5.0 ISOLATION CIRCUIT FOR RTC

Q5 and Q1 are the disconnecting devices. Q5 which is controlled by the bias chain R2, R11, D2 and R6 switches Q5 ground potential through to the system. Q1 is controlled by Q3. Q3 is switched on by its bias chain R2, D2 and R6. Q3 and R4 turns on Q1, which connect the positive supply to the system. D1 isolates the backup battery from the supply. Q1 is necessary to prevent the battery discharging during mains failure as the interrupt output is being used.

If Q1 was not used the battery would discharge via the path shown in FIG 5.1.

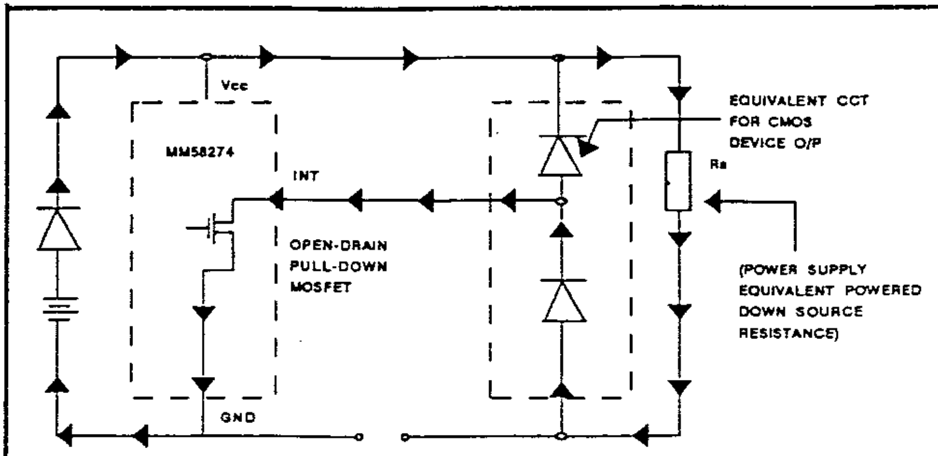


FIG 5.1 DISCHARGE PATH

As the ~WR line is 'locked out' by S1 after the RTC has been programmed, no data corruption on power fail can take place.

The purpose of C20 is to keep the clock operating without any supply for approximately 30-40 seconds should the batteries require replacing for whatever reason. Should the switching between power supplies be 'slow', the RTC would draw power from C20 during the interim period.

Calculation of time for C20 :

$$dQ/dt = I$$

$$\Rightarrow \quad \int dQ = \int I dt$$

$$Q = I.t$$

$$\text{and} \quad Q = CV$$

where C = 100uF, V = 5 volts and I = 10uF

=> t = RC and R = V/I

therefore t = CV/I

and I.t = CV but I.t = Q

=> Q = CV

=> I.t = CV

=> t = CV/I

= 100 x 10-6 x5/10 x 10-6

= 50 seconds

6 INTERFACING

6.1 Introduction

The interfacing of the viewing controller processor board to the TV might have to be slightly modified for different makes of TVs. For this application a National monochrome television was used.

6.2 Interfacing Methods

Refer to FIG 6.0

The following methods of interfacing the viewing controller board to the television were attempted:

(i) To control the video out signal, point C.

(ii) To control the intermediate frequency(IF), point D.

(iii) Switching the power to the tuner, the audio circuits, the IF and video out circuit, point B.

The results of the various methods attempted are discussed as follows. Refer to Appendix A, drawings A4 and A5.

(i) Video Out Signal

Control was taken from P1.6 of the 80C31 and connected to the base of Q15 of the video out circuit of the TV. By correctly biasing the base of Q15, the video out signal could be controlled. This resulted in the screen blanking during the tv 'off' period with sound. During the 'on' period, the picture was clear with a slight interferance.

A possible reason for this interference is that Q15 could have been overdriven to saturation point which caused some distortion at its output, thus causing a ripple effect through the video output stage which resulted in ghosting on the screen.

(ii) IF Control

This was done by using P1.0, from the 80C31, as the control via the 4066(U2), pin 13, for switching the IF. During the 'off' period the screen was 'snowy' and no tuning was possible. During the 'on' period the picture was slightly blurred.

The possible cause of the blurring could be due to:

(1) the offset delay of the 4066.

(2) impedance mismatch between the IF coaxial cable and the 4066.

(3) attenuation of signal due to extra length of coaxial cable introduced. The length of the coaxial cable between the tuner and IF stage might have been critical.

(iii)Power Control

P1.5 of the 80C31 was connected to Q7 via R13, the emitter being connected to U9, pin 6. Q7 under processor control was switched on and off according to the viewing dates. R131 from the TV circuit was removed. This resistor was then incorporated into the design of the viewing controller board as R14. R14 and C144 (from the TV board) form a low pass filter for the audio circuits. R14 also gave a voltage drop of approximately 1v for the correct supply voltage, 10.7v, for IC12.

During the 'off' period the screen blanked with no sound present. Although the power was interrupted to the AGC amplifier, tuner and audio circuits, power on the television tube remained as if the tv was in 'standby' mode. The picture during the 'on'period was clear.

Of the methods investigated, the method of controlling the power to the AGC amplifier, tuner and audio circuits was chosen as this method would override all the others and their results would not be seen.
If the connecting wires were simply removed from the viewing controller board as to bypass it, this would result in a fuzzy picture and scratchy sound.

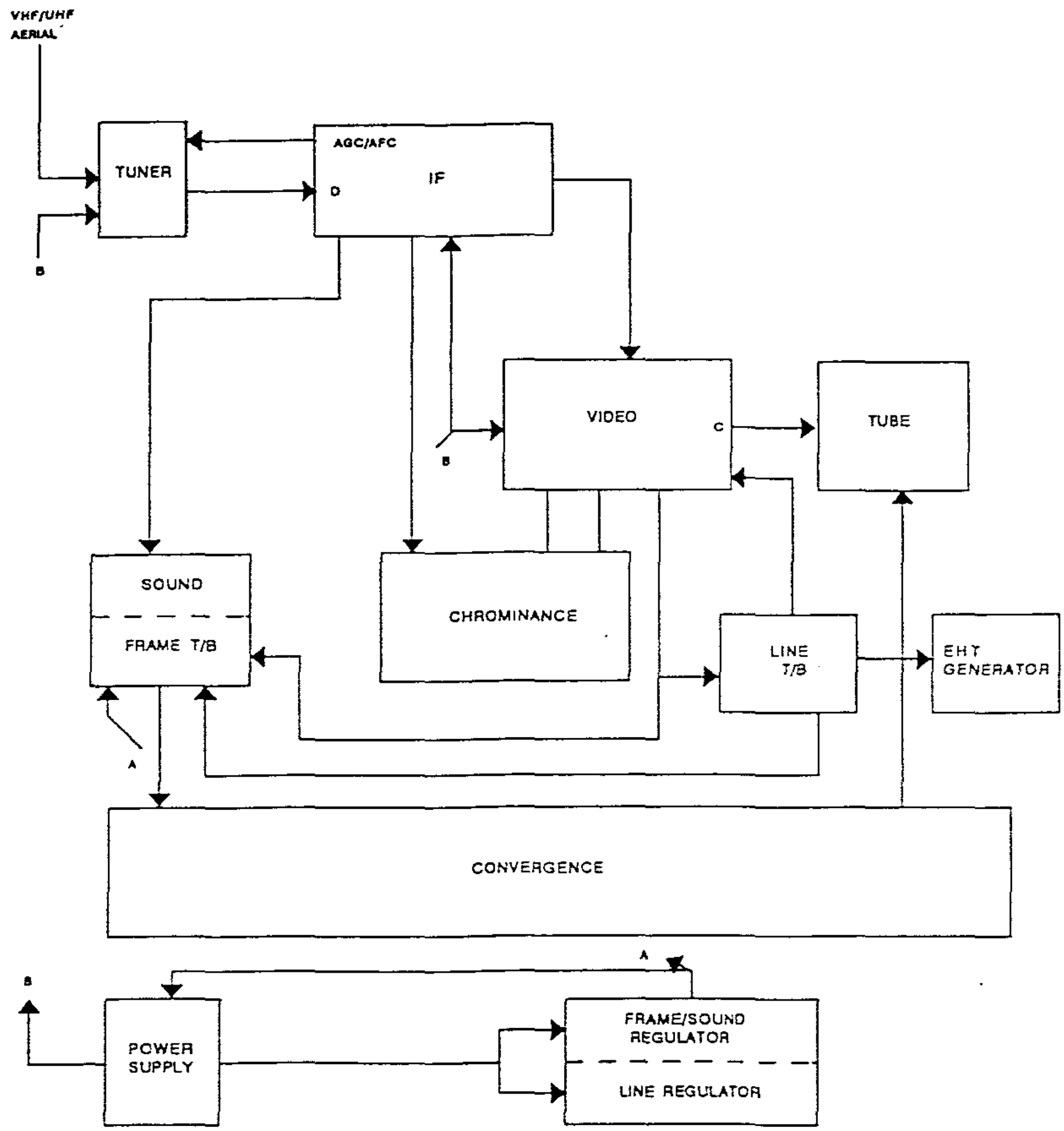


FIG 6.0 BASIC TV LAYOUT

7 SOFTWARE DEVELOPMENT

7.1 Program Development Tools

The Intel Ice 5100/52 was used for the development of the software for this project. The Ice in circuit emulator is a high level, interactive debugging system that is used to test hardware and software of a project that is based on the MC-51 family of micro-controllers. Using the Ice 5100/52 emulator in this application proved to be a great asset as real time operation was required.

The 80C31 in the programming unit and the viewing controller is programmed with programmes that were written in assembly language. The Asm51 compiler was used to compile the programs. The advantage of writing in assembly language, is the minimizing of execution time.

Refer to Appendix B and Appendix C for the programme listings and flow diagrams.

7.2 Programming Unit

The processor is memory mapped as follows:

7.21 Ram

CS1

Start Address: 2000 Hex

End Address: 3FFF Hex

7.22 Remote Control

CS3

Start Address: 6000 Hex

End Address : 7FFF Hex

7.23 Keyboard Controller

CS4 selects the keyboard controller

The keyboard controller has to be initialized to the mode it is to operate in. FIG 7.0 shows the various options.

The internal prescaler is set to select the desired operating frequency. In this case 100 KHz. Refer to FIG 7.1. This value is determined by the operating clock frequency. The maximum value that the prescaler can be programmed to is 31. For this application, 20 is programmed to the internal prescaler because a clock frequency of 2MHz is used.

FIG 7.0 KEYBOARD MODE SET

MSB							LSB
*	*	*	*	*	k	k	k

where bits:

KKK	
000	ENCODED SCAN KEYBOARD - 2 KEY LOCKOUT
001	DECODED SCAN KEYBOARD - 2 KEY LOCKOUT
010	ENCODED SCAN KEYBOARD - N KEY ROLLOVER
011	DECODED SCAN KEYBOARD - N KEY ROLLOVER

FIG 7.1 PROGRAM CLOCK

*	*	*	P	P	P	P	P

WHERE BITS PPPPP DETERMINE THE VALUE OF THE INTEGER

The " End of Interrupt " mode must be set to inform the processor it is the end of the interrupt.

7.24 Display

CS5 selects the DMC 32216 display. The display address is 0A000 Hex. The DMC 32216 operates from an extensive instruction set where the instructions are sent to the " Control Word Address" 0A000H.

Refer to FIG 7.2 for the full instruction set.

7.3 VIEWING CONTROLLER

As the memory map is similar to that of the programming unit, except for the real time clock, only the RTC's addressing will be discussed.

7.31 Real Time Clock

CS3

Starting Address: 08000H

End Address : 0800FH

Refer to FIG 7.3 for the address decoding of the internal registers of the RTC.

There are three control registers that control different operations of the internal registers of the RTC.

(i) The clock setting register, which is used for setting up:

(a) the leap year indicator

ADDRESS / INSTRUCTION	RS	R/W	D7	D6	D5	D4	D3	D2	D1	D0
DISPLAY CLEAR	0	0	0	0	0	0	0	0	0	1
CURSOR HOME	0	0	0	0	0	0	0	0	1	.
ENTRY MODE SET	0	0	0	0	0	0	0	1	I/D	S
DISPLAY ON/OFF CONTROL	0	0	0	0	0	0	1	D	C	B
CURSOR DISPLAY SHIFT	0	0	0	0	0	1	S/C	R/L	.	.
FUNCTION SET	0	0	0	0	1	DL	1	0	.	.
CG RAM ADDRESS SET	0	0	0	1	ACG					
DD RAM ADDRESS SET	0	0	1	ADD						
BUSY FLAG/ADDRESS READ	0	1	BF	AC						
CG RAM/DD RAM DATA WRITE	1	0	WRITE DATA							
CG RAM/DD RAM DATA READ	1	1	READ DATA							

NOTES:

I/D = 1 :INCREMENT — 0 = DECREMENT
S = 1 :DISPLAY SHIFT — 0 = DISPLAY FREEZE
D = 1 :DISPLAY ON — 0 = DISPLAY OFF
C = 1 :CURSOR ON — 0 = CURSOR OFF
B = 1 :CHAR BLINKS — 0 = CHAR DOES NOT BLINK
S/C = 1 :DIPLAY SHIFT — 0 = CURSOR MOVE
R/L = 1 :RIGHT SHIFT — 0 = LEFT SHIFT
D/L = 1 :8 BITS — 0 = 4 BITS
B/F = 1 :INTERNAL OPERATION — 0 = END OF INT. OPERATION

FIG 7.2 INSTRUCTION SET

(b) AM/PM indicator

(c) 12/24 hour mode

Refer to FIG 7.4 for the layout of the clock setting register.

(ii) The interrupt register can be programmed as follows:

(a) to control the interrupt time which generates interrupts at time intervals.

(b) to select the required delay period.

(c) to be a single or repeated interrupt timer.

Refer to FIG 7.5 for the listing of the different time delays and the data words that select them in the interrupt register.

(iii)The Control Register

The control register is responsible for controlling the operation of the clock and supplying status information to the processor. Refer to FIG 7.6 for the control register layout.

FIG 7.3 ADDRESS DECODING OF REAL TIME CLOCK INTERNAL REGISTERS

REGISTER SELECT	ADDRESS (BINARY)				(HEX)	ACCESS
	AD3	AD2	AD1	AD0		
0 CONTROL REGISTER	0	0	0	0	8000	SPLIT READ AND WRITE
1 TENTH OF SECONDS	0	0	0	1	8001	READ ONLY
2 UNITS SECONDS	0	0	1	0	8002	R/W
3 TENS SECONDS	0	0	1	1	8003	R/W
4 UNITS MINUTES	0	1	0	0	8004	R/W
5 TENS MINUTES	0	1	0	1	8005	R/W
6 UNITS HOURS	0	1	1	0	8006	R/W
7 TENS HOURS	0	1	1	1	8007	R/W
8 UNITS DAYS	1	0	0	0	8008	R/W
9 TENS DAYS	1	0	0	1	8009	R/W
10 UNITS MONTHS	1	0	1	0	800A	R/W
11 TENS MONTHS	1	0	1	1	800B	R/W
12 UNITS YEARS	1	1	0	0	800C	R/W
13 TENS YEARS	1	1	0	1	800D	R/W
14 DAY OF WEEK	1	1	1	0	800E	R/W
15 CLOCK SETTING/ INTERUPT REGISTERS	1	1	1	1	800F	R/W

FIG 7.4 CLOCK SETTING REGISTER LAYOUT

FUNCTION	DATA BITS USED				ACCESS	COMMENTS
	DB3	DB2	DB1	DB0		
LEAP YEAR COUNTER	X	X			R/W	0 INDICATES LEAP YR
AM/PM INDICATOR			X		R/W	0 = AM 1 = PM
12/24HR SELECTED BIT				X	R/W	0 = 12HR 1 = 24HR MODE

FIG7.5 INTERUPT CONTROL REGISTER

FUNCTION	CONTROL WORD				COMMENTS
	DB3	DB2	DB1	DB0	
NO INTERRUPT	X	0	0	0	INTERRUPT OUTPUT CLEARED START/STOP BIT SET TO 1
0.1 SECONDS	0/1	0	0	1	
0.5 SECONDS	0/1	0	1	0	
1 SECONDS	0/1	0	1	1	DB3 = 0 FOR SINGLE INTERRUPT
5 SECONDS	0/1	1	0	0	DB3 = 1 REPEATED INTERRUPT
10 SECONDS	0/1	1	0	1	
30 SECONDS	0/1	1	1	0	
60 SECONDS	0/1	1	1	1	

FIG 7.6 THE CONTROL REGISTER LAYOUT

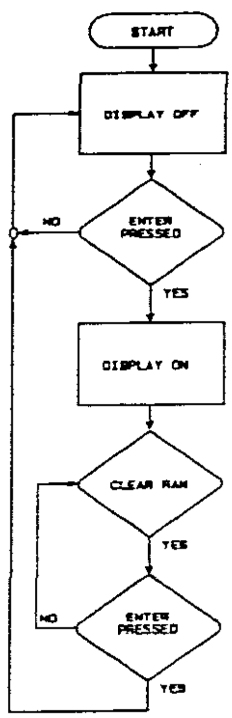
ACCESS (ADDRESS 0)	DB3	DB2	DB1	DB0
READ FROM :	DATA CHANGE FLAG	0	0	INTERRUPT FLAG
WRITE TO :	TEST 0 = NORMAL 1 = TEST MODE	CLOCK ST/STP 0 = CLK RUN 1 = CLK STOP	INT SELECT 0 = CLK REG 1 = INT REG	INT START/STOP 0 = INT RUN 1 = INT STOP

8 OPERATING INSTRUCTIONS

8.1 Setup Procedure

To set up the PU, its RAM must first be preprogrammed for database purposes as a scan technique is used to keep track of the memory locations. This is done as follows:

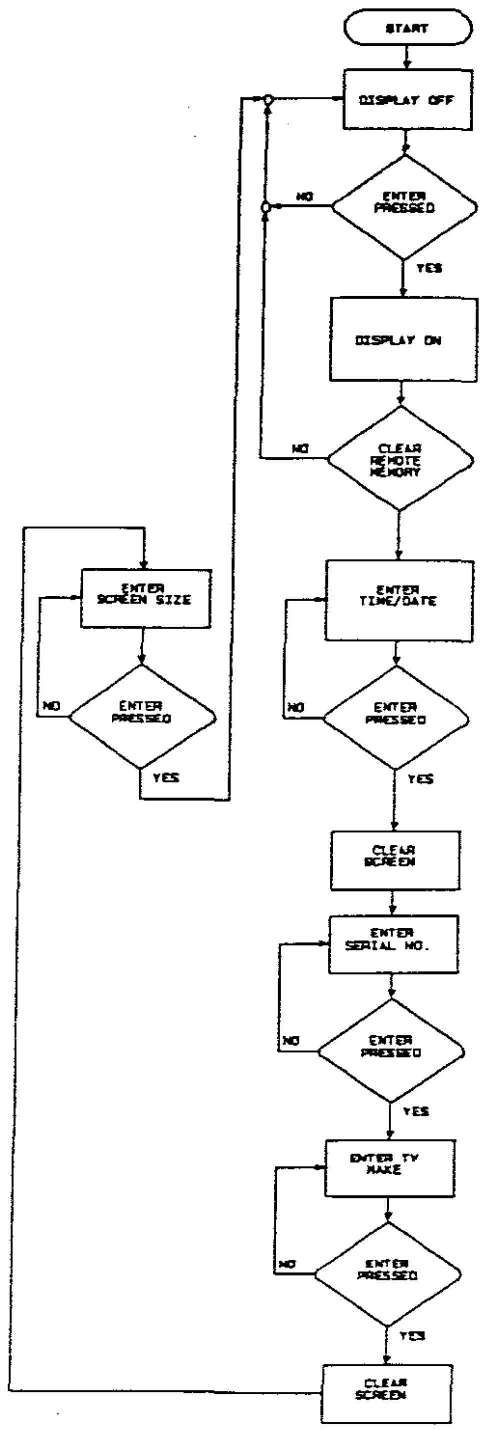
The RAM and its smart socket is removed from the PU and plugged into the RC socket. The RC is plugged into the PU. When power is applied to the PU, the 80C31 microprocessor is reset and the program is initialized and executed in the following manner:



The RC is unplugged from PU and the RAM and its smart socket is removed from the RC and placed into its socket on the PU board.The PU is now ready for use.

8.2 Time, Date and Rebooting Setup

For the RAM on the VC to be setup the above procedure is applied but with a different EPROM. On powerup the 80C31 microprocessor is reset, the program is initialized and executed in the following manner :



Appendix D shows the programme listing for preprogramming the RAM of the VC. The RAM and smart socket is then replaced onto the VC board.

Upon initial power up, the time, date and rebooting information is set according to appendix C.

The above procedure refers to 'FACTORY SETUP'.

The PU would be installed at places where television licences could be paid eg. Post Office.

Each television set would have a VC board with its unique security code installed, documentation and a RC.

8.3 Operating the PU

On powering up, the 80C31 microprocessor is reset and the programme is initialized and executed as in Appendix B.

The user presses the 'enter ' key and the following message is displayed:

```
* CLEAR REMOTE CONTROL MEMORY *
         YES/NO - Y/N  _
```

With this message on the screen, the authority plugs the subscriber's RC into the PU. If 'N' is entered, the screen clears and the RC is not cleared. If 'Y' is entered and the 'enter' key is pressed, the RC is cleared and the following message appears on the screen:

```
NAME:
```

The name is entered eg. J SMITH. Each letter of the name entered is displayed and successively placed into RAM memory starting from 0000H. If the user makes a mistake, the ' -> 'key is pressed which backspaces to the letter to be corrected, at the same time the ram memory location would also be corrected.
When the 'enter'key is pressed the following message is displayed:

NAME:J SMITH

ADD:

The address is entered eg. 10 LOOP ST CAPE TOWN 8001. The above procedure is repeated. At this stage the display would be as follows:

NAME:J SMITH

ADD:10 LOOP ST CAPE TOWN 8001

When the 'enter' key is pressed, the screen clears and the following message is displayed:

SERIAL NO   :XXX XXX XXX X

Each letter or number eg. ABC 123 DEF 0, of the 'serial no ' entered is displayed and successively stored into the RC's RAM location selected. When the 'enter' key is pressed the following message is displayed:

SERIAL NO :ABC 123 DEF 0

TV MAKE :XXXXXXXXXX

The 'TV make' is entered eg. SONY, and stored in the RC's memory. All unused 'XXX's' must be deleted by using ' -> 'key, forward space. The display on the screen would be:

SERIAL NO :ABC 123 DEF 0

TV MAKE :SONY

When the 'enter ' key is pressed the screen clears and the following message is displayed:

SCREEN SIZE:XXXX

The screen size would be entered and stored in the RC's memory.

The above-entered data forms the 27 bit security code. This data could have any combination provided the information corresponds to that of the preprogrammed ram of the VC.

When the 'enter' key is pressed the screen clears and the following message is displayed:

START DATE :YY:MM:DD

The 'start date' eg. 93:09:20 , is entered and stored in the RC's memory. When the'enter' key is pressed the following message is displayed:

START DATE :93:09:20

END DATE :YY:MM:DD

The 'end date' eg.94:09:20, is entered and stored in the RC's memory. The display would be:

START DATE :93:09:20

END DATE :94:09:20

When the 'enter' key is pressed, the screen clears, the RC is unplugged from the PU and handed back to the subscriber upon payment.

This procedure is repeated for every subscriber.

8.4 OPERATING THE VIEWING CONTROLLER

Assume that the RAM has been preprogrammed with security bits, time, date and rebooting information as described previously. Let us assume the time and date is 09:33 and 93:09:20.

On powering-up the VC board the 80C31 is reset and the programme is initialized and executed as shown in Appendix C .The subscriber would see the following on the display.

ST: 09:33

END: 20

WHERE

ST	Start of viewing date
END	End of viewing date
09:33	Time
20	Date - day

The RD LED flashes and the LD LED is permanently on. The RD LED flashes for approximately 20 seconds. During this period the subscriber must plug the RC into the VC. If the subscriber is not sure how long the RD led has been flashing, he/she should wait until the next flashing period occurs. If the security codes and the 'start date' are accepted, the viewing dates are downloaded into the VC's RAM and the display would be as follows:

```
ST:93 09 20    09:33
END:94 09 20      20
```

Both LEDS come on permanently for approximately 20 seconds. During this period the subsriber must remove the RC. When both LEDS extinguish, the microprocessor switches the TV on. When the 'end date' rolls over the microprocessor switches the TV off and the dates are cleared from memory and the display. The screen display should be as follows:

```
ST:            00:00
END:              20
```

Both LEDS come on again. The RD LED flashes and the LD LED is permanently on.

To switch the TV 'on' again, the licence fee has to be paid, the RC programmed and the above procedure repeated.

9 CONSTRUCTION AND INSTALLATION

9.1 Viewing Controller

Refer to FIG 9.0 for drawing of casing.

The VC processor board will be housed in a 'cash register' style casing. This housing would contain the power transformer, AC adapter plug, display and keypad. The display should be mounted at an angle of between 26-30 degrees to facilitate viewing. Any angle less than 26 degrees causes the lcd to be refractive.

The casing has a removable bottom lid to which the board and transformer will be mounted.

9.2 Remote Control

This board is mounted in plastic box , having outside dimensions 6 x 5 x 2 cm. A slot was cut in at one end to house the plug.

9.3 Viewing Controller

This board measures 15 x 8.5 cm. It is installed in a suitable place in the tv. Precaution must be taken on the length of the leads of the oscillator circuit of the RTC as they are susceptible to parasitic track to track capacitance and noise thereby reducing the oscillator stability. The absolute maximum length of the pcb track on either pin 14 and 15 is 2.5 cm.

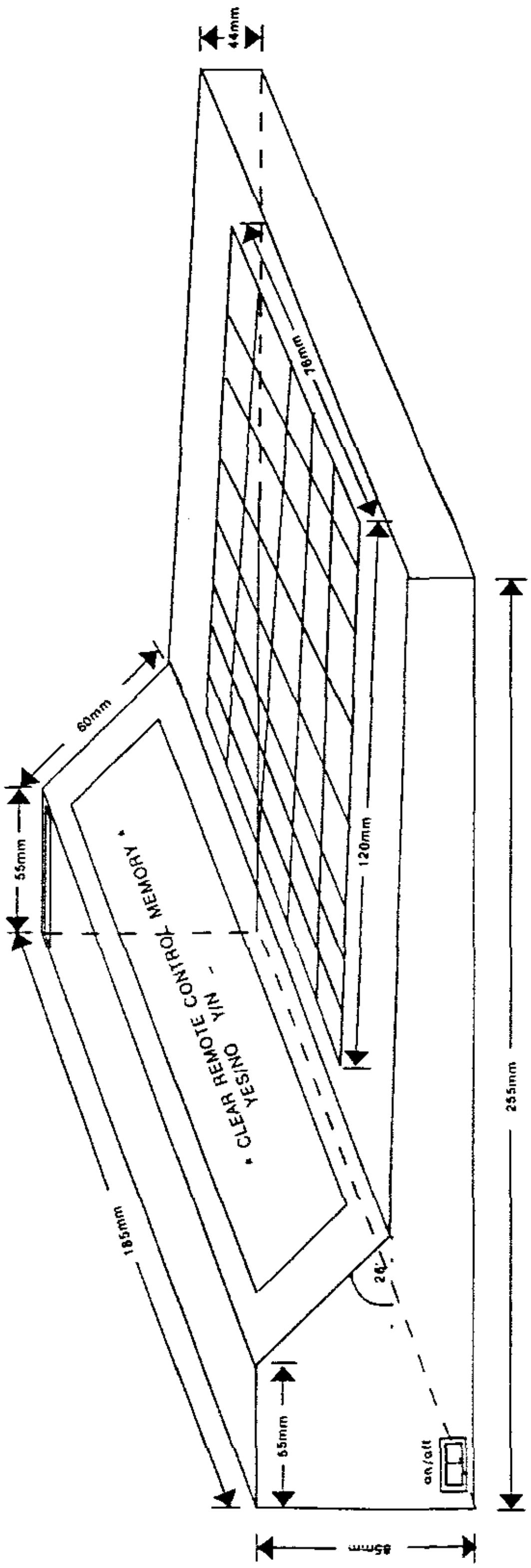

44mm
76mm
120mm
60mm
55mm
185mm
255mm
55mm
85mm
CLEAR REMOTE CONTROL MEMORY
YES/NO Y/N

FIG 9.0

10 PROBLEMS ENCOUNTERED

A number of problems were encountered with the design but only a few are worth mentioning i.e.

(i) Power switching between PU and RC, VC and RC.

(ii) Data lost from the RC.

(iii) Time data corrupted on power fail.

The above problems will be discussed and the possible solutions given.

10.1 Possible Solutions

(i) Power Switching

Refer to Appendix B, page B5 and Appendix C, page C3.
When the RC was plugged into the PU during programming and into the VC during downloading, the power seemed 'lost' or the executed program seemed to miss 'the power command'. This was solved by inserting a delay period after power was switched to the RC/VC, as if to give the power 'settling time'.

(ii) Data Lost or Corrupted

Refer to schematic drawing A4, Appendix A.
After the RC was programmed with data, i.e. the viewing dates and security codes, the data was lost or corrupted when it was plugged into the VC. This was overcome by keeping the WR line 'high' during plug in/out. The write line is kept "high" by using R1 as a pullup resistor on the RC.

(iii)Time Data Corrupted

Refer to schematic drawing A4, Appendix A.

The problem with the time data occurred during the main supply power fail. Although, there was a backup power supply and the WR line switched by S1, the time and date were corrupted when the main power supply was restored.

At first, it seemed that the switching between the power supplies were 'slow'. To remedy this, C20 was inserted across the power supply of the RTC. The RTC would draw power from C20 during this interim period, however, this did not solve the problem.

As the WR line was disconnected from the RTC, this left the WR line floating. R13 was connected to the WR line and to pin 16 of the RTC. This caused the WR line to be permanently 'high' and solved the problem of data corruption on power restore.

11 FUTURE DEVELOPMENT

As can be seen from FIG 1.1 (CH 1),high revenue loss occurs due to unpaid licences. One possible way to prevent this problem, is for future television sets to have built in decoders which could either be time or frequency controlled from a central broadcasting station.

Another alternative would be to upgrade the existing telecommunication network to accept television transmission to each household. The television signal would be controlled from the exchange that feeds the area. Detailed billing could then be done monthly on a time base system. (Detailed billing to be introduced by Telkom in 1994).

When the television set is switched on, a signal from the television is transmitted to the exchange equipment which switches the television signal thru. Full duplex communication takes place. When the television is switched off, the communication link is broken and the exchange equipment would cut the television signal to that particular household.

A further possibility is to establish an optic fibre point to point transmission system. Each city/suburb would have its own network, teed off from the main system. Individual streets could be grouped and linked to a central control box to control the signal to the individual homes. Upon payment the control box is activated to switch the television signal thru to that particular household.

12 CONCLUSION

When one considers the revenue losses the SABC TV incurrs due to unpaid TV licences, then some method of control must be implemented. The proposed system seems feasible as the system is easy to implement and is not costly. The PU is fully portable and can be relocated in any region without the extra cost of installing transmitting equipment as compared to M-NET. Wherever the broadcasting television signal is received, the system can be implemented. The operation of the PU is straight forward and is display driven.

The subscriber's operation of the remote control is simple, easy and requires no expertise knowledge.

The only drawback is that of a legal aspect as the system cannot be implemented in an existing private television set. However, with the production of new televisions, the system can be fully implemented.

Maintenance of the proposed system is minimal. The only maintenance of the VC board is the resetting of the time and date if ever required.

Thus, with costs, installation, maintenance and ease of operations considered, the proposed system should be considered in order to prevent pirate television, thereby saving the SABC TV revenue in unpaid licence fee.

13 BIBLIOGRAPHY

Designing and maintaining the CATV and Small TV Studio

(Ch 9 : Distribution Systems)

......................Kenneth Knecht

Cable and Satellite Tv

(Ch 1 : Multichannel Choice)

(Ch 2 : Cable Revolution)

(Ch 5 : Cinema in the home)

.......................Alan Burkitt

Television and Video Systems

(Ch 15: Cable Television)

.......................Bernard Grob

Radio Frequency Transmission Sysytems

(Ch 10: Transmission Line and Waveguide)

..............Jerry C Whitaker

Digital Communications

Ch 1: Fibre Optic Transmission

....................Thomas C Bartee

8 Bit Embedded Processors, Memory Products, Peripherals

......................Intel Data Books

Various Data Books

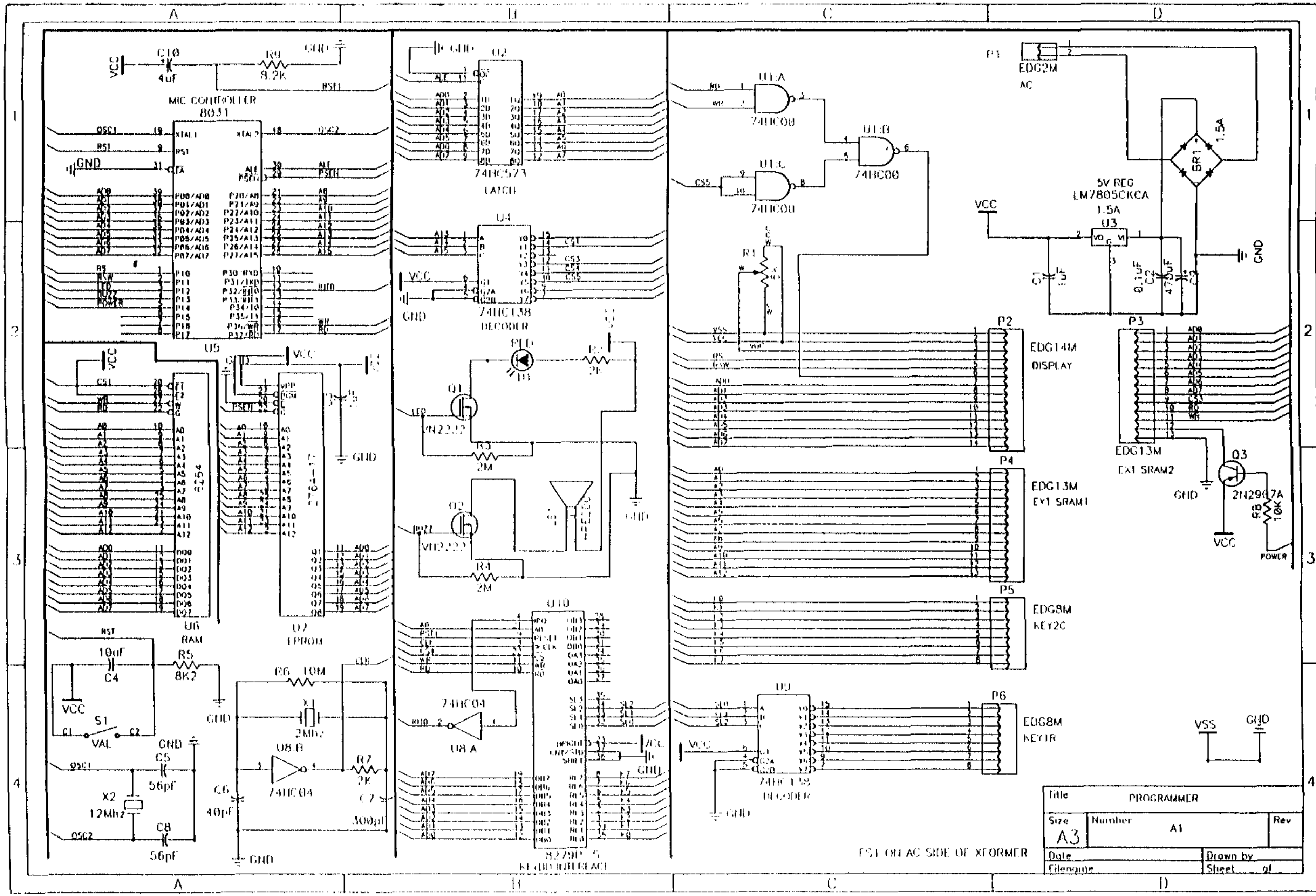
List of Abbreviations

PU	Programming Unit
RC	Remote Control
VC	Viewing Controller
IF	Intermediate Frequency
CS	Chip Select
~CS	'low' Chip Select
~WR	'low' Write
~RD	'low' Read
~OE	'low' Output Enable
~INTO	'low' External Interrupt
RST	Reset
RD	Request Dates
LD	Licence Due
RTC	Real Time Clock
HBI	Horizontal Blanking Interval
VBI	Vertical Blanking Interval
LED	Light Emitting Diode

APPENDIX A

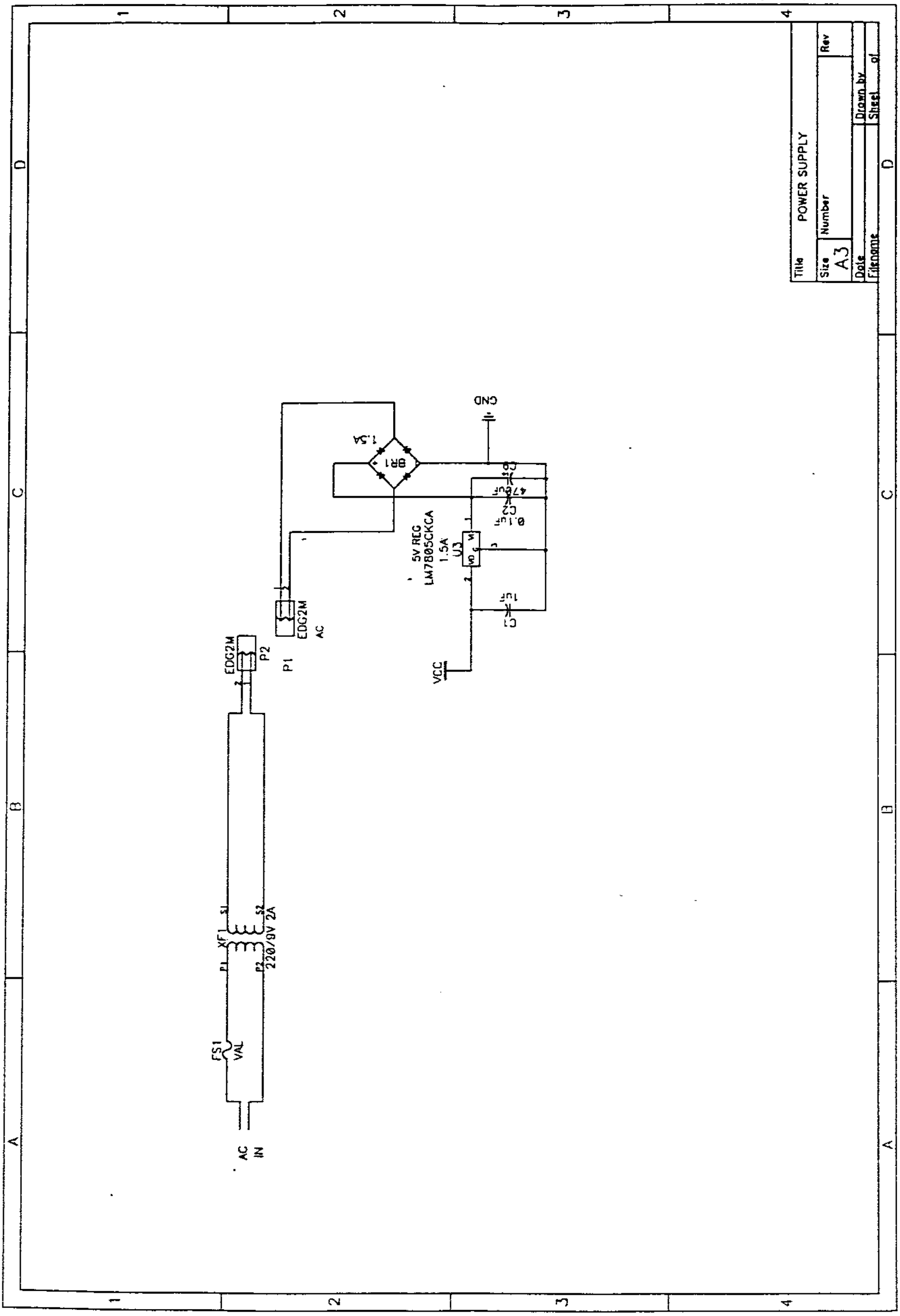
Schematic drawings of the Programming Unit and Viewing Controller.

A2



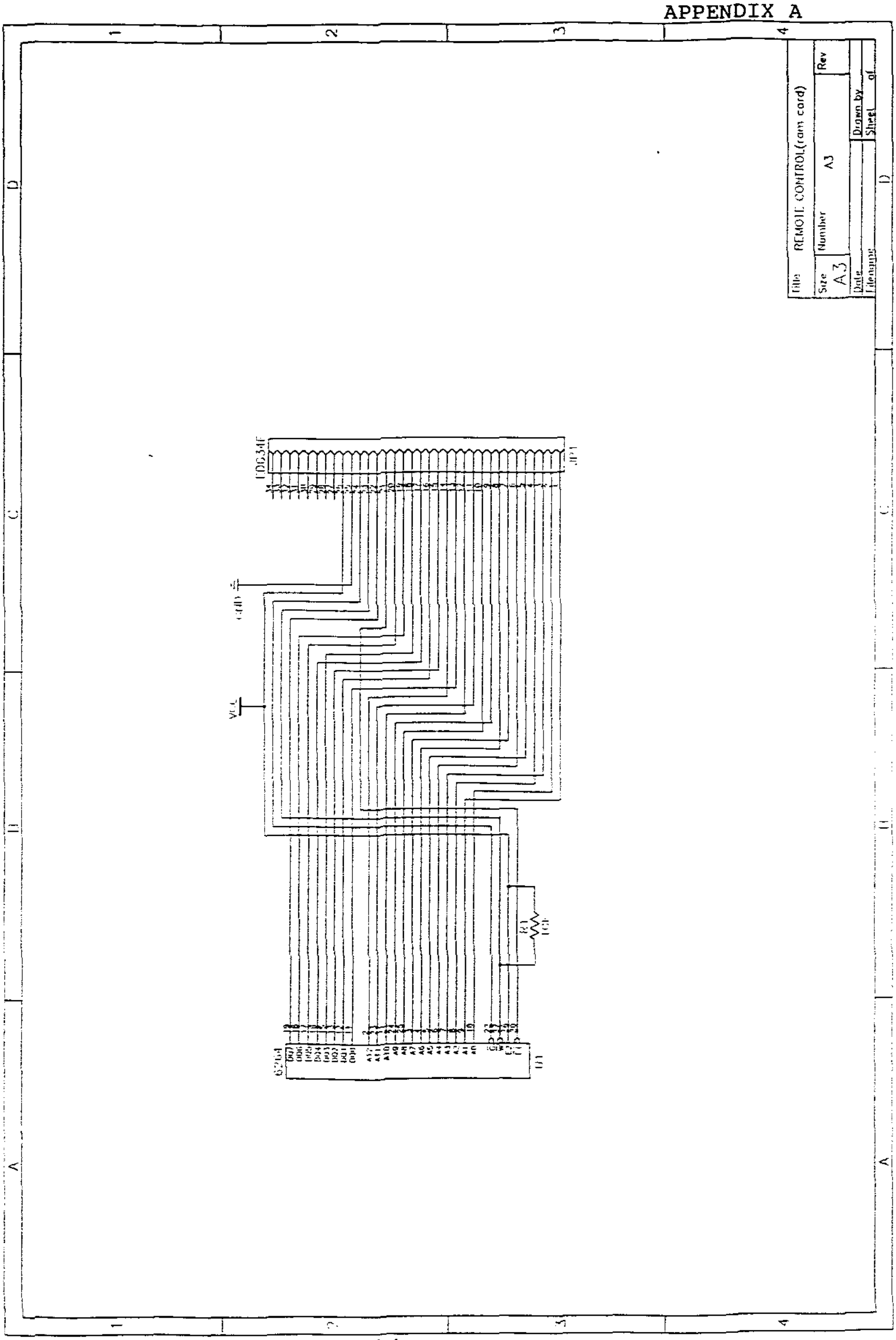
APPENDIX A

APPENDIX A



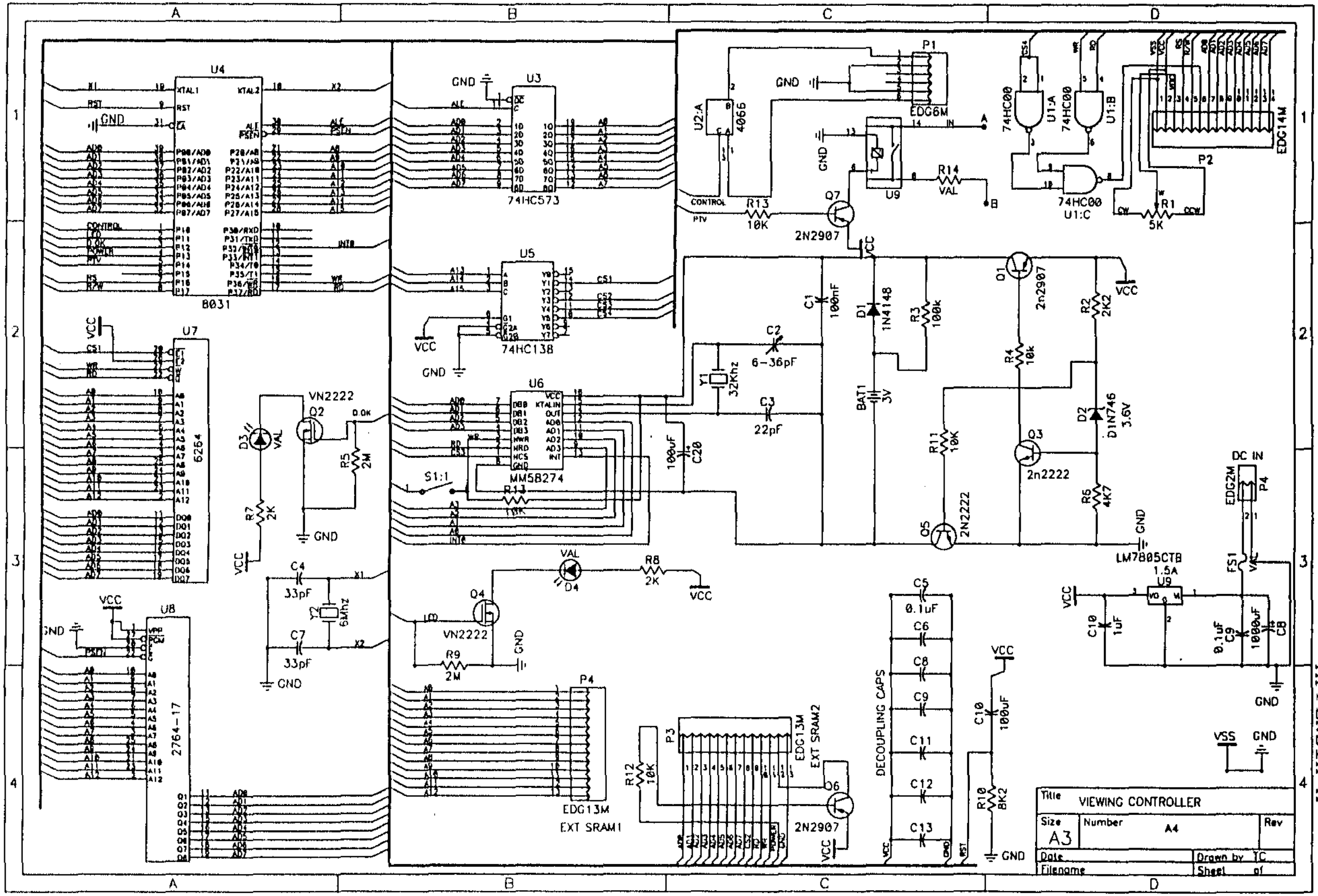
Title POWER SUPPLY
Size A3
Number
Rev
Date
Drawn by
Filename
Sheet of
GND
BR1
1.5A
C2
470uF
0.1uF
5V REG
LM7805CKCA
1.5A
U3
1uF
VCC
EDG2M
AC
P1
EDG2M
P2
XF1
220/9V 2A
FS1
VAL
AC
IN

APPENDIX A

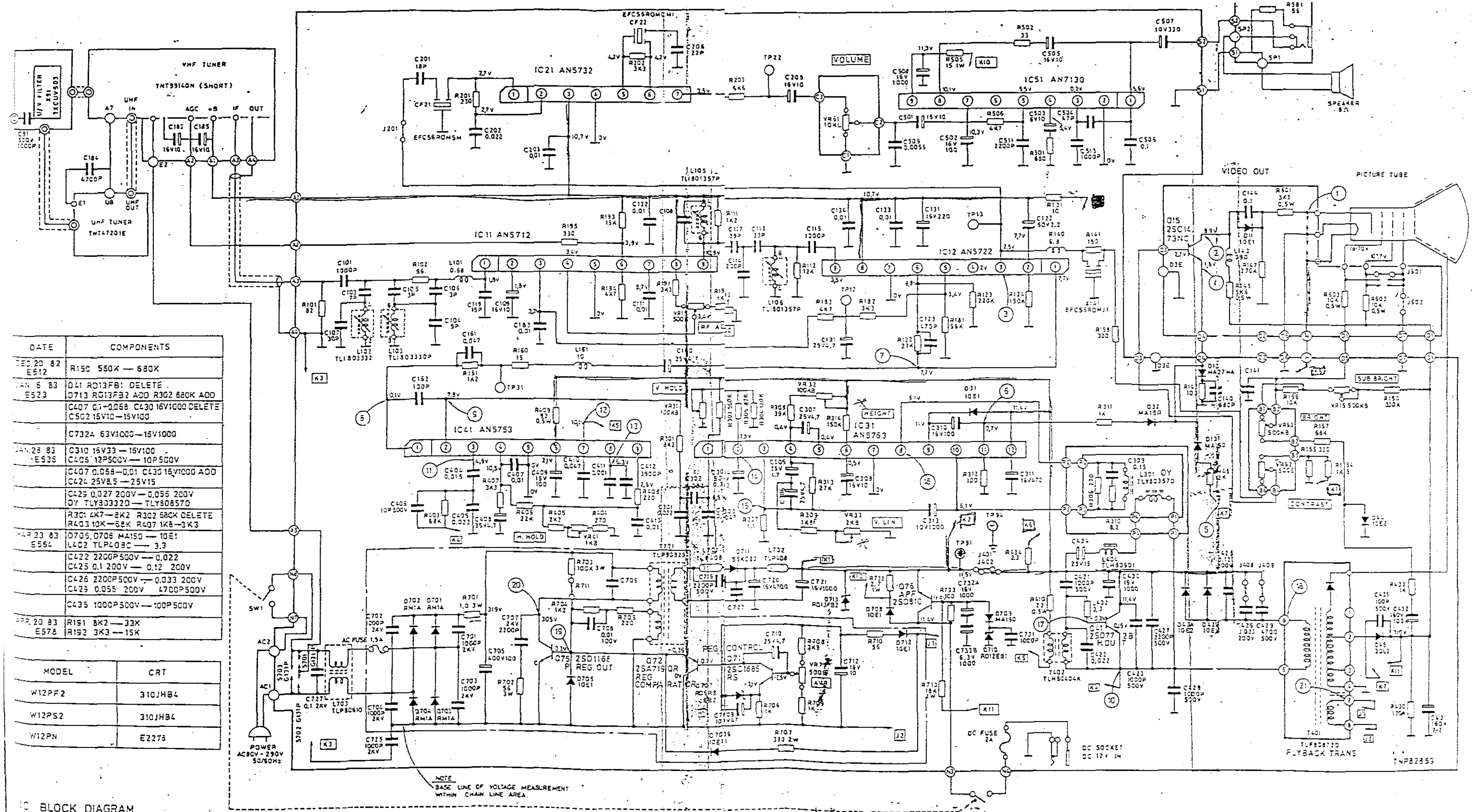


Title
REMOTE CONTROL(ram card)
Size
A3
Number
Rev
Drawn by
Sheet
of
Date
Filename
VCC
GND
JP1
U1
6264
R1

APPENDIX A



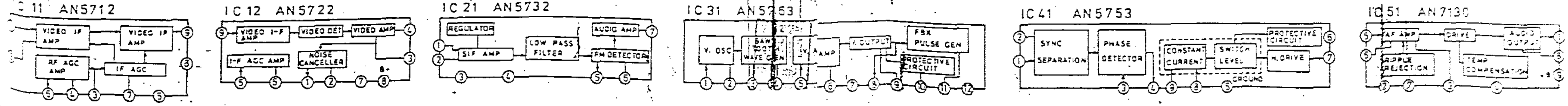
VIEWING CONTROLLER
Size A3
Number A4
Drawn by TC
U4 8031
U3 74HC573
U5 74HC138
U6 MM58274
U7 6264
U8 2764-17
U2:A 4066
U9
U1:A 74HC00
U1:B 74HC00
U1:C 74HC00
P1 EDG6M
P2 EDG14M
P3 EDG13M EXT SRAM2
P4 EDG13M EXT SRAM1
P4 EDG2M DC IN
Q1 2n2907
Q2 VN2222
Q3 2n2222
Q4 VN2222
Q5 2N2222
Q6 2N2907
Q7 2N2907
D1 1N4148
D2 D1N746 3.6V
D3 VAL
D4 VAL
R1 5K
R2 2K2
R3 100k
R4 10k
R5 2M
R6 4K7
R7 2K
R8 2K
R9 2M
R10 8K2
R11 10K
R12 10K
R13 10K
R14 VAL
C1 100nF
C2 6-36pF
C3 22pF
C4 33pF
C7 33pF
C10 1uF
C9 0.1uF
C8 1000uF
C20 100uF
C10 100uF
C5 0.1uF
DECOUPLING CAPS
Y1 32Khz
Y2 6Mhz
BAT1 3V
S1:1
FS1 1.5A
U9 LM7805CTB
VCC
GND
VSS



DATE	COMPONENTS
DEC.20 82 E512	R150 560K — 680K
JAN. 6 83 E523	D41 RO13FB1 DELETE. D713 RO13FB2 ADD R302 680K ADD
	C407 0.1→0.068 C430 16V1000 DELETE C502 16V10—16V100
	C732A 63V1000—16V1000
JAN.26 83 E535	C310 16V33 — 16V100 C405 12P500V — 10P500V
	C407 0.068—0.01 C430 16V1000 ADD C424 25V8.5 → 25V15
	C429 0.027 200V — 0.056 200V DY TLY303320 — TLY308570
	R301 4K7—8K2 R302 680K DELETE R403 10K—68K R407 1K8→3K3
MAR 23 83 E554	D705, D706 MA150 — 10E1 L402 TLP408C — 3.3
	C422 2200P500V — 0.022 C425 0.1 200V — 0.12 200V
	C426 2200P500V — 0.033 200V C429 0.056 200V 4700P500V
	C435 1000P500V — 100P500V
APR. 20 83 E578	R191 8K2 — 33K R192 3K3 — 15K

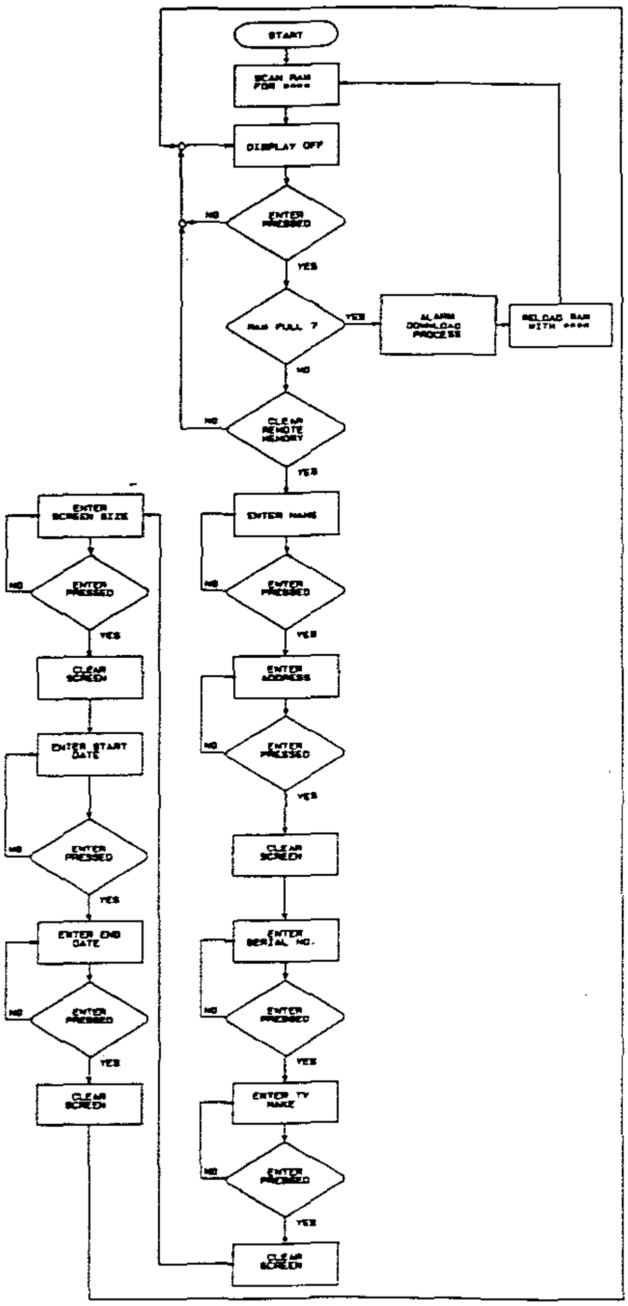
MODEL	CRT
W12PF2	310JHB4
W12PS2	310JHB4
W12PN	E2275

IC BLOCK DIAGRAM



APPENDIX B

Flow chart and programme listing of Programming Unit.


START
DISPLAY OFF
ENTER PRESSED
NO
YES
RAM FULL ?
ALARM DOWNLOAD PROCESS
CLEAR REMOTE MEMORY
ENTER NAME
ENTER ADDRESS
CLEAR SCREEN
ENTER SERIAL NO.
ENTER TV MAKE
ENTER SCREEN SIZE
ENTER START DATE
ENTER END DATE

```
;TITLE          PREVENTION OF PIRATE TELEVISION
;SUBTTL         PROGRAMMER
;--------------------BSEG--------------------------------------

RS          BIT     P1.0
RD_WR       BIT     P1.1
LED         BIT     P1.2
BUZZ        BIT     P1.3
POWER       BIT     P1.4

EPROM_H     DATA    30H
EPROM_L     DATA    31H
RAM_H       DATA    32H
RAM_L       DATA    33H
XRAM_H      DATA    34H
XRAM_L      DATA    35H
EVENT       DATA    36H                 ;REPITITION COUNTER
COUNTER     DATA    37H
TRACK       DATA    38H                 ;TRACK CURSOR
;--------------------ENABLE INTERUPTS----------------------------

                     CSEG
            ORG     00H

            LJMP    INIT
            ORG     03H                 ;VECTOR ADD INT0
            LJMP    KEY

            ORG     30H

INIT:       SETB    EA                  ;ENABLE INTERUPTS
            SETB    EX0                 ;ENABLE INT0
            CLR     P1.2                ;BUZZER
            SETB    P1.4                ;POWER OFF
;------------------INITIALIZE DISPLAY----------------------------

            CLR     RS
            CLR     RD_WR

            MOV     DPTR,#0A000H
            MOV     A,#01H              ;CLEAR DISPLAY
            MOVX    @DPTR,A
            LCALL   DELAY

            MOV     A,#02H              ;CURSOR HOME
            MOVX    @DPTR,A
            LCALL   DELAY

            MOV     A,#38H              ;FUNCTION SET
            MOVX    @DPTR,A
            LCALL   DELAY

            MOV     A,#06H              ;ENTRY MODE
            MOVX    @DPTR,A
            LCALL   DELAY

            MOV     A,#0EH              ;DISPLAY ON\CURSOR ON
```

```
            MOVX      @DPTR,A
            LCALL     DELAY
;-----------------INITIALIZE KEYBOARD--------------------------

            MOV       DPTR,#8001H
            MOV       A,#02H              ;MODE
            MOVX      @DPTR,A

            MOV       A,#34H              ;PROG CLOCK SPEED
            MOVX      @DPTR,A


;                     MAIN PROGRAM
;                     -------------

BASE:       MOV       RAM_H,#20H          ;START ADDRESS
            MOV       RAM_L,#00
            LCALL     SCAN                ;SCAN ASTERISK

BEGIN:      LCALL     CURSOR_OFF
            MOV       A,R7                ;R7=0DH ?
            MOV       R7,#00H
            CJNE      A,#0DH,BEGIN        ;'ENTER' SCREEN ON
            MOV       R7,#00H

            LCALL     CURSOR_ON           ;DISPLAY ON

            MOV       DPH,RAM_H           ;FETCH LATEST RAM POS
            MOV       DPL,RAM_L           ;TO CHECK RAM FULL
            MOV       A,RAM_H
            CJNE      A,3FH,MEM_OK        ;CHECK HIGH
            LCALL     MEM_FULL            ;DOWNLOADING PROCESS
            LCALL     BLANK_LCD
            LJMP      BASE

MEM_OK:     MOV       RAM_H,DPH           ;RESTORING LATEST RAM
            MOV       RAM_L,DPL           ;POSITION

            LCALL     BLANK_LCD
            MOV       DPTR,#CLR_MEM       ;'CLEAR REMOTE MEMORY'
            LCALL     DISP_MSG

            MOV       A,#40H
            LCALL     CURSOR
            MOV       DPTR,#CLR_MEM1      ;'YES/NO - Y/N'
RRR:        LCALL     DISP_MSG

            MOV       A,R7                ;GET KEY VAL
            MOV       R7,#00H
            CJNE      A,#59H,DONE         ;YES
            LCALL     DISP_CHAR

BACK:       MOV       A,R7
            CJNE      A,#0DH,BACK         ;CR?
            LCALL     DELAY1

            CLR       P1.4                ;POWER ON REMOTE
```

```
            LCALL   DELAY
            LJMP    DDD                 ;CLR MEMORY

DONE:       CJNE    A,#4EH,RRR          ;NO
            LCALL   DISP_CHAR
            LCALL   DELAY1
            LCALL   DELAY1
            LCALL   DELAY1
            LJMP    FIN                 ;QUIT

DDD:        MOV     R7,#00H
            LCALL   DELAY
            LCALL   CLR_MEMORY          ;CLEAR REMOTE MEMORY
            LCALL   BLANK_LCD           ;CLEAR DISPLAY

            MOV     R0,#00H
            MOV     TRACK,#05H          ;TRACK CURSOR
            MOV     DPTR,#PROMPT        ;'** NAME: **'
            LCALL   DISP_MSG
            LCALL   DISPLAY

            MOV     A,#40H
            LCALL   CURSOR
            MOV     TRACK,#44H
            MOV     DPTR,#ADDRESS       ;'**ADDRESS: **'
            LCALL   DISP_MSG
            LCALL   DISPLAY

            LCALL   BLANK_LCD           ;CLEAR DISPLAY

            MOV     XRAM_H,#60H         ;REMOTE CONTROL ADDRESS
            MOV     XRAM_L,#00H

            MOV     TRACK,#0CH          ;TRACK CURSOR
            MOV     DPTR,#SERIAL_NO     ;'** SERIAL NO ** '
            LCALL   DISP_MSG

            MOV     A,#0CH              ;DISPLAY POSITION
            LCALL   CURSOR
            LCALL   DISPLAY2

            MOV     A,#40H
            LCALL   CURSOR              ;CURSOR ON 2ND LINE
            MOV     TRACK,#4CH
            MOV     DPTR,#TV            ;' ** TV MAKE ** '
            LCALL   DISP_MSG

            MOV     A,#4CH
            LCALL   CURSOR
            LCALL   DISPLAY2

            LCALL   BLANK_LCD
            MOV     A,#00H
            LCALL   CURSOR              ;CURSOR ON 1ST LINE

            MOV     TRACK,#0CH
            MOV     DPTR,#SIZE          ;' ** SCREEN SIZE ** '
```

```
            LCALL   DISP_MSG

            MOV     A,#0CH
            LCALL   CURSOR
            LCALL   DISPLAY2

            LCALL   BLANK_LCD
            MOV     A,#00H
            LCALL   CURSOR                  ;CURSOR ON 1ST LINE

            MOV     TRACK,#0CH
            MOV     DPTR,#ST_DATE           ;' ** START DATE : YY-MM-DD
            ** '
            LCALL   DISP_MSG
            MOV     A,#0CH
            LCALL   CURSOR
            LCALL   DISPLAY2

            MOV     A,#40H
            LCALL   CURSOR                  ;CURSOR ON 2ND LINE

            MOV     TRACK,#4CH
            MOV     DPTR,#END_DATE          ;' ** END DATE : YY-MM-DD **
            '
            LCALL   DISP_MSG

            MOV     A,#4CH
            LCALL   CURSOR
            LCALL   DISPLAY2

            SETB    P1.4                    ;POWER OFF REMOTE
;       *** AT THIS MOMENT ONLY THE NAME AND ADDRESS IS IN RAM ***

  FIN:      LCALL   BLANK_LCD
            LJMP    BEGIN

;-------------------------SUBROUTINES START------------------------
;-------------------------SCAN ROUTINE,FIND LATEST ADDRESS---------

SCAN:       MOV     DPH,RAM_H               ;BASE ADDRESS
            MOV     DPL,RAM_L
SCAN_MORE:  MOVX    A,@DPTR
            CJNE    A,#2AH,SCAN_ON          ;ASTERISK
            LJMP    FOUND
SCAN_ON:    INC     DPTR
            LJMP    SCAN_MORE
FOUND:      MOV     RAM_H,DPH               ;SAVE ADD WHERE ASTERISK
            MOV     RAM_L,DPL               ;     FOUND
            RET
;-------------------------DISPLAY AND SAVE TO RAM-----------------

DISPLAY:    MOV     A,R7
            MOV     R7,#00H
            CJNE    A,#7FH,SCREEN           ;BACKSPACE,DELETE MEM ADDRESS
            LCALL   BACKSPACE
SCREEN:     CJNE    A,#0DH,SHOW             ;CR ?
            LJMP    WWW
```

```
SHOW:      CJNE    A,#00H,SHOW1
           LJMP    DISPLAY
SHOW1:     PUSH    ACC
           LCALL   DISP_CHAR
           INC     TRACK              ;TRACK CURSOR POSITION

           MOV     DPH,RAM_H          ;RAM ADDRESS
           MOV     DPL,RAM_L
           POP     ACC
           MOVX    @DPTR,A
           INC     DPTR
           MOV     RAM_H,DPH
           MOV     RAM_L,DPL

           LJMP    DISPLAY
WWW:       MOV     R7,#00H
           RET
;-----------------------BACKSPACE ROUTINE/RECTIFIES MEMORY TOO---

BACKSPACE: DEC     TRACK              ;BACKSPACE CURSOR ON
           MOV     A,TRACK            ;   DISPLAY
           LCALL   CURSOR
           MOV     A,#20H             ;DELETE CHAR.
           LCALL   DISP_CHAR

           MOV     A,TRACK            ;DELETED PREVIOUS CHAR.
           LCALL   CURSOR             ;RESET CURSOR CORRECT
           POSITION
           MOV     DPH,RAM_H
           MOV     A,DPH
           CJNE    A,#60H,RAM_MEM     ;RAM OR XRAM?
           LJMP    XRAM_MEM

RAM_MEM:   MOV     DPL,RAM_L          ;DEC MEMORY ADDRESS
           DEC     DPL
           MOV     RAM_L,DPL
           MOV     A,#00H
           RET

BACSPACE:  DEC     TRACK
           MOV     A,TRACK
           LCALL   CURSOR
           MOV     A,#20H
           LCALL   DISP_CHAR
           MOV     A,TRACK
           LCALL   CURSOR
           MOV     DPH,XRAM_H
           MOV     A,DPH
           CJNE    A,#60H,DON
XRAM_MEM:  MOV     DPH,XRAM_H          ;REMOTE MEMORY
           MOV     DPL,XRAM_L
           DEC     DPL
           MOV     XRAM_L,DPL
DON:       MOV     A,#00H
           RET
```

```
;-----------------------DISPLAY AND SAVE TO REMOTE CONTROL-------

DISPLAY2:   MOV     A,R7
            MOV     R7,#00H
            CJNE    A,#7FH,VIEWER
            LCALL   BACSPACE           ;BACKSPACE ROUTINE - XRAM

VIEWER:     CJNE    A,#0DH,VIEW
            LJMP    OUTT
VIEW:       CJNE    A,#00H,VIEW1
            LJMP    DISPLAY2

VIEW1:      PUSH    ACC
            LCALL   DISP_CHAR
            INC     TRACK

            MOV     DPH,XRAM_H
            MOV     DPL,XRAM_L
            POP     ACC
            CJNE    A,#3AH,PASS       ;CHECK FOR COLON
            LJMP    DISPLAY2          ;  NO SAVE

PASS:       MOVX    @DPTR,A           ;SAVE LETTER/NO
            INC     DPTR
            MOV     XRAM_L,DPL

            LJMP    DISPLAY2
OUTT:       MOV     R7,#00H
            RET

;-----------------------CLEAR DISPLAY---------------------------

BLANK_LCD:  PUSH    DPH
            PUSH    DPL
            CLR     RS
            CLR     RD_WR

            MOV     A,#01H            ;CLEAR DISPLAY
            MOV     DPTR,#0A000H
            MOVX    @DPTR,A
            LCALL   DELAY
            POP     DPL
            POP     DPH
            RET

;------------------------DISPLAY MESSAGE ROUTINE----------------

DISP_MSG:   SETB    RS
 NEXT_CH:   CLR     A
            MOVC    A,@A+DPTR         ;GET CHAR
            JZ      EXIT
            INC     DPTR
            PUSH    DPH
            PUSH    DPL
            MOV     DPTR,#0A000H      ;CS5
            MOVX    @DPTR,A           ;DISPLAY CHAR
            LCALL   DELAY
```

```
            POP     DPL
            POP     DPH
            LJMP    NEXT_CH              ;NEXT CHARACTER
   EXIT:    RET
 ;----------------------MOVE CURSOR------------------------------

CURSOR:     CLR     RS
            CLR     RD_WR
            ADD     A,#80H               ;POINT CURSOR TO FIRST\SECOND
            MOV     DPTR,#0A000H         ;         ROW
            MOVX    @DPTR,A              ;DISPLAY LINE
            LCALL   DELAY
            RET
;-------------------------KEY LOOKUP---------------------------

KEY:        MOV     A,#40H               ;SETUP READ RAM
            MOV     DPTR,#8001H
            MOVX    @DPTR,A

            MOV     DPTR,#8000H
            MOVX    A,@DPTR              ;READ RAM
            ANL     A,#3FH               ;RECTIFIES,GET CORRECT KEY
            ADD.
            MOV     DPTR,#KEY_LOOKUP
            MOVC    A,@A+DPTR            ;FETCH KEY
            MOV     R7,A                 ;SAVE LETTER

            MOV     A,#0F0H              ;END OF INTERUPT
            MOV     DPTR,#8001H
            MOVX    @DPTR,A
            MOV     A,#00H
            RETI
;----------------------SWITCH DISPLAY OFF----------------------

CURSOR_OFF: PUSH    DPH
            PUSH    DPL
            CLR     RS
            CLR     RD_WR
            MOV     A,#08H
            MOV     DPTR,#0A000H
            MOVX    @DPTR,A
            LCALL   DELAY
            POP     DPL
            POP     DPH
            RET
;----------------------SWITCH DISPLAY ON------------------------

CURSOR_ON:  PUSH    DPH
            PUSH    DPL
            CLR     RS
            CLR     RD_WR
            MOV     A,#0EH
            MOV     DPTR,#0A000H
            MOVX    @DPTR,A
            LCALL   DELAY
            POP     DPL
            POP     DPH
```

```
            RET
;----------------------CLEAR REMOTE MEMORY------------------------

CLR_MEMORY: MOV     DPTR,#6000H
      CONT: MOV     A,#2AH              ;********
            MOVX    @DPTR,A
            INC     DPTR
            MOV     A,DPH
            CJNE    A,#7FH,CONT         ;CHECK H-ADDRESS
            MOV     A,DPL
            CJNE    A,#40H,CONT
            LCALL   DELAY
            RET

;----------------------CLEAR RAM MEMORY---------------------------

CLEAN_UP:   MOV     DPTR,#2000H
    GAIN:   MOV     A,#2AH              ;ASTERISK
            MOVX    @DPTR,A
            INC     DPTR
            MOV     A,DPH
            CJNE    A,#3FH,GAIN
            RET
;----------------------DELAY ROUTINE------------------------------

DELAY:      PUSH    ACC
            MOV     EVENT,#013H
RELOAD:     MOV     A,#0FFH
AGAIN:      DEC     A
            JNZ     AGAIN
            DJNZ    EVENT,RELOAD
            POP     ACC
            RET

DELAY1:     PUSH    ACC
            MOV     COUNTER,#0FFH
RELOAD2:    MOV     EVENT,#0FFH
RELOAD1:    MOV     A,#0FFH
AGAIN1:     DEC     A
            JNZ     AGAIN1
            DJNZ    EVENT,RELOAD1
            DEC     COUNTER
            JNZ     RELOAD2
            POP     ACC
            RET

;--------------------- MEMORY FULL--------------------------------

MEM_FULL:   MOV     DPTR,#PROMPT2       ;'MEMORY FULL'
            LCALL   DISP_MSG

            MOV     R3,#014H
RING:       SETB    BUZZ                ;ON
            SETB    LED
            LCALL   DELAY1
            LCALL   DELAY1
            CLR     BUZZ
```

```
            CLR     LED                 ;OFF
            LCALL   DELAY1
            LCALL   DELAY1
            DJNZ    R3,RING
            SETB    BUZZ                ; CLR     BUZZ

            MOV     A,#40H
            LCALL   CURSOR
            MOV     DPTR,#MESS4         ;PRESS ENTER TO CONTINUE
            LCALL   DISP_MSG

 POST:      LCALL   DELAY
            MOV     A,R7
            CJNE    A,#0DH,POST
            MOV     R7,#00H
            LCALL   DELAY1
            LCALL   PC_COMM
            SETB    P1.4                ;POWER OFF REMOTE
            LCALL   DELAY1
            RET
;--------------------PC COMMUNICATION---------------------------

 PC_COMM:   LCALL   BLANK_LCD
            LCALL   DELAY1
            MOV     DPTR,#MESS1         ;PLUG CONTROL IN
            LCALL   DISP_MSG

            MOV     A,#40H
            LCALL   CURSOR
            MOV     DPTR,#MESS4         ;ENTER TO CONT.
            LCALL   DISP_MSG

 PLUG:      LCALL   DELAY
            MOV     A,R7
            CJNE    A,#0DH,PLUG

            LCALL   BLANK_LCD
            CLR     P1.4               ;REMOTE POWER ON
            LCALL   DELAY1
            MOV     DPTR,#MESS2         ;PC COMMUNICATIONS
            LCALL   DISP_MSG

            MOV     RAM_H,#20H          ;RAM,INITIALIZE BASE ADD
            MOV     RAM_L,#00H

            MOV     XRAM_H,#60H         ;XRAM,INITIALIZE BASE ADD
            MOV     XRAM_L,#00H

DOWN:       MOV     R5,#00H
            MOV     DPH,RAM_H           ;FETCH BASE ADDRESS
            MOV     DPL,RAM_L
            MOVX    A,@DPTR
            MOV     R5,A                ;TEMP STORE
            INC     DPTR
            MOV     RAM_H,DPH           ;STORE NEW ADDRESS
            MOV     RAM_L,DPL
            MOV     DPH,RAM_H
```

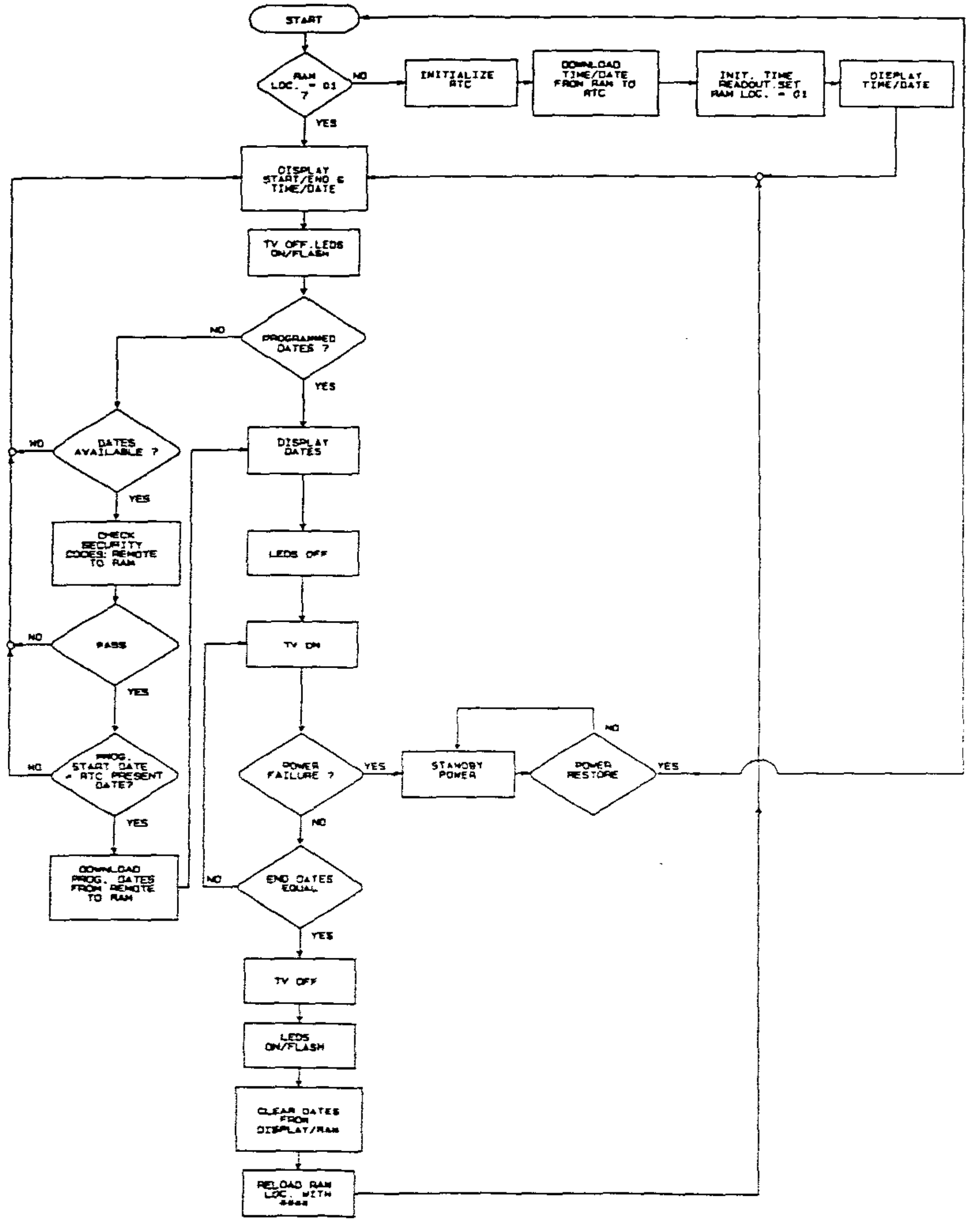
```
            MOV     A,DPH
            CJNE    A,#3FH,SEND         ;CHECK H ADDRESS
            LJMP    ZZZ

 SEND:      MOV     DPH,XRAM_H
            MOV     DPL,XRAM_L          ;FETCH XRAM BASE ADDRESS
            MOV     A,R5
            MOVX    @DPTR,A
            INC     DPTR
            MOV     XRAM_H,DPH          ;STORE NEW ADD
            MOV     XRAM_L,DPL
            LJMP    DOWN                ;NEXT

ZZZ:        LCALL   CLEAN_UP            ;CLEAR RAM
            LCALL   BLANK_LCD
            MOV     R7,#00H

            MOV     DPTR,#MESS3         ;DOWNLOAD COMPLETED
            LCALL   DISP_MSG

            MOV     A,#40H
            LCALL   CURSOR
            MOV     DPTR,#MESS4         ;ENTER TO CONTINUE
            LCALL   DISP_MSG
WAIT:       LCALL   DELAY
            MOV     A,R7
            CJNE    A,#0DH,WAIT
            RET

;------------------------DISPLAY CHARACTER------------------------

DISP_CHAR:  SETB    RS
            CLR     RD_WR
            PUSH    DPH
            PUSH    DPL
            MOV     DPTR,#0A000H
            MOVX    @DPTR,A
            LCALL   DELAY
            POP     DPL
            POP     DPH
            RET
;-----------------------------------------------------------------

CLR_MEM:    DB      '* CLEAR REMOTE CONTROL MEMORY *',00H
CLR_MEM1:   DB      '          YES/NO - Y/N  ',00H
PROMPT:     DB      'NAME:',00H
SIZE:       DB      'SCREEN SIZE:XXXX',00H
ADDRESS:    DB      'ADD:',00H
ST_DATE:    DB      'START DATE :YY:MM:DD',00H
END_DATE:   DB      'END DATE   :YY:MM:DD',00H
SERIAL_NO:  DB      'SERIAL NO  :XXX XXX XXX X',00H
TV:         DB      'TV MAKE    :XXXXXXXXXX',00H
PROMPT2:    DB      'MEMORY FULL: ** DOWNLOAD INFO **',00H
MESS1:      DB      '*PLEASE PLUG MASTER CONTROL IN*',00H
MESS2:      DB      'PLEASE WAIT - PC COMMUNICATIONS',00H
MESS3:      DB      '    *** DOWNLOAD COMPLETED ***',00H
MESS4:      DB      '*****PRESS ENTER TO CONTINUE*****',00H
```

```
KEY_LOOKUP: DB        41H             ;A
            DB        42H             ;B
            DB        43H             ;C
            DB        44H             ;D
            DB        45H             ;E
            DB        00H
            DB        00H
            DB        00H
            DB        46H             ;F
            DB        47H             ;G
            DB        48H             ;H
            DB        49H             ;I
            DB        4AH             ;J
            DB        00H
            DB        00H
            DB        00H
            DB        4BH             ;K
            DB        4CH             ;L
            DB        4DH             ;M
            DB        4EH             ;N
            DB        4FH             ;O
            DB        00H
            DB        00H
            DB        00H
            DB        50H             ;P
            DB        51H             ;Q
            DB        52H             ;R
            DB        53H             ;S
            DB        54H             ;T
            DB        00H
            DB        00H
            DB        00H
            DB        55H             ;U
            DB        56H             ;V
            DB        57H             ;W
            DB        58H             ;X
            DB        59H             ;Y
            DB        00H
            DB        00H
            DB        00H
            DB        5AH             ;Z
            DB        30H             ;0
            DB        31H             ;1
            DB        32H             ;2
            DB        33H             ;3
            DB        00H
            DB        00H
            DB        00H
            DB        34H             ;4
            DB        35H             ;5
            DB        36H             ;6
            DB        37H             ;7
            DB        38H             ;8
            DB        00H
            DB        00H
            DB        00H
            DB        39H             ;9
```

```
DB      7FH         ;DELETE
DB      3AH         ;COLON
DB      20H         ;FORWARDSPACE
DB      0DH         ;CR
END
```

APPENDIX C

Flow chart and programme listing of the Viewing Controller.


START
RAM LOC. = 01 ?
NO
YES
INITIALIZE RTC
DOWNLOAD TIME/DATE FROM RAM TO RTC
INIT. TIME READOUT.SET RAM LOC. = 01
DISPLAY TIME/DATE
DISPLAY START/END & TIME/DATE
TV OFF.LEDS ON/FLASH
PROGRAMMED DATES ?
DATES AVAILABLE ?
CHECK SECURITY CODES: REMOTE TO RAM
PASS
PROG. START DATE = RTC PRESENT DATE?
DOWNLOAD PROG. DATES FROM REMOTE TO RAM
DISPLAY DATES
LEDS OFF
TV ON
POWER FAILURE ?
STANDBY POWER
POWER RESTORE
END DATES EQUAL
TV OFF
LEDS ON/FLASH
CLEAR DATES FROM DISPLAY/RAM
RELOAD RAM LOC. WITH

```
;TITLE          PREVENTION OF PIRATE TV\PAY TV
;SUBTITLE       REMOTE CONTROL VIEWING
;------------------------------------------------------------------
                                    BSEG

RS          BIT     P1.6
RD_WR       BIT     P1.7
ON_OFF      BIT     P1.0            ;TV CONTROL
LPI         BIT     P1.1            ;LICENCE DUE
DATES_OK    BIT     P1.2            ;LOAD DATES
POWER       BIT     P1.3            ;POWER CONTROL,REMOTE
PTV         BIT     P1.4            ; "         "       TV
DATE_H      DATA    30H             ;TEMP RAM STORES
DATE_L      DATA    31H             ; "     "     "
BUFFER      DATA    32H             ;DELAY COUNTER
CLK_H       DATA    33H             ;REAL TIME CLK REG
CLK_L       DATA    34H             ;"      "    "    "
EPROM_H     DATA    35H             ;EPROM POINTER
EPROM_L     DATA    36H             ; "        "
RAM_H       DATA    37H             ;RAM POINTER
RAM_L       DATA    38H             ; "        "
XRAM_H      DATA    39H             ;EXT RAM POINTER
XRAM_L      DATA    3AH             ;  "   "     "
COUNTER     DATA    3BH             ;DIGITS COUNTER
TRACK       DATA    3CH             ;TRACK CURSOR
D_COUNT     DATA    3DH             ;DIGITS COUNTER
COUNT       DATA    3EH             ;REDISPLAY PURPOSE

;----------------------------ENABLE INTERUPTS---------------------
                              CSEG
            ORG     00H
            LJMP    START

            ORG     03H             ;VECTOR ADD INT0
            LJMP    TIME_OUT

            ORG     30H

START:      SETB    EA              ;ENABLE INTERRUPTS
            SETB    EX0             ;ENABLE INT0
            SETB    P3.2

;----------------------------INITIALIZE DISPLAY------------------

            CLR     RS
            CLR     RD_WR

            MOV     DPTR,#0A000H
            MOV     A,#01H          ;CLEAR DISPLAY
            MOVX    @DPTR,A
            LCALL   DELAY

            MOV     A,#02H          ;CURSOR HOME
            MOVX    @DPTR,A
            LCALL   DELAY

            MOV     A,#38H          ;FUNCTION SET
```

```
          MOVX      @DPTR,A
          LCALL     DELAY

          MOV       A,#06H              ;ENTRY MODE
          MOVX      @DPTR,A
          LCALL     DELAY

          MOV       A,#0CH              ;DISPLAY ON/CURSOR OFF
          MOVX      @DPTR,A
          LCALL     DELAY

          SETB      P1.1                ;'LICENCE DUE' LED ON
          CLR       P1.0                ;SIGNAL OFF
          SETB      P1.4                ;SIGNAL INT.
          CLR       P1.5                ;  "       "
          CLR       P1.2                ;'REQUEST DATES' LOADED INDICATOR
                                        ;             OFF
          SETB      P1.3                ;POWER OFF - REMOTE CONTROL

;----------------------------MAIN PROGRAM-------------------------

          MOV       DPTR,#2020H
          MOVX      A,@DPTR
          CJNE      A,#01H,INT_RTC      ;RTC INITIALIZATION ?
          LJMP      BYPASS              ;BYPASS RTC INITIALIZATION

INT_RTC:  LCALL     INTL_RTC            ;INITIALIZE RTC
          MOV       RAM_H,#20H
          MOV       RAM_L,#13H          ;START ADD. HRS
          LCALL     TIMEDATE            ;SET TIME,DATE,DAY &
          NOP                           ;   DISPLAY TIME
          NOP
          LCALL     MIN_DELAY           ;INITIALIZE 60 SEC DELAY OF RTC

BYPASS:   MOV       R0,#00H

          MOV       A,#00H              ;CURSOR POSITION,1ST LINE
          LCALL     CURSOR
          MOV       DPTR,#START_D       ;'ST'- START DATE
          LCALL     DISP_MSG

          MOV       A,#40H
          LCALL     CURSOR              ;CURSOR ON 2ND LINE
          MOV       DPTR,#END_D         ;'END,- END DATE
          LCALL     DISP_MSG

          MOV       DPH,#20H            ;CHECK RAM FOR PROGRAMMED
          MOV       DPL,#00H            ;           DATES
          MOVX      A,@DPTR
          CJNE      A,#2AH,DO           ;****
          LJMP      BEGIN
DO:       CJNE      A,#39H,DO1          ;93 - 9 OF DATE
          LJMP      DO2
DO1:      CJNE      A,#32H,BEGIN        ;20 - 2 OF DATE

DO2:      MOV       RAM_H,#20H
          MOV       RAM_L,#00H
```

```
        LCALL   NDISPLAY        ;RESTORE DATES TO DISPLAY
        NOP
        NOP
        LJMP    POW_ON          ;     ON POWER RESTORE

BEGIN:  LCALL   STOR_DATE       ;READ RTC & STORE 2100H

        MOV     TRACK,#52H      ;TRACK CURSOR POSITION
        MOV     A,#52H          ;DATE,DAYS POS.
        LCALL   CURSOR
        LCALL   READ_DATE       ;READ 2100 & DISPLAY CURRENT DATE

        LCALL   PLUG_IN         ;PLUG REMOTE IN,20SEC DELAY
        MOV     R1,#04H
        LCALL   SAFE            ;DELAY PERIOD
        CLR     P1.3            ;POWER ON REMOTE
        LCALL   DELAY1

        MOV     XRAM_H,#60H     ;STORE XRAM PTR
        MOV     XRAM_L,#00H

        MOV     RAM_H,#22H      ;PREPROGRAMMED DATA
        MOV     RAM_L,#00H

        MOV     COUNTER,#0DH    ;LENGTH - SERIAL NO
        LCALL   SECURITY
        MOV     A,R6
        CJNE    A,#00H,OUTT
        SJMP    B100
OUTT:   LJMP    FIN1            ;TV OFF

B100:   MOV     COUNTER,#0AH    ;TV MAKE
        LCALL   SECURITY
        MOV     A,R6
        CJNE    A,#00H,F100
        SJMP    B200
F100:   LJMP    FIN1

B200:   MOV     COUNTER,#04H    ;SCREEN SIZE
        LCALL   SECURITY
        MOV     A,R6
        CJNE    A,#00H,F200
        SJMP    B300
F200:   LJMP    FIN1

B300:   LCALL   STOR_DATE        ;READ & STORE RTC DATE RAM 2100H

        MOV     RAM_H,#21H
        MOV     RAM_L,#00H

        MOV     COUNTER,#06H
PLUS:   MOV     R7,#00H
        MOV     DPH,XRAM_H       ;RECALLING XRAM PTR
        MOV     DPL,XRAM_L
        MOVX    A,@DPTR          ;START DATE TO ACC
        MOV     R7,A
        INC     DPL
```

```
        MOV     XRAM_L,DPL

        MOV     DPH,RAM_H           ;RECALLING RAM PTR
        MOV     DPL,RAM_L
        MOVX    A,@DPTR
        INC     DPL
        MOV     RAM_L,DPL
        XRL     A,R7                ;CHECKING
        CJNE    A,#00H,F300         ; PROGRAMMED "START DATE"
        DJNZ    COUNTER,PLUS        ;  TO RTC DATE
        SJMP    B400
F300:   LJMP    FIN1

B400:   MOV     DPH,XRAM_H          ;RESTORING XRAM PTR TO
        MOV     DPL,XRAM_L          ;  BEGINNING OF
        MOV     COUNTER,#06H        ;    "START DATE"
A100:   DEC     DPL
        DJNZ    COUNTER,A100
        MOV     XRAM_H,DPH
        MOV     XRAM_L,DPL

        MOV     RAM_H,#20H          ;ADDRESS,"START DATE"
        MOV     RAM_L,#00H

        MOV     TRACK,#03H          ;TRACK CURSOR
        MOV     A,#03H
        LCALL   CURSOR              ;START DATE DISPLAY POS.

        LCALL   TRANSFER            ;XFER START DATE FROM XRAM
                                    ; TO RAM AND DISPLAY
        MOV     TRACK,#44H
        MOV     A,#44H
        LCALL   CURSOR              ;END DATE DISPLAY POS.

        LCALL   TRANSFER            ;XFER END DATE,XRAM TO RAM
POW_ON: LCALL   DELAY

        SETB    P1.3                ;POWER OFF REMOTE
        SETB    P1.2                ;LED ON - 'DATES' LOADED
        MOV     R1,#10H
        LCALL   SAFE                ;LONG DELAY,REMOVE REMOTE
        CLR     P1.2                ;OFF

RESET1: MOV     COUNTER,#06H
        MOV     DPH,#20H
        MOV     DPL,#0CH
A400:   DEC     DPL                 ;RESTORING RAM PTR TO
        DJNZ    COUNTER,A400        ;BEGINNING OF "END DATE"
        MOV     RAM_L,DPL

VIEW:   SETB    P1.0                ;TV_ON
        CLR     P1.4                ;TV POWER ON
        SETB    P1.5
        CLR     P1.1                ;'LICENSE DUE'LED OFF
        LCALL   DELAY

        LCALL   STOR_DATE           ;READ RTC DATE,STORE - RAM 2100H
```

```
        MOV     TRACK,#52H
        MOV     A,#52H
        LCALL   CURSOR
        LCALL   READ_DATE         ;DISPLAY 2100H,CURRENT RTC DATE

        MOV     B,#00H            ;ZEROES COUNTER
        MOV     COUNTER,#06H      ;DATE LENGTH

        MOV     DATE_H,#21H       ;STORING RTC
        MOV     DATE_L,#00H       ; DATE PTR

        CLR     A

LOOK:   MOV     R7,#00H
        MOV     B,A
        MOV     DPH,#20H          ;RECALL RAM PTR AT BEGINNING
        MOV     DPL,RAM_L         ;OF PROG. " END DATE "
        MOVX    A,@DPTR
        MOV     R7,A
        INC     DPL
        MOV     RAM_L,DPL

        MOV     DPH,DATE_H        ;RECALL STORED RTC DATE
        MOV     DPL,DATE_L
        MOVX    A,@DPTR
        INC     DPL
        MOV     DATE_L,DPL
        XRL     A,R7              ;COMPARING PROG. "END DATE" TO
        CJNE    A,#00H,Q1         ;      RTC DATE
        DEC     COUNTER
        INC     B
        MOV     A,B
        CJNE    A,#06H,LOOK       ;EQUAL DATES

FIN:    CLR     P1.0              ;TV_OFF
        SETB    P1.4              ;TV_POWER OFF
        CLR     P1.5
        SETB    P1.1              ;LED 'LICENSE DUE' ON

        MOV     TRACK,#03H
        MOV     A,#03H
        LCALL   CURSOR
        LCALL   DELETE            ;DELETE START DATE

        MOV     TRACK,#44H
        MOV     A,#44H
        LCALL   CURSOR
        LCALL   DELETE            ;DELETE END DATE
        LCALL   CLEAN             ;CLEAR RAM DATES
FIN1:   SETB    P1.3              ;REMOTE POWER OFF
        LCALL   DELAY
        LJMP    BEGIN

Q1:     MOV     A,COUNTER
        CJNE    A,#00H,LOOP1
        LJMP    FIN
```

```
LOOP1:   MOV    COUNTER,A
LOOP2:   MOV    DPH,#20H
         MOV    DPL,RAM_L          ;ADVANCING PROG "END DATE" TO
         INC    DPL                ; DAYS(UNITS)
         MOV    RAM_L,DPL
         DJNZ   COUNTER,LOOP2
         LJMP   RESET1

;---------------------------SUBROUTINES START--------------------
;-------------------REAL TIME CLOCK INITIALIZATION---------------

INTL_RTC:MOV    A,#0FH
         MOV    DPTR,#8000H
         MOVX   @DPTR,A            ;15 INTO CONTROL REG

         MOV    A,#00H
         MOV    DPTR,#800FH
         MOVX   @DPTR,A            ;0 INTO INTERRUPT REG
         MOV    A,#05H
         MOV    DPTR,#8000H
         MOVX   @DPTR,A            ;INTIALIZE CLOCK

         MOV    A,#01H
         MOV    DPTR,#800FH
         MOVX   @DPTR,A            ;24HR MODE,LEAP YR

         MOV    A,#00H
         MOV    DPTR,#8000H
         MOVX   @DPTR,A            ;START CLK

         MOV    CLK_H,#80H         ;SET CLK TO TENS OF HRS REG
         MOV    CLK_L,#07H         ;"    "  "   "  "  "  "
         RET
;-------------------INTIALIZE 60 SEC DELAY OF RTC----------------

 MIN_DELAY:MOV    A,#03H           ;STOP TIMING
           MOV    DPTR,#8000H
           MOVX   @DPTR,A

           MOV    A,#0FH           ;INT. TIME DELAY
           MOV    DPTR,#800FH
           MOVX   @DPTR,A

           MOV    A,#00H
           MOV    DPTR,#8000H      ;INT. TIME STARTS
           MOVX   @DPTR,A
           RET
;-------------------RESTORES SCREEN ON POWER RESTORE------------

 NDISPLAY:MOV     COUNT,#02H
          MOV     TRACK,#03H       ;TRACK CURSOR
          MOV     A,#03H           ;CURSOR POS.
          LCALL   CURSOR

NEWD:     MOV     R2,#02H          ;UNITS OF TWO'S COUNTER
          MOV     COUNTER,#06H     ;DIGITS COUNTER
```

```
NSCREEN:  MOV     DPH,RAM_H          ;START ADD.
          MOV     DPL,RAM_L
          MOVX    A,@DPTR
          PUSH    ACC
          INC     DPL
          MOV     RAM_L,DPL

          POP     ACC
          LCALL   DISP_CHAR          ;SCREEN DISPLAY
          INC     TRACK
          DJNZ    R2,Q3
          MOV     A,#20H             ;BLANK FOR DISPLAY FORMAT
          LCALL   DISP_CHAR          ;      YY MM DD
          INC     TRACK
          MOV     R2,#02H
Q3:       DJNZ    COUNTER,NSCREEN

          DJNZ    COUNT,NDISP
          LJMP    REDISP

NDISP:    MOV     TRACK,#44H
          MOV     A,#44H            ;2ND ROW,CURSOR POS.
          LCALL   CURSOR
          LJMP    NEWD

REDISP:   RET
;--------------------SAFE/LONG DELAY PERIOD----------------------

SAFE:     NOP
BETT:     LCALL   DELAY1
          LCALL   DELAY1
          LCALL   DELAY1
          LCALL   DELAY1
          DEC     R1
          MOV     A,R1
          CJNE    A,#00H,BETT
          RET
;--------------------PLUG REMOTE CONTROL IN----------------------

PLUG_IN:  PUSH    DPH
          PUSH    DPL

          MOV     D_COUNT,#25H
FLASH:    SETB    P1.2              ;'REQUEST DATES'LED ON
          LCALL   DELAY1
          CLR     P1.2              ; OFF
          LCALL   DELAY1
          MOV     A,D_COUNT
          DEC     A
          MOV     D_COUNT,A
          CJNE    A,#00H,FLASH

          POP     DPL
          POP     DPH
          RET
```

```
;--------------------DISPLAY RTC DATE - DAYS----------------------

READ_DATE:MOV    RAM_H,#21H         ;DAYS
          MOV     RAM_L,#04H
          MOV     D_COUNT,#02H

DATE:     MOV     DPH,RAM_H
          MOV     DPL,RAM_L
          MOVX    A,@DPTR
          INC     DPL
          MOV     RAM_L,DPL

          LCALL   DISP_CHAR
          INC     TRACK
          DJNZ    D_COUNT,DATE
          RET
;--------------------CHECK SECURITY CODES------------------------

SECURITY: MOV     R6,#00H            ;FAILURE DETECTOR
          MOV     R5,COUNTER         ;SECURITY CODE LENGTH
 AGAIN:   MOV     R7,#00H
          MOV     DPH,XRAM_H         ;FETCH XRAM PTR
          MOV     DPL,XRAM_L
          MOVX    A,@DPTR            ;CODE TO ACC
          MOV     R7,A
          INC     DPL
          MOV     XRAM_L,DPL

          MOV     DPH,RAM_H          ;PREPROGRAMMED DATA
          MOV     DPL,RAM_L
          MOVX    A,@DPTR
          INC     DPL
          MOV     RAM_L,DPL
          XRL     A,R7
          CJNE    A,#00H,OUT         ;QUIT
          DEC     R5
          MOV     A,R5
          JNZ     AGAIN
          SJMP    BACK
OUT:      INC     R6
BACK:     NOP
          RET
;--------------------CLEAR MEMORY---------------------------------

CLEAN:    MOV     DPTR,#2000H
CONT:     MOV     A,#2AH             ;ASTERISK ****
          MOVX    @DPTR,A
          INC     DPTR
          MOV     A,DPH
          CJNE    A,#21H,CONT
          MOV     A,DPL
          CJNE    A,#3AH,CONT
          MOV     DPTR,#2020H        ;REBOOTING PROCESS
          MOV     A,#01H
          MOVX    @DPTR,A
          RET
```

```
;--------------------DELETE DATES FROM DISPLAY--------------------

DELETE:   MOV     COUNTER,#08H
DECK :    MOV     A,#20H
          LCALL   DISP_CHAR
          INC     TRACK
          DJNZ    COUNTER,DECK
          RET
;--------------------WEEK DAY SETTING-----------------------------

SWD:      MOV     RAM_H,DPH          ;CALL FOR INPUT,WED=4
          MOV     RAM_L,DPL          ; SUN=1
          MOVX    A,@DPTR
          INC     DPL
          MOV     RAM_L,DPL
          MOV     DPTR,#800EH
          MOVX    @DPTR,A
          RET
;--------------------SETTING OF TIME\DATE\DAY--------------------

TIMEDATE: MOV     A,#05H             ;STOP CLK
          MOV     DPTR,#8000H
          MOVX    @DPTR,A

          MOV     A,#0FH             ;TIME POS.ON DISPLAY
          LCALL   CURSOR

          MOV     R2,#06H            ;DIGIT COUNTER
REPEAT:   MOV     R0,#02H            ;UNITS OF 2 COUNTER

HRS2:     MOV     DPH,RAM_H
          MOV     DPL,RAM_L          ;START ADD. HRS
          MOVX    A,@DPTR
          ANL     A,#0FH
          ORL     A,#30H
          INC     DPL
          MOV     RAM_L,DPL
          PUSH    ACC
          LCALL   DISP_CHAR
          POP     ACC
          MOV     DPH,CLK_H          ;FETCH HRS POS.RTC
          MOV     DPL,CLK_L
          MOVX    @DPTR,A
          DEC     DPL
          MOV     CLK_L,DPL
          DEC     R2
          MOV     A,R2
          JZ      DATED              ;START DATE ENTRY
          DJNZ    R0,HRS2

          MOV     A,#3AH             ;COLON BETWEEN TIME FORMAT
          LCALL   DISP_CHAR          ;      HH:MM:SS
          LJMP    REPEAT
;------------- SETTING OF DATE  ---------------
DATED:    MOV     CLK_L,#0DH         ;YRS - TENS POS.
          MOV     R2,#06H
```

```
REPEAT1:  MOV     R0,#02H
YR2:      MOV     DPH,RAM_H
          MOV     DPL,RAM_L
          MOVX    A,@DPTR
          ANL     A,#0FH
          ORL     A,#30H
          INC     DPL
          MOV     RAM_L,DPL
          MOV     DPH,CLK_H
          MOV     DPL,CLK_L
          MOVX    @DPTR,A
          DEC     DPL
          MOV     CLK_L,DPL
          DEC     R2
          MOV     A,R2
          JZ      WEEK                ;WEEK DAY SETTING
          DJNZ    R0,YR2
          LJMP    REPEAT1

WEEK:     LCALL   SWD                 ;SET WEEK DAY

          MOV     A,#01H
          MOV     DPTR,#2020H         ;REBOOTING PROCESS
          MOVX    @DPTR,A

          MOV     A,#00H
          MOV     DPTR,#8000H
          MOVX    @DPTR,A             ;START CLK

          RET
;--------------------READ RTC FOR TIME/DISPLAY--------------------

CLOCK:    MOV     A,#0FH              ;DISPLAY POSITION
          LCALL   CURSOR

          MOV     CLK_H,#80H
          MOV     CLK_L,#07H

          MOV     R3,#07H             ;CYCLES COUNTER
          MOV     R4,#03H             ;DIGITS COUNTER

          MOV     DPTR,#8000H
          MOVX    A,@DPTR             ;DCF TEST

          MOV     DPH,CLK_H
          MOV     DPL,CLK_L
NEW1:     MOVX    A,@DPTR             ;READ CLOCK
          DEC     DPL
          MOV     CLK_L,DPL
          PUSH    ACC
          DJNZ    R3,CONVERT1
          POP     ACC
          RET

CONVERT1: DJNZ    R4,DIGIT
          MOV     R4,#02H
          MOV     A,#3AH              ;COLON
```

```
          LCALL   DISP_CHAR

DIGIT:    POP     ACC
          ANL     A,#0FH
          ORL     A,#30H
          LCALL   DISP_CHAR
          LJMP    NEW1
;--------------------READ/STORE RTC DATE READING-----------------

STOR_DATE:MOV    D_COUNT,#06H      ;CYCLES COUNTER
          MOV     CLK_L,#0DH        ;TENS OF YR

          MOV     RAM_H,#21H
          MOV     RAM_L,#00H
          MOV     DPTR,#8000H
          MOVX    A,@DPTR           ;TEST DCF

BBB:      MOV     DPH,CLK_H
          MOV     DPL,CLK_L
          MOVX    A,@DPTR           ;READ RTC
          ANL     A,#0FH            ;CONVERT ASCII
          ORL     A,#30H
          DEC     DPL
          MOV     CLK_L,DPL
          MOV     DPH,RAM_H
          MOV     DPL,RAM_L
          MOVX    @DPTR,A           ;STORE DATE DIGIT IN RAM
          INC     DPL
          MOV     RAM_L,DPL
          DJNZ    D_COUNT,BBB
          RET
;--------------------DELAY ROUTINE--------------------------------

DELAY:    PUSH    ACC
          MOV     BUFFER,#05H
RELOAD:   MOV     A,#04FH
AGAIN1:   DEC     A
          JNZ     AGAIN1
          DJNZ    BUFFER,RELOAD
          POP     ACC
          RET

DELAY1:   PUSH    ACC
          MOV     BUFFER,#0BFH
RELOAD1:  MOV     A,#0FFH
AGAIN2:   DEC     A
          JNZ     AGAIN2
          DJNZ    BUFFER,RELOAD1
          POP     ACC
          RET
;--------------------RTC INTERUPT ROUTINE------------------------

TIME_OUT: PUSH    DPH
          PUSH    DPL
          PUSH    D_COUNT           ;DATE COUNTER
          PUSH    TRACK             ;CURSOR POSITION
```

```
          LCALL    CLOCK              ;READ TIME
          POP      TRACK
          MOV      A,TRACK
          LCALL    CURSOR             ;RESTORE CURSOR POS.
          MOV      TRACK,#00H
          POP      D_COUNT
          POP      DPL
          POP      DPH
          RETI
;-----------------XFER & DISPLAY START/END DATE (XRAM TO RAM)-----

TRANSFER: MOV      R2,#02H            ;TENS/UNITS OF DIGITS
          MOV      COUNTER,#06H       ;DIGITS COUNTER

A200:     MOV      DPH,XRAM_H
          MOV      DPL,XRAM_L
          MOVX     A,@DPTR
          MOV      R5,A
          INC      DPL
          MOV      XRAM_L,DPL

          MOV      DPH,RAM_H
          MOV      DPL,RAM_L
          MOV      A,R5
          MOVX     @DPTR,A            ;STORING
          INC      DPL
          MOV      RAM_L,DPL
          MOV      A,R5
          LCALL    DISP_CHAR
          INC      TRACK
          DJNZ     R2,Q2
          MOV      A,#20H             ;SPACE FOR DATE
          LCALL    DISP_CHAR
          INC      TRACK
          MOV      R2,#02H
Q2:       DJNZ     COUNTER,A200
          RET
;--------------------DISPLAY MESSAGE ROUTINE----------------------

DISP_MSG: SETB     RS
NEXT_CH:  CLR      A
          MOVC     A,@A+DPTR          ;GET CHAR
          JZ       EXIT
          INC      DPTR
          PUSH     DPH
          PUSH     DPL
          MOV      DPTR,#0A000H       ;CS5
          MOVX     @DPTR,A            ;DISPLAY CHAR
          LCALL    DELAY
          POP      DPL
          POP      DPH
          LJMP     NEXT_CH
EXIT:     RET
;--------------------DISPLAY CHARACTER----------------------------

DISP_CHAR:SETB   RS
          CLR      RD_WR
```

```
          PUSH      DPH
          PUSH      DPL
          MOV       DPTR,#0A000H
          MOVX      @DPTR,A
          POP       DPL
          POP       DPH
          RET
;--------------------MOVE  CURSOR--------------------------------

CURSOR:   CLR       RS
          CLR       RD_WR
          ADD       A,#80H
          MOV       DPTR,#0A000H
          MOVX      @DPTR,A
          RET
;----------------------------------------------------------------

START_D:  DB        'ST:',00H
END_D:    DB        'END:',00H
          END
```

APPENDIX D

Programme listing of Time, Date and Rebooting information

PAGE D2-D9

```
;TITLE          PREVENTION OF PIRATE TELEVISION
;SUBTTL         SETTING TIME/DATE/REBOOTING PROGRAM
;------------------------------------------------------------------
                      BSEG

RS          BIT     P1.0
RD_WR       BIT     P1.1
LED         BIT     P1.2
BUZZ        BIT     P1.3
POWER       BIT     P1.4

EPROM_H     DATA    30H
EPROM_L     DATA    31H
RAM_H       DATA    32H
RAM_L       DATA    33H
XRAM_H      DATA    34H
XRAM_L      DATA    35H
EVENT       DATA    36H                ;REPITITION COUNTER
COUNTER     DATA    37H                ;GENERAL COUNTER
TRACK       DATA    38H                ;TRACK CURSOR
;---------------------ENABLE INTERUPTS----------------------------

                     CSEG
            ORG     00H

            LJMP    INIT
            ORG     03H                ;VECTOR ADD INT0
            LJMP    KEY

            ORG     30H

INIT:       SETB    EA                 ;ENABLE INTERUPTS
            SETB    EX0                ;ENABLE INT0
            CLR     P1.2               ;BUZZER
            SETB    P1.4               ;POWER OFF
;-------------------INITIALIZE DISPLAY----------------------------

            CLR     RS
            CLR     RD_WR

            MOV     DPTR,#0A000H
            MOV     A,#01H             ;CLEAR DISPLAY
            MOVX    @DPTR,A
            LCALL   DELAY

            MOV     A,#02H             ;CURSOR HOME
            MOVX    @DPTR,A
            LCALL   DELAY

            MOV     A,#38H             ;FUNCTION SET
            MOVX    @DPTR,A
            LCALL   DELAY

            MOV     A,#06H             ;ENTRY MODE
            MOVX    @DPTR,A
            LCALL   DELAY
```

```
            MOV     A,#0EH              ;DISPLAY ON\CURSOR ON
            MOVX    @DPTR,A
            LCALL   DELAY
;------------------INITIALIZE KEYBOARD---------------------------

            MOV     DPTR,#8001H
            MOV     A,#02H              ;MODE
            MOVX    @DPTR,A

            MOV     A,#34H              ;PROG CLOCK SPEED
            MOVX    @DPTR,A


;                   MAIN PROGRAM
;                   ------------

BEGIN:      LCALL   CURSOR_OFF
            MOV     A,R7                ;R7=0DH ?
            MOV     R7,#00H
            CJNE    A,#0DH,BEGIN        ;'ENTER' SCREEN ON
            MOV     R7,#00H

            LCALL   CURSOR_ON           ;DISPLAY ON

            LCALL   BLANK_LCD
            MOV     DPTR,#CLR_MEM       ;'CLEAR REMOTE MEMORY'
            LCALL   DISP_MSG

            MOV     A,#40H
            LCALL   CURSOR
            MOV     DPTR,#CLR_MEM1      ;'YES/NO - Y/N'
RRR:        LCALL   DISP_MSG

            MOV     A,R7                ;GET KEY VAL
            MOV     R7,#00H
            CJNE    A,#59H,DONE         ;YES
            LCALL   DISP_CHAR

BACK:       MOV     A,R7
            CJNE    A,#0DH,BACK         ;CR?
            LCALL   DELAY1

            CLR     P1.4                ;POWER ON REMOTE
            LCALL   DELAY
            LJMP    DDD                 ;CLR MEMORY

DONE:       CJNE    A,#4EH,RRR          ;NO
            LCALL   DISP_CHAR
            LCALL   DELAY1
            LCALL   DELAY1
            LCALL   DELAY1
            LJMP    FIN                 ;QUIT

DDD:        MOV     R7,#00H
            LCALL   DELAY
            LCALL   CLR_MEMORY          ;CLEAR REMOTE MEMORY
            LCALL   BLANK_LCD           ;CLEAR DISPLAY
```

```
          MOV     R0,#00H
          MOV     XRAM_H,#60H         ;REMOTE CONTROL ADDRESS
          MOV     XRAM_L,#13H

          MOV     TRACK,#0CH          ;TRACK CURSOR
          MOV     DPTR,#TIME_DATE     ;'** TIME/DATE ** '
          LCALL   DISP_MSG            : HHMMSSYYMMDDW0


          MOV     A,#0CH              ;DISPLAY POSITION
          LCALL   CURSOR
          LCALL   DISPLAY2

          LCALL   BLANK_LCD

          MOV     XRAM_H,#62H
          MOV     XRAM_L,#00H

          MOV     TRACK,#0CH
          MOV     DPTR,#SERIAL_NO      ;**SERIAL NO**
          LCALL   DISP_MSG

          MOV     A,#0CH
          LCALL   CURSOR
          LCALL   DISPLAY2

          MOV     A,#40H
          LCALL   CURSOR              ;CURSOR ON 2ND LINE
          MOV     TRACK,#4CH
          MOV     DPTR,#TV            ;' ** TV MAKE ** '
          LCALL   DISP_MSG

          MOV     A,#4CH
          LCALL   CURSOR
          LCALL   DISPLAY2

          LCALL   BLANK_LCD
          MOV     A,#00H
          LCALL   CURSOR              ;CURSOR ON 1ST LINE

          MOV     TRACK,#0CH
          MOV     DPTR,#SIZE          ;' ** SCREEN SIZE ** '
          LCALL   DISP_MSG

          MOV     A,#0CH
          LCALL   CURSOR
          LCALL   DISPLAY2

          LCALL   DELAY
          SETB    P1.4                ;POWER OFF REMOTE
          LCALL   DELAY

FIN:      LCALL   BLANK_LCD
          LJMP    BEGIN
```

```
;-----------------------SUBROUTINES START-----------------------

;-----------------------BACKSPACE ROUTINE/RECTIFIES MEMORY TOO---

BACSPACE:   DEC     TRACK               ;BACKSPACE CURSOR ON
            MOV     A,TRACK             ;       DISPLAY
            LCALL   CURSOR
            MOV     A,#20H              ;DELETE CHAR.
            LCALL   DISP_CHAR

            MOV     A,TRACK             ;DELETED PREVIOUS CHAR.
            LCALL   CURSOR              ;RESET CURSOR CORRECT
            POSITION
            MOV     DPH,XRAM_H
            MOV     A,DPH
            CJNE    A,#60H,DUO          ;ADDRESS ?
            LJMP    XRAM_MEM
DUO:        CJNE    A,#62H,LEAVE
XRAM_MEM:   MOV     DPL,XRAM_L
            DEC     DPL                 ;RECTIFIES MEMORY
            MOV     XRAM_L,DPL
LEAVE:      MOV     A,#00H
            RET
;-----------------------DISPLAY AND SAVE TO REMOTE CONTROL-------

DISPLAY2:    MOV     A,R7
             MOV     R7,#00H
             CJNE    A,#7FH,VIEWER
             LCALL   BACSPACE           ;BACKSPACE ROUTINE - XRAM

VIEWER:      CJNE    A,#0DH,VIEW
             LJMP    OUTT
VIEW:        CJNE    A,#00H,VIEW1
             LJMP    DISPLAY2

VIEW1:       PUSH    ACC
             LCALL   DISP_CHAR
             INC     TRACK

             MOV     DPH,XRAM_H
             MOV     DPL,XRAM_L
             POP     ACC
             CJNE    A,#3AH,PASS       ;CHECK FOR COLON
             LJMP    DISPLAY2          ;  NO SAVE

PASS:        MOVX    @DPTR,A           ;SAVE LETTER/NO
             INC     DPTR
             MOV     XRAM_L,DPL

             LJMP    DISPLAY2
OUTT:        MOV     R7,#00H
             RET
```

```
;------------------------CLEAR DISPLAY---------------------------

BLANK_LCD:   PUSH     DPH
             PUSH     DPL
             CLR      RS
             CLR      RD_WR

             MOV      A,#01H              ;CLEAR DISPLAY
             MOV      DPTR,#0A000H
             MOVX     @DPTR,A
             LCALL    DELAY
             POP      DPL
             POP      DPH
             RET

;------------------------DISPLAY MESSAGE ROUTINE----------------

DISP_MSG:    SETB     RS
NEXT_CH:     CLR      A
             MOVC     A,@A+DPTR           ;GET CHAR
             JZ       EXIT
             INC      DPTR
             PUSH     DPH
             PUSH     DPL
             MOV      DPTR,#0A000H        ;CS5
             MOVX     @DPTR,A             ;DISPLAY CHAR
             LCALL    DELAY
             POP      DPL
             POP      DPH
             LJMP     NEXT_CH             ;NEXT CHARACTER
EXIT:        RET
;------------------------MOVE CURSOR-----------------------------

CURSOR:      CLR      RS
             CLR      RD_WR
             ADD      A,#80H              ;POINT CURSOR TO FIRST\SECOND
             MOV      DPTR,#0A000H        ;          ROW
             MOVX     @DPTR,A             ;DISPLAY LINE
             LCALL    DELAY
             RET
;--------------------------KEY LOOKUP----------------------------

KEY:         MOV      A,#40H              ;SETUP READ RAM
             MOV      DPTR,#8001H
             MOVX     @DPTR,A

             MOV      DPTR,#8000H
             MOVX     A,@DPTR             ;READ RAM
             ANL      A,#3FH              ;RECTIFIES,GET CORRECT KEY
             ADD.
             MOV      DPTR,#KEY_LOOKUP
             MOVC     A,@A+DPTR           ;FETCH KEY
             MOV      R7,A                ;SAVE LETTER

             MOV      A,#0F0H             ;END OF INTERUPT
             MOV      DPTR,#8001H
             MOVX     @DPTR,A
```

```
            MOV     A,#00H
            RETI

;---------------------SWITCH DISPLAY OFF------------------------

CURSOR_OFF: PUSH    DPH
            PUSH    DPL
            CLR     RS
            CLR     RD_WR
            MOV     A,#08H
            MOV     DPTR,#0A000H
            MOVX    @DPTR,A
            LCALL   DELAY
            POP     DPL
            POP     DPH
            RET
;---------------------SWITCH DISPLAY ON-------------------------

CURSOR_ON:  PUSH    DPH
            PUSH    DPL
            CLR     RS
            CLR     RD_WR
            MOV     A,#0EH
            MOV     DPTR,#0A000H
            MOVX    @DPTR,A
            LCALL   DELAY
            POP     DPL
            POP     DPH
            RET
;---------------------CLEAR REMOTE MEMORY-----------------------

CLR_MEMORY: MOV     DPTR,#6000H
CONT:       MOV     A,#2AH              ;********
            MOVX    @DPTR,A
            INC     DPTR
            MOV     A,DPH
            CJNE    A,#7FH,CONT         ;CHECK H-ADDRESS
            MOV     A,DPL
            CJNE    A,#40H,CONT
            LCALL   DELAY
            RET
;---------------------DELAY ROUTINE-----------------------------

DELAY:      PUSH    ACC
            MOV     EVENT,#013H
RELOAD:     MOV     A,#0FFH
AGAIN:      DEC     A
            JNZ     AGAIN
            DJNZ    EVENT,RELOAD
            POP     ACC
            RET

DELAY1:     PUSH    ACC
            MOV     COUNTER,#0FFH
RELOAD2:    MOV     EVENT,#0FFH
RELOAD1:    MOV     A,#0FFH
```

```
AGAIN1:      DEC     A
             JNZ     AGAIN1
             DJNZ    EVENT,RELOAD1
             DEC     COUNTER
             JNZ     RELOAD2
             POP     ACC
             RET

;----------------------DISPLAY CHARACTER-----------------------

DISP_CHAR:   SETB    RS
             CLR     RD_WR
             PUSH    DPH
             PUSH    DPL
             MOV     DPTR,#0A000H
             MOVX    @DPTR,A
             LCALL   DELAY
             POP     DPL
             POP     DPH
             RET
;-------------------------------------------------------------

CLR_MEM:     DB      '* CLEAR REMOTE CONTROL MEMORY *',00H
CLR_MEM1:    DB      '       YES/NO - Y/N   ',00H
SIZE:        DB      'SCREEN SIZE:XXXX',00H
TIME_DATE:   DB      'TIME/DATE  :XXXXXXXXXXXXXX',00H
SERIAL_NO:   DB      'SERIAL NO  :XXX XXX XXX X',00H
TV:          DB      'TV MAKE    :XXXXXXXXXX',00H
KEY_LOOKUP:  DB      41H         ;A
             DB      42H         ;B
             DB      43H         ;C
             DB      44H         ;D
             DB      45H         ;E
             DB      00H
             DB      00H
             DB      00H
             DB      46H         ;F
             DB      47H         ;G
             DB      48H         ;H
             DB      49H         ;I
             DB      4AH         ;J
             DB      00H
             DB      00H
             DB      00H
             DB      4BH         ;K
             DB      4CH         ;L
             DB      4DH         ;M
             DB      4EH         ;N
             DB      4FH         ;O
             DB      00H
             DB      00H
             DB      00H
             DB      50H         ;P
             DB      51H         ;Q
             DB      52H         ;R
             DB      53H         ;S
             DB      54H         ;T
```

```
DB      00H
DB      00H
DB      00H
DB      55H         ;U
DB      56H         ;V
DB      57H         ;W
DB      58H         ;X
DB      59H         ;Y
DB      00H
DB      00H
DB      00H
DB      5AH         ;Z
DB      30H         ;0
DB      31H         ;1
DB      32H         ;2
DB      33H         ;3
DB      00H
DB      00H
DB      00H
DB      34H         ;4
DB      35H         ;5
DB      36H         ;6
DB      37H         ;7
DB      38H         ;8
DB      00H
DB      00H
DB      00H
DB      39H         ;9
DB      7FH         ;DELETE
DB      3AH         ;COLON
DB      20H         ;FORWARDSPACE
DB      0DH         ;CR
END
```