

Freescale Semiconductor

AN495/D

RDS decoding for an HC11-controlled radio

P. Topping,
Motorola Ltd,
East Kilbride, Scotland

Introduction

This application note describes, and lists the software of, the RDS aspects of the HC11 radio controller described in AN494/D. The complete application constitutes a synthesised multiband radio which includes RDS decoding (FM, band II) and uses an MC68HC(7)11 microprocessor whose program can be on-chip or contained in an external EPROM. Both LCD and VFD 16-character dot-matrix display modules can be used to display RDS and tuning information. Traffic messages on the current frequency or on another frequency, initiated by the reception of RDS EON data, are handled. The station carrying the TA is tuned for the duration of the message, followed by a return to the original frequency.

Figure 1 shows a block diagram of the application. The microprocessor used is the MC68HC(7)11. The K4 (and similar chips such as the P2 and PH8) can be used in expanded mode but the application has been included in the ROM of an E32 (ZC403311) and two PH8s (ZC428200 and ZC428202). In order to use the ROMed parts in this application, the first three bytes of EEPROM should contain an extended jump to the appropriate state address. The E32 (ZC403311) requires \$7E, \$90 and \$00 at addresses \$B600, \$B601 and \$B602, while the PH8 requires \$7E, \$40 and \$00 at addresses \$0D00, \$0D01 and \$0D02. This can be done using either PCbug11 or the Buffalo monitor (see reference 5). The E32 version uses all the I/O and can therefore only be used in single-chip mode. The circuit diagram of the HC11E controller is shown in figure 2. The 40 programs (10 on FM and MW and 20 on SW) which can be stored using the HC11E's on-chip EEPROM contain, in addition to frequency, an 8-character name (PS name on a station with RDS) and, on FM only, PI code and a traffic announcement inhibit bit.

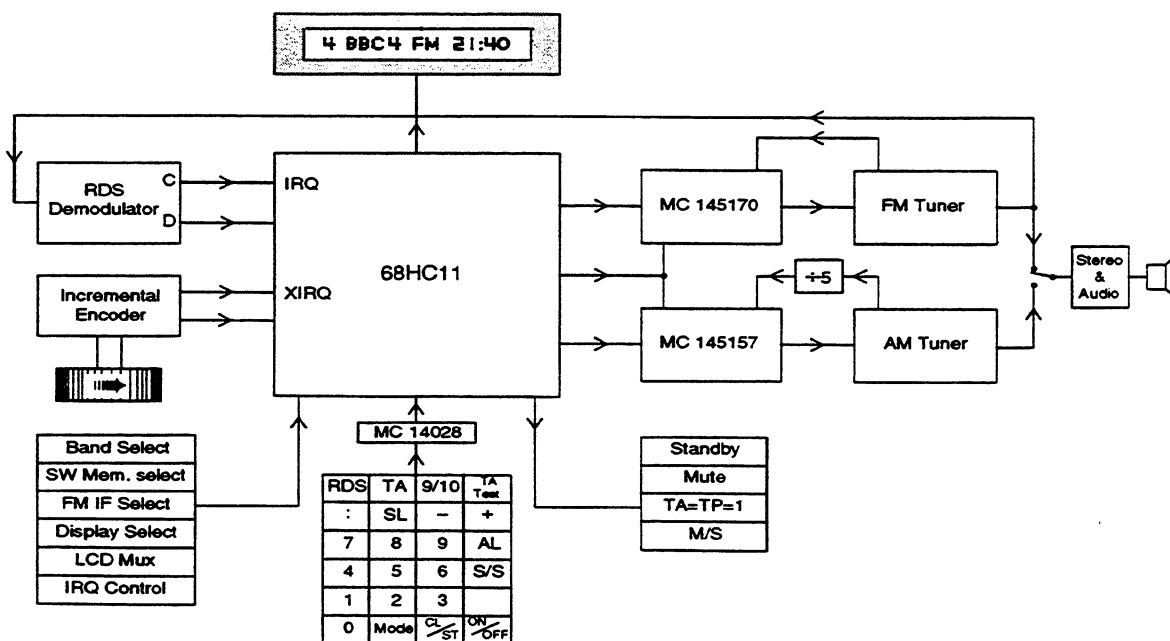


Figure 1. Main block diagram

This document was originated outside of the United States and may, therefore, contain information on devices not actively supported within the United States Marketing groups. Please contact your local Sales Office for more information.

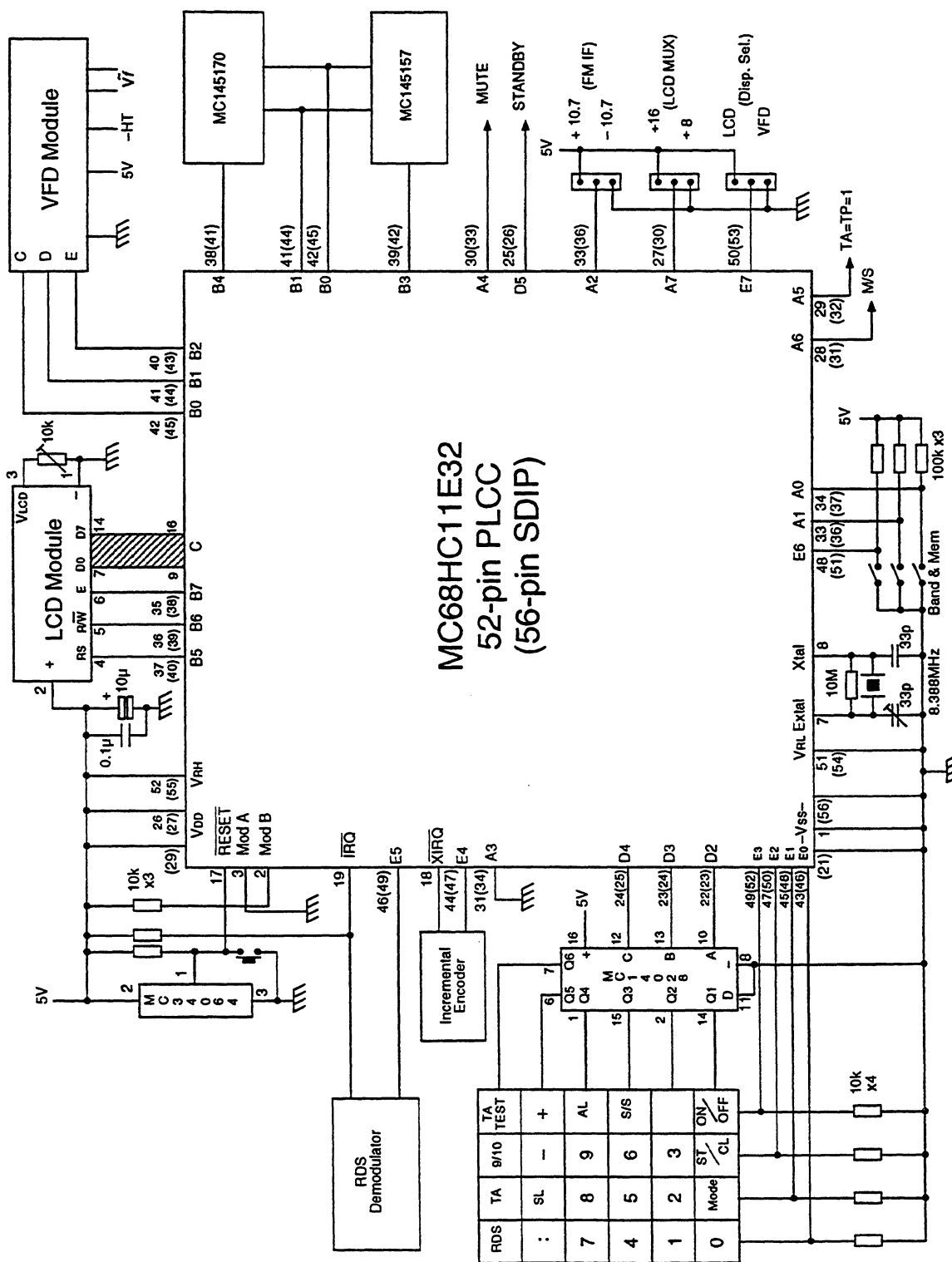


Figure 2 68HC11E32 circuit

RDS features

The Radio Data System adds a digital data capability to the FM VHF transmissions on band II (87.5 to 108 MHz). The specification is defined in CENELEC EN 50067 (formerly EBU Technical Document 3244, reference 2). An MC68HC05E0 implementation of RDS was described in AN460/D (reference 3). It monitored the RDS activity on the MPX signal of a VHF radio but was not able to tune the radio and could therefore not use AF or EON data. This application can tune the radio and uses EON data to retune the radio when a traffic announcement (TA) is taking place on another frequency. An announcement is initiated by an RDS group 14B and the radio retunes if TAs are enabled. At the end of the announcement the original station is re-tuned. TAs are not active in standby mode (standby line high).

To transmit the data, a subcarrier is added at 57 kHz. This subcarrier is amplitude modulated with the shaped bi-phase coded data signal. The subcarrier itself is suppressed to avoid data modulated cross-talk in phase-locked loop stereo decoders and to maintain compatibility with the German ARI system which uses the same subcarrier frequency. Information is sent in groups of four 26-bit blocks. Each group of 104 bits is one of several types containing different information. It is up to the broadcaster to decide which features are transmitted as long as the specified format is adhered to and PI, PTY and TP are included. Each group contains a different sub-set of the RDS features. A list of all currently defined features is shown in table 1.

Table 1. RDS features

Feature	Information
PI	Program identification
PTY	Program type
PS	Program service name
RT	Radiotext
CT	Clock time and date
AF	Alternative frequencies
TA	Traffic announcement
TP	Traffic program
MS	Music/speech switch
DI	Decoder identification
PIN	Programme item number
EON	Enhanced other networks
TDC	Transparent data channel
INH	In-house data

The retrieval of data is carried out by demodulation hardware which generates clock and data signals that can be used by the microprocessor. Suitable devices which can perform this function include SAA6579, SAA7579T (plus an external filter), TDA7330, LA2231 and RDS hybrids.

This application supports PI, PTY, PS, RT, CT, TP, TA, MS, DI, PIN and EON. These features facilitate permanent display of the 8-digit station name (PS) and time (CT) and, on request, can display program type (PTY), radiotext data (RT) and the status of the other RDS information (see table 6). EON data can be displayed and used to switch to traffic announcements, but the retuning features associated with AF are not supported as they are only appropriate for a radio intended for use in a vehicle. In a car radio, AF data would be used to tune the radio to the strongest signal carrying the selected service.

Table 2 shows all the currently defined RDS group types and the RDS features they contain. PI, PTY and TP are contained in all groups. This allows this information to be gathered quickly after the radio has been retuned. The other features are contained only in specific group types and the update frequency is thus largely up to the broadcaster. The next most important information for a car radio (AF and TA) is sent using type 0 groups which also contain the M/S and DI bits which can be used to control hardware within the radio. There are two methods of increasing the repetition rate of important information. Type B groups contain the PI code twice (in blocks 1 and 3) and type 15B groups also repeat their block 2 information (TP, PTY, M/S and DI) in block 4. Information which is required less frequently (e.g., PIN) is sent less often, while type 4 groups (CT) are sent only once per minute.

Table 2. RDS Groups

Group	Features
All	PI, PTY, TP
0	TA, DI, MS, PS, AF
1	PIN
2	RT
3	ON (replaced by EON)
4A	CT
5	TDC
6	INH
14	EON
15B	TA, DI, MS

Other network (ON) information was originally sent using type 3 groups. Limitations in the definition of this group has caused it to be superseded by type 14 groups. This enhanced other network (EON) group type effectively replaces type 3 groups which are no longer used. This application uses type 14 groups but does not handle the old type 3 groups.

Decoding

Each 26-bit block contains 16 bits of data and 10 extra bits which are used for synchronisation and error detection. There are no gaps between blocks or groups, the synchronisation being done by looking for specific checkwords in the incoming data. In order to look for a checkword a stream of 26 consecutive data bits has to be multiplied by a fixed 10x26 matrix (Figure 3).

The result of this multiplication is a 10-bit word which is compared with allowed values. There are 5 of these 10-bit "syndromes", one for each of blocks 1, 2 and 4 and two for block 3 (see table 3). The alternative syndrome for block 3 is used in the B version of a group. In this version the PI code is sent in block 3, replacing what would be sent in the A version of the same group type. This is done to increase the frequency of sending the PI code so that it can be acquired more quickly.

```

10 0000 0000 ($02,$00)
01 0000 0000 ($01,$00)
00 1000 0000 ($00,$80)
00 0100 0000 ($00,$40)
00 0010 0000 ($00,$20)
00 0001 0000 ($00,$10)
00 0000 1000 ($00,$08)
00 0000 0100 ($00,$04)
00 0000 0010 ($00,$02)
00 0000 0001 ($00,$01)
10 1101 1100 ($02,$DC)
01 0110 1110 ($01,$6E)
00 1011 0111 ($00,$B7)
10 1000 0111 ($02,$B7)
11 1001 1111 ($03,$9F)
11 0001 0011 ($03,$13)
11 0101 0101 ($03,$55)
11 0111 0110 ($03,$76)
01 1011 1011 ($01,$EB)
10 0000 0001 ($02,$01)
11 1101 1100 ($03,$DC)
01 1110 1110 ($01,$EE)
00 1111 0111 ($00,$F7)
10 1010 0111 ($02,$A7)
11 1000 1111 ($03,$8F)
11 0001 1011 ($03,$1B)

```

Figure 3 10x26 decoding matrix

Table 3. Syndromes

Block	Syndrome	Binary	Hex.
1	A	11 1101 1000	\$03,\$D8
2	B	11 1101 0100	\$03,\$D4
3	C	10 0101 1100	\$02,\$5C
	C'	11 1100 1100	\$03,\$CC
4	D	01 0101 1000	\$01,\$58

This syndrome test has to take place after each bit is received, using the last 26 bits, until a valid syndrome is found. In this application, only syndrome A is accepted during the bit-by-bit syndrome check and the data is used only after four valid syndromes have been acquired. A more complex algorithm could allow any syndrome to be accepted during initial synchronisation and require less than four valid syndromes before the data is used. This would reduce the time taken to acquire the PI code, which is also included in block 3 of type B groups, but increases the likelihood that random data, giving a valid syndrome, will be used in error. The bit rate is 1187.5 Hz so the control microprocessor is busy during this initial synchronisation (about 20% of the capability of an HC11 at a 2 MHz bus frequency). Once a valid syndrome has been found, subsequent syndrome checks need be done only after the next 26 bits have been received, as this is when the next valid syndrome would be expected. If it is not found, then bit-by-bit checking is re-started. Once consecutive A, B, C (or C') and D syndromes have been detected, a complete group has been acquired and the data can be used.

Software

The software for this application is in three modules and was assembled and linked using the Introl relocatable assembler and linker. The first module is listed in AN494/D and contains all the main control routines including the main loop and keyboard scanning and the function to be performed by each key. The second and third modules are listed in this application note. The second module (Appendix 1) contains the RDS and display functions while the third module (Appendix 2) is the 4-function 9-digit integer BCD arithmetic required for the MJD date calculations. EB419/D (reference 5) describes and lists additional debug code contained in the ROMed parts.

The second module services the RDS and timer interrupts and performs all RDS and display functions. The RDS functions were converted from the HC05 application described in AN460/D (reference 3) using the methods described in AN478/D (reference 4).

The RTI timer interrupt routine (TINTB) updates the RT scrolling pointers (DISP1 and DISP2). These pointers are incremented regularly whether or not an RT display is active so that the software can be easily converted to using a 2-line LCD module in which the top line is the normal display of PS-name, time etc., and the lower line is a permanent display of scrolling RT. The timer interrupt also decrements the sleep timer and updates the RAM locations used to store hours, minutes and seconds. All RDS data (except date and time) is cleared by this routine if no valid RDS data is detected for a period of 10 seconds. The RTI timer is enabled to cause an interrupt every 31.25 ms to run the real-time clock. Correct operation of this clock in the absence of an RDS signal requires that a 8.388 MHz crystal be used.

Hardware interrupts are vectored to jump to SDATA where serial data is received from the RDS demodulator. The clock edge causes an interrupt and a data bit is read by bit 5 on port E. The bit is shifted into a 4-byte RAM register and, if appropriate, the matrix multiplication is performed. The state of the flag at bit 0 of STAT2 determines if the multiplication is to take place after every bit or only after 26 bits have arrived. The multiplication is performed using two EOR instructions for every bit (two are required as the 10-bit syndrome requires two bytes). As the top of the matrix (see figure 3) is the unity matrix, the first 10 bits are transferred directly into the accumulators. This reduces what is a rather long and repetitive piece of code. It could be shortened by using a loop but this would incur an unacceptable penalty in execution time.

After the multiplication has been performed, the resultant 10-bit number is compared with the allowed syndromes (see table 3). The variable LEV records the current block level. It is initially zero but is incremented each time a valid syndrome is found. When it is zero, only syndrome A is accepted. If this is found then syndrome B is expected 26 bits later so when LEV is one, only syndrome B is accepted. If an invalid syndrome is found, LEV is cleared, the syndrome confidence level CONF is decremented and the interrupt is ended.

When a valid syndrome is found, CONF is increased by 4 and the 16 data bits saved in the relevant bytes of TMPGRP. If the valid syndrome is type D then a complete group has been received and all 8 bytes are transferred to the 8 RAM locations at GROUP. This double buffering means that the data in GROUP can be used while interrupts are overwriting TMPGRP with new data. Complete groups of data are handled in the subsequent routines according to their group type.

The confidence level CONF is used to decide what should be done if the data becomes unreliable due to a poor RF input to the receiver. When the first valid syndrome is found it is initialised to 42. Subsequent valid syndromes increment it by four and invalid ones decrement it by 1. If CONF falls below 41, then it is assumed that synchronisation has been lost and a bit-by-bit re-synchronisation is carried out. If it falls below 10, the signal is deemed unacceptable and the displays are re-initialised. The confidence level is not incremented by the detection of a valid syndrome if it is higher than 56.

The listed modules contain the display routines described in AN494/D. The displays are only updated when there is a change in the displayed data. At 8 Hz a check is made to see if any characters have changed, and if there has been a change, the display update routine is executed. This is done to minimise interference caused by communication with the displays. The colon between the hours and minutes of the time display changes at 1 Hz. This can be disabled (colon permanently displayed) by using the Time Colon key. The display routine (MOD) is executed in the idle loop if the flag bit 3 of STAT2 is set. It is set every 125 ms by timer B interrupts. If flag bit 4 of STAT2 is set the display is initialised indicating no valid RDS data. The dot-matrix modules are then updated, if necessary, with new data. Before each occasion that something is written to the LCD module, the subroutine WAIT is used. This checks that the controller in the module is not busy. The different display formats are selected by checking the various flags and the relevant routine executed. The normal display permanently shows PS name and time. As the locations in RAM used for hours and minutes contain binary numbers, they are converted to BCD before being written to the relevant bytes in DISP. Once all 16 bytes in DISP have been loaded, loops are used to send the data to the display modules. The standby display (alarm not enabled) shows date and time. After a power-up the display "Mon 0 inv 0:00" indicates that the date and time are invalid. The date and time will be correct once a valid RDS CT group has been received.

The VFD routine sends the same data as is shown on the LCD module to the serial VFD module. The display driver used has a different character set from the standard ASCII set used by the LCD module. The table VTAB is used to convert ASCII data into the required character in the VFD module. The small table INITF is used to send the required initialisation bytes to the VFD module. This module does not require a busy check but does require a delay between successive bytes. This is satisfied by the wait loop within the serial output loop VFDL.

Table 4. RDS Block and Group structure

Group	Block 1	Block 2	Block 3	Block 4
0,15B	PI code	15-12: Group no. 11: Group type 10: TP flag 9-5: PTY code 4: TA flag 3: M/S bit 2: DI bit 1-0: PS/DI address	AF (PI code in type 0B and 15B)	PS name 2 ASCII characters (as block 2 for 15B)
1	PI code	15-12: 0001 11: Group type 10: TP flag 9-5: PTY code 4-0: not used	not used (PI code in type 1B)	PIN data 15-11: day-of-month 10-6: hour 5-0: minute
2A	PI code	15-12: 0010 11: 0 10: TP flag 9-5: PTY code 4: Text A/B flag 3-0: Text address	RT 2 ASCII characters	RT 2 ASCII characters
4A	PI code	15-12: 0100 11: 0 10: TP flag 9-5: PTY code 4-2: not used 1-0: MJD (16-15)	CT 15-1: MJD (14-0) 0: hour (4)	CT 15-12: hour (3-0) 11-6: minute (5-0) 5: offset sense 4-0: offset (4-0)
14A	PI code	15-12: 1110 11: 0 10: TP flag 9-5: PTY code 4: TP (ON) flag 3-0: usage code	EON code: 0-3: PS 4: AF 5-9: AF (map) 10-11: not used 12-15: not imp.	PI (ON)
14B	PI code	15-12: 1110 11: 1 10: TP flag 9-5: PTY code 4: TP (ON) flag 3: TA (ON) flag 2-0: not used	PI code	PI (ON)

Table 4 shows the bit structure of the groups which are used in this application. Block 1 always contains the PI code. The five most significant bits in block 2 determine the group number and type. Block 2 also contains TP and PTY data. The uses of the other bits in blocks 2, 3 and 4 depend on the group number and type. Type B groups repeat the PI code in block 3 but type A groups contain a variety of information in blocks 3 and 4 depending on the group number.

PI, PTY, and TP

If a complete group has been received, the data can be processed. The buffering used would allow this to be done outside the interrupt but in this case there is sufficient time to do it within the interrupt. As they

are contained in all groups, PI, PTY and TP are handled first. PI is a 2-byte number which identifies the country, coverage area and service. It can be used by the control microprocessor but is not normally intended for display. This application facilitates the display of the current PI code. A change in PI code causes the initialisation of all RDS data as it indicates that the radio has been retuned. When a program is stored in NVM, its PI code is saved with the frequency and PS name. This information is used to find the correct frequency to tune to when a traffic announcement is initiated by EON. The EON information (contained in a group 14B) includes the PI code of the station transmitting the message. Further use would be made of the PI code in an application which used AF information.

PTY is a 5-bit number which indicates the current program type. At present 16 of these types are defined. Examples include "no programme type", "Current affairs" and "Pop music", although the actual syntax which is displayed is determined by the software of the controlling microprocessor. In this example PTY can be displayed on request. Table 5 shows the display used for each PTY code.

Table 5. PTY Types

PTY	Display
0	no program type
1	News
2	Current affairs
3	Information
4	Sport
5	Education
6	Drama
7	Culture
8	Science
9	Varied
10	Pop music
11	Rock music
12	Easy listening
13	Light classics
14	Serious classics
15	Other music
16-31	no program type

TP is a single bit flag and is set if the transmitter normally carries traffic information. After PI, PTY and TP have been updated, the group type (A/B) and group number (0 to 15) are identified. Group types 0A, 0B, 1A, 1B, 2A, 4A, 14A, 14B and 15B are handled. Table 2 shows the type of information contained in each group and table 4 the detailed structure of the groups actually used. The different groups are treated as detailed below.

PS, AF, TA, M/S and DI (groups 0 and 15B)

PS is the eight-character name of the station and is permanently displayed (except in standby mode). In the absence of RDS (e.g., AM bands) this application allows the name to be manually entered. If none is entered, then the frequency is used as the station name when the program is stored in EEPROM. AF would be used by a car radio to retune to the strongest signal carrying the selected service. AF data, along with TDC and INH, is not used in this application. TA, like TP, is a flag. TP is permanently set if the transmitter normally carries traffic information and TA is set when a traffic announcement is actually in progress. The combination TA=1, TP=0 is used to indicate that EON data is being used to supply information on other networks including traffic announcements. A port line (port A, bit 5) is asserted (low) when TA=TP=1. This can be used to demute or switch from another source (e.g., cassette) when a traffic announcement occurs. M/S is a single bit indicating either music or speech and is intended to be used to make a tone or volume adjustment to a radio's audio stage. The M/S bit is displayed on request. A port line (port A, bit 6) is asserted (low) when M/S=1. This can be used to control external hardware. The ROMed PH8s (ZC428200 and ZC428202) do not include the TA=TP=1 and M/S outputs. Decoder information (DI) constitutes four bits indicating the type of transmission (mono, stereo, binaural etc.). It is not currently in use in the UK but can be displayed as a number between 0 and 15.

As AF data is not handled, there is no difference in the treatment of groups 0A and 0B. PS data is extracted and placed in RAM according to the address bits in block 2 (see table 4). TA, DI and MS data are then read, DI is sent a single bit at a time and uses the same address bits as the PS name to determine which of the four bits is being updated. Groups of type 15B also contains all this switching information. They are used to increase the repetition rate of this data but contain no PS or AF information.

PIN (group 1)

Programme item number or PIN is used to identify the programme currently being broadcast. The format is a 2-byte number which includes the scheduled time and date (day-of-month) of the start of the programme.

Group types 1A and 1B are again treated identically as they contain the same data except for the repetition of the PI code in type 1B. The PIN data is recovered and saved in RAM. This is intended for future use to control external hardware, for example a tape recorder. This would facilitate the unattended recording of a pre-selected program. At present this application simply allows the display of PIN data both in its raw hexadecimal form and fully decoded to day-of-month and time (see table 6). Full use of PIN data would require continuously comparing the PIN day-of-month and time with a manually entered day-of-month and time and asserting an output pin when there was a match.

RT (group 2A)

Radiotext (RT) constitutes a string of up to 64 characters which give additional information regarding the service or programme currently being transmitted. In this application, RT is displayed on request on the 16-digit dot-matrix displays using scrolling. RT data from blocks 3 and 4 is written to RAM according to the address included in block 2. There are 4 address bits and four ASCII encoded bytes giving the possibility of 64 characters. The data often contains extra spaces to centre the text on a 2x32 character display. As these are not appropriate for a 16-character scrolling display, the software reduces all sequences of two or more spaces to a single space. If the Text A/B flag changes state, the RT area in RAM is cleared as this indicates that the message has changed. Group 2B is not handled as it is rarely if ever used (2B or not 2B.....).

Table 6. RDS display formats

RDS feature	Display format
CT date and time	Thu 12 May 21:35
PS name and CT time	4 BBC 4 FM 21:40
RT	Kaleidoscope
PTY	Culture
PI	PI code - C204
TA & TP	TP - 0 TA - 1
PIN(hex)	PIN no. - 655E
PIN(decoded)	12th at 21:30
MJD	MJ day - 49484
MS & DI	M/S M- DI 01
last TA 1	last TA PI C514
2	TA rtn: EON PI
EON 1	BBC 3 FM 92.10
2	BBC Gael 103.70
3	BBC Nwcl 96.00
4	BBC Scot 94.30
5	BBC Scot 92.50
6	BBC Scot 94.70
7	BBC Scot 93.50
8	Classic 101.70
9	BBC Eng 107.90
10	BBC 1 FM 99.50
11	BBC 2 FM 89.90
12	BBC R5 909kHz
13	-----
14	-----
15	-----
16	-----

CT (group 4A)

CT data is transmitted every minute on the minute and facilitates a very accurate clock, traceable to national standards. The (Modified Julian) date and local time variation are also transmitted. Except in manual mode, when it is replaced with the frequency, the time is permanently displayed. In standby mode the date is displayed instead of the PS name (see table 6). The MJD number, which is the form in which the date is received, can also be displayed.

Two of the more complex tasks to be performed are required to process the CT data. These are for the local time difference and the conversion of the MJD number into a recognisable date. The broadcast time is Universal Co-ordinated Time (UTC, effectively the same as GMT). Time differences from UTC, including summer (daylight saving) time, are sent as an offset of up to +/- 12 hours in half-hour increments. The date is transmitted as the MJD (Modifier Julian Day) number and has to be converted to day-of-week, day-of-month, month and year using the formulae:

```

Y'          =   int[(MJD-15078.2)/365.25]
M'          =   int[(MJD-14956.1-int{Y'x365.25})/30.6001]
Day         =   MJD-14956-int(Y'x365.25)-int(M'x30.6001)
If M'=14 or M'=15, then K=1; else K=0
Year        =   Y'+K
Month       =   M'-1-12K

```

The third software module (Appendix 2) contains the 4-function 9-digit integer BCD arithmetic required to make the MJD calculations. The main code uses these routines to display the time and date in conventional form, adjusting the time (and perhaps the date) according to the local offset.

EON (group 14)

EON (Enhanced Other Networks) replaces the older ON format. If type 14 groups are used to provide EON data, then type 3 groups (ON) will not be used; table 2 shows the currently defined group types. Type 14A groups are used to send data about other networks. A large amount of EON information can be sent using this group. It takes up to two minutes for all the data to arrive after the radio has been retuned. This application saves the PI code, PS name and principal frequency of up to 16 networks although more networks, each with many frequencies, and other data (e.g., PTY(ON), PIN(ON), TA(ON) etc.) may be sent. Table 6 shows the format of the EON displays.

Type 14B groups are used to switch to traffic announcements on a different frequency. They include the PI code of the station carrying the announcement. This PI code is searched for in NVM and the required station tuned if it is stored in NVM. This method allows the user to select which TAs are allowed (they will not occur if the station is not in NVM or if its TA inhibit bit is set) and avoids attempts to jump to an announcement which is not relevant or not receivable with sufficient signal strength to be useful. The complete procedure is described below.

Traffic announcement procedure

The radio can respond to EON initiated traffic announcements if they are enabled by the TRAFFIC (TA) key. This status is indicated by a decimal point at the 11th character on the dot-matrix displays. A switch to a TA on another frequency will only occur if the station has previously been stored in NVM (the EON data which can be displayed using the RDS key is not used for TA switching). The PI code of the last TA (or attempted TA) can be displayed by pressing the RDS key eight times. A further press displays one of the TA return/inhibit messages shown below. TAs which are the result of TA=TP=1 on the current frequency do not update the last TA PI or TA return/inhibit messages.

When a 14B group is received the following occurs:

Check traffic flag; if enabled proceed, otherwise set TA rtn/inhb message to:

TA inhb: flag - Traffic key inhibit flag (d.p. at the 11th character position).

Search for TA PI code in NVM; if found proceed, otherwise set TA rtn/inhb message to:

TA inhb: EON PI - The PI code given in 14B is not in the NVM.

Check station TA inhibit flag in NVM; if clear proceed, otherwise set TA rtn/inhb message to:

TA inhb: NVM - User inhibit of station using bit stored in NVM.

Retune to frequency stored in NVM against EON PI code. The PS name display changes to show the PS name of the service carrying the traffic announcement and the time display is replaced by the new frequency. If the service has its TP flag high, then the 10s of kHz digit will flash as in the manual mode display. After one second, check TP flag at the new frequency. If high then proceed, otherwise return to original frequency and set TA rtn/inhb message to:

TA rtn: TP low - TP station does not have TP bit high.

Check PI code at new frequency. If correct (same as 14B EON TA PI code) then proceed, otherwise retune to original frequency and set TA rtn/inhb message to:

TA rtn: PI code - PI code of TP station was not as expected.

After an additional 2 seconds, start to monitor the TA flag; if high, remain on current frequency, if low return to original frequency and set TA rtn/inhb message to:

TA rtn: TA low - TA flag of TP station low. This is the normal return method.

If, during a TA, the radio is manually retuned, the TA rtn/inhb message is set to:

TA rtn: manual - User initiated manual return.

References

1. AN494/D, An HC11-controlled Multi-band RDS Radio.
2. CENELEC EN 50067, Specifications of the Radio Data System (RDS), formerly EBU Technical Document. 3244).
3. AN460/D, An RDS Decoder using the MC68HC05E0.
4. AN478/D, HC05 to HC11 code conversion.
5. EB419/D, ROMed HC11E32 and HC11PH8 including Buffalo monitor and PCbug11 talker.

Appendix 1

```

2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100
101
102
103
104
105
106
107
108
109
110
111
112
113
114
115
116
117
118
119
120
121
122
123
124
125
126
127
128
129
130
131
132
133
134
135
136
137
138
139
140
141
142
143
144
145
146
147
148
149
150
151
152
153
154
155
156
157
158
159
160
161
162
163
164
165
166
167
168
169
170
171
172
173
174
175
176
177
178
179
180
181
182
183
184
185
186
187
188
189
190
191
192
193
194
195
196
197
198
199
200
201
202
203
204
205
206
207
208
209
210
211
212
213
214
215
216
217
218
219
220
221
222
223
224
225
226
227
228
229
230
231
232
233
234
235
236
237
238
239
240
241
242
243
244
245
246
247
248
249
250
251
252
253
254
255
256
257
258
259
260
261
262
263
264
265
266
267
268
269
270
271
272
273
274
275
276
277
278
279
280
281
282
283
284
285
286
287
288
289
290
291
292
293
294
295
296
297
298
299
300
301
302
303
304
305
306
307
308
309
310
311
312
313
314
315
316
317
318
319
320
321
322
323
324
325
326
327
328
329
330
331
332
333
334
335
336
337
338
339
340
341
342
343
344
345
346
347
348
349
350
351
352
353
354
355
356
357
358
359
360
361
362
363
364
365
366
367
368
369
370
371
372
373
374
375
376
377
378
379
380
381
382
383
384
385
386
387
388
389
390
391
392
393
394
395
396
397
398
399
400
401
402
403
404
405
406
407
408
409
410
411
412
413
414
415
416
417
418
419
420
421
422
423
424
425
426
427
428
429
430
431
432
433
434
435
436
437
438
439
440
441
442
443
444
445
446
447
448
449
450
451
452
453
454
455
456
457
458
459
460
461
462
463
464
465
466
467
468
469
470
471
472
473
474
475
476
477
478
479
480
481
482
483
484
485
486
487
488
489
490
491
492
493
494
495
496
497
498
499
500
501
502
503
504
505
506
507
508
509
510
511
512
513
514
515
516
517
518
519
520
521
522
523
524
525
526
527
528
529
530
531
532
533
534
535
536
537
538
539
540
541
542
543
544
545
546
547
548
549
550
551
552
553
554
555
556
557
558
559
560
561
562
563
564
565
566
567
568
569
570
571
572
573
574
575
576
577
578
579
580
581
582
583
584
585
586
587
588
589
590
591
592
593
594
595
596
597
598
599
600
601
602
603
604
605
606
607
608
609
610
611
612
613
614
615
616
617
618
619
620
621
622
623
624
625
626
627
628
629
630
631
632
633
634
635
636
637
638
639
640
641
642
643
644
645
646
647
648
649
650
651
652
653
654
655
656
657
658
659
660
661
662
663
664
665
666
667
668
669
670
671
672
673
674
675
676
677
678
679
680
681
682
683
684
685
686
687
688
689
690
691
692
693
694
695
696
697
698
699
700
701
702
703
704
705
706
707
708
709
710
711
712
713
714
715
716
717
718
719
720
721
722
723
724
725
726
727
728
729
730
731
732
733
734
735
736
737
738
739
740
741
742
743
744
745
746
747
748
749
750
751
752
753
754
755
756
757
758
759
760
761
762
763
764
765
766
767
768
769
770
771
772
773
774
775
776
777
778
779
780
781
782
783
784
785
786
787
788
789
790
791
792
793
794
795
796
797
798
799
800
801
802
803
804
805
806
807
808
809
810
811
812
813
814
815
816
817
818
819
820
821
822
823
824
825
826
827
828
829
830
831
832
833
834
835
836
837
838
839
840
841
842
843
844
845
846
847
848
849
850
851
852
853
854
855
856
857
858
859
860
861
862
863
864
865
866
867
868
869
870
871
872
873
874
875
876
877
878
879
880
881
882
883
884
885
886
887
888
889
890
891
892
893
894
895
896
897
898
899
900
901
902
903
904
905
906
907
908
909
910
911
912
913
914
915
916
917
918
919
920
921
922
923
924
925
926
927
928
929
930
931
932
933
934
935
936
937
938
939
940
941
942
943
944
945
946
947
948
949
950
951
952
953
954
955
956
957
958
959
960
961
962
963
964
965
966
967
968
969
970
971
972
973
974
975
976
977
978
979
980
981
982
983
984
985
986
987
988
989
990
991
992
993
994
995
996
997
998
999
1000
1001
1002
1003
1004
1005
1006
1007
1008
1009
1010
1011
1012
1013
1014
1015
1016
1017
1018
1019
1020
1021
1022
1023
1024
1025
1026
1027
1028
1029
1030
1031
1032
1033
1034
1035
1036
1037
1038
1039
1040
10
```

```

16
16
16
16
16
16
16 000000a6      STAT   RMB   1           0: MODE 1: STATION, 0: FREQ
16                  *           1: STEP 1: 50KHZ, 0: 10KHZ
16                  *           2: CLRD 1: CLEAR IF NO. KEYED
16                  *           3: TIMER MS BIT TOGGLE (64 HZ)
16                  *           4: RDS DATA CLEARING ENABLE
16                  *           5: KEY FUNCTION PERFORMED
16                  *           6: KEY REPEATING
16                  *           7: NOT JUST POWERED UP
16 000000a7      STAT2  RMB   1           0: VALID SYNDROME
16                  *           1: VALID GROUP
16                  *           2: RT DISPLAY
16                  *           3: UPDATE DISPLAY
16                  *           4: CLEAR DISPLAY
16                  *           5: SPACE FLAG
16                  *           6: NOT ON PROGRAM (AM)
16                  *           7: TA RETUNE DONE
16 000000a8      STAT3  RMB   1           0: NOT ON PROGRAM (FM)
16                  *           1: TEXTA/TEXTB BIT (RT)
16                  *           2: TA FLAG
16                  *           3: TP FLAG
16                  *           4: SHAFT DIRECTION
16                  *           5: SHAFT ROTATION
16                  *           6: UPDATE DATE
16                  *           7: SHAFT INTERRUPTS
16 000000a9      STAT4  RMB   1           0: DISPLAY (OR TA SWITCH) TRANSIENT
16                  *           1: SLEEP TIMER RUNNING
16                  *           2: TRAFFIC ENABLED
16                  *           3: ALARM DISPLAY
16                  *           4: ALARM ARMED
16                  *           5: ALARM SET-UP
16                  *           6: ALARM HOURS (SET-UP)
16                  *           7: VALID GROUP 14B RECEIVED
16 000000aa      STAT5  RMB   1           0: BAND CHANGE TIMEOUT
16                  *           1: RDS DISPLAYS
16                  *           2: SLEEP DISPLAY
16                  *           3: M/S 0: M, 1: S
16                  *           4: RETUNE FLAG (FREQUENCY MODE)
16                  *           5: TA INHIBIT FLAG (NVM)
16                  *           6: STORE MODE
16                  *           7: WEEKDAY ONLY ALARM
16
16 000000ab      STAT6  RMB   1           BAND/BANK (.MW STEP,COLON, .A1,A0,,E6)
16 000000ac      BCTO   RMB   1           BAND CHANGE TIMEOUT
16 000000ad      SONT   RMB   1           SHAFT DETENT COUNTER
16
16                  SECTION .RAM2.COMM
16 00000000      EQN     RMB   256           EQN DATA (16 NETWORKS)
16                  SECTION .RAM3.COMM
16 00000000      DISPP  RMB   16           LCD MODULE BUFFER
16 00000010      DISPP  RMB   16           CURRENT LCD MODULE CONTENTS
16 00000020      PSN    RMB   8            RADIOTEXT
16 00000028      RT     RMB   69
16
16                  SECTION .ROM3
17
18
19
20
21
22
23
24
25
26 00000000      >7c0000      TINTB   INC     THB2
27 00000003      >9600        LDAA     THB2
28 00000005      8104         CMPA     #4
29 00000007      2709         BEQ      FOUR
30 00000009      18ce1000     LDY      #$1000
31 0000000d      181d25bf     BCLR     S25,Y,SBF
32 00000011      3b          RTI
33
34 00000012      >7f0000      FOUR    CLR     THB2
35 00000015      >7c0000      INC      DISP1
36 00000018      >9600        LDAA     DISP1
37 0000001a      8108         CMPA     #8
38 0000001c      2307         BLS      NWR
39 0000001e      814e         CMPA     #78
40 00000020      2203         BHI      NWR
41 00000022      >7c0000      INC      DISP2
42 00000025      8158         CMPA     #88
43 00000027      2503         BLO      NWR2
44 00000029      >150004     BCLR     STAT2,$04
45 0000002c      >140008     BSET     STAT2,$08
46 0000002f      18ce1000     LDY      #$1000
47 00000033      181d25bf     BCLR     S25,Y,SBF
48
49 00000037      >7c0000      CLCK    INC     THB
50 0000003a      >7a0000      DEC     DIG2
51 0000003d      >7a0000      DEC     DIST
52 00000040      >7c0000      INC     RDSTO
53 00000043      >9600        LDAA     RDSTO
54 00000045      8150         CMPA     #80
55 00000047      251d         BLO      RDSOK
56 00000049      >150004     BCLR     STAT3,$04
57 0000004c      >13008003    BRCLR    STAT4,$80,N14B
58 00000050      >150080     BCLR     STAT4,$80
59 00000053      >7f0000      N14B   CLR     PI
60 00000056      >7f0001     CLR     PI+1
61 00000059      >12001009    BRSET    STAT,$10,RDSOK
62 0000005d      >140010     BSET     STAT,$10
63 00000060      >bd0000     JSR      CLREON
64 00000063      >140010     BSET     STAT2,$10

```

```

66
67
68
69
70
71
72 00000066 >9600
73 00000068 8108
74 0000006a 263e
75 0000006c >7f0000
76 0000006f >7c0000
77 00000072 >9600
78 00000074 8101
79 00000076 2603
80 00000078 >7a0000
81 0000007b 813c
82 0000007d 262b
83 0000007f >7f0000
84 00000082 >7c0000
85 00000085 >9600
86 00000087 813c
87 00000089 261f
88 0000008b >7f0000
89 0000008e >7c0000
90 00000091 >9600
91 00000093 8118
92 00000095 2613
93 00000097 >7f0000
94 0000009a >7c0002
95 0000009d 2608
96 0000009f >7c0001
97 000000a2 2603
98 000000a4 >7c0000
99 000000a7 >140040
100 000000aa 3b
101
102
103
104
105
106
107
108 000000ab >1200204
109 000000af >140020
110 000000b2 39
111 000000b3 >150020
112 000000b6 39
113
114
115
116
117
118
119
120 000000b7 18ce1000
121 000000bb 181f00807
122 000000c0 >13008003
123 000000c4 >7e0000
124 000000c7 >1200081c
125 000000cb 0d
126 000000cc 181e0a2001
127 000000d1 0c
128 000000d2 >790003
129 000000d5 >790002
130 000000d8 >790001
131 000000db >790000
132 000000de >1300010a
133 000000e2 >7a0000
134 000000e5 2701
135 000000e7 3b
136
137 000000e8 861a
138 000000ea >9700
139 000000ec >9601
140 000000ee >d600
141 000000f0 c403
142
143 000000f2 >13030104
144 000000f6 881b
145 000000f8 c803
146
147 000000fa >13030204
148 000000fe 888f
149 00000100 c803
150
151 00000102 >13030404
152 00000106 88a7
153 00000108 c802
154
155 0000010a >13030802
156 0000010e 88f7
157
158 00000110 >13031004
159 00000114 88ee
160 00000116 c801
161
162 00000118 >13032004
163 0000011c 88dc
164 0000011e c803

```

```

*****
*
*      Update clock.
*
*****

RDSOK  LDAA  TH8          EIGHTHS OF SECONDS
        CMPA  #8
        BNE  NOTC
        CLR  TH8
        INC  SEC
        LDAA  SEC
        CMPA  #1
        BNE  NOTS
        DEC  SLEPT
        CMPA  #60
        BNE  NOTC
        CLR  SEC
        INC  MIN
        LDAA  MIN
        CMPA  #60
        BNE  NOTC
        CLR  MIN
        INC  OUR
        LDAA  OUR
        CMPA  #24
        BNE  NOTC
        CLR  OUR
        INC  BMD*2
        BNE  NOTD
        INC  BMD*1
        BNE  NOTD
        INC  BMD
        NOTD BSET  STAT3,$40      UPDATE DATE
        NOTC RTI

*****
*
*      Toggle flashing colon control bit.
*
*****

TFCC  BRSET  STAT6,$20,CCBH
        BSET  STAT6,$20
        RTS

CCBH  BCLR  STAT6,$20
        RTS

*****
*
*      RDS clock (FM) or SHAFT (AM) interrupt.
*
*****

SDATA  LDY  #S100C
        BRCLR PORTA,Y,$08,RDSD    RDS INTERRUPTS ONLY (A3) ?
        BRCLR STAT3,$80,RDSD      NO, USE CONTROL BIT FROM BAND INPUTS
        JMP  SHAFT                NO, INTERRUPT FROM SHAFT
RDSD  BRSET  STAT6,$08,NOTFM      RDS, BUT IS IT AN FM BAND ?
        SEC
        BRSET PORTE,Y,$20,DHIGH   YES
        CLC
        ROL  DAT*3
        ROL  DAT*2
        ROL  DAT*1
        ROL  DAT
        BRCLR STAT2,$01,TRY2      BIT BY BIT CHECK ?
        DEC  BIT                  NO, WAIT FOR BIT 26
        BEQ  TRY1                 THIS TIME ?
        NOTFM RTI

TRY1  LDAA  #26
        STAA BIT
TRY2  LDAA  DAT*1
        LDAB DAT
        ANDB #3
        S03  BRCLR  DAT*3,$01,S13
        EORA  #S1B
        EORB  #S03
        S13  BRCLR  DAT*3,$02,S23
        EORA  #S8F
        EORB  #S03
        S23  BRCLR  DAT*3,$04,S33
        EORA  #SA7
        EORB  #S02
        S33  BRCLR  DAT*3,$08,S43
        EORA  #SF7
        S43  BRCLR  DAT*3,$10,S53
        EORA  #SEE
        EORB  #S01
        S53  BRCLR  DAT*3,$20,S63
        EORA  #SDC
        EORB  #S03

```

```

166
167
168
169
170
171
172 00000120 >13034004
173 00000124 8801
174 00000126 c802
175
176 00000128 >13038004
177 0000012c 88bb
178 0000012e c801
179
180 00000130 >13020104
181 00000134 8876
182 00000136 c803
183
184 00000138 >13020204
185 0000013c 8855
186 0000013e c803
187
188 00000140 >13020404
189 00000144 8813
190 00000146 c803
191
192 00000148 >13020804
193 0000014c 889f
194 0000014e c803
195
196 00000150 >13021004
197 00000154 8887
198 00000156 c802
199
200 00000158 >13022002
201 0000015c 88b7
202
203 0000015e >13024004
204 00000162 886e
205 00000164 c801
206
207 00000166 >13028004
208 0000016a 88dc
209 0000016c c802
210
211 0000016e >d700
212 00000170 >9701
213
214
215
216
217
218
219
220 00000172 >9600
221 00000174 8103
222 00000176 2764
223 00000178 8102
224 0000017a 2723
225 0000017c 8101
226 0000017e 2711
227 00000180 >720000
228
229 00000183 >9601
230 00000185 81d8
231 00000187 2632
232 00000189 >9600
233 0000018b 8103
234 0000018d 262c
235 0000018f 205d
236
237 00000191 >9601
238 00000193 81d4
239 00000195 2624
240 00000197 >9600
241 00000199 8103
242 0000019b 261e
243 0000019d 204f
244
245 0000019f >1202080c
246 000001a3 >9601
247 000001a5 815c
248 000001a7 2612
249 000001a9 >9600
250 000001ab 8102
251 000001ad 200a
252
253 000001af >9601
254 000001b1 81cc
255 000001b3 2606
256 000001b5 >9600
257 000001b7 8103
258 000001b9 2733

```

```

*****
*                                     *
*      Calculate syndrome (cont.).    *
*                                     *
*****

```

```

S63  BRCLR  DAT+3,$40,S73
      EORA  #S01
      EORB  #S02

S73  BRCLR  DAT+3,$80,S02
      EORA  #S8B
      EORB  #S01

S02  BRCLR  DAT+2,$01,S12
      EORA  #S76
      EORB  #S03

S12  BRCLR  DAT+2,$02,S22
      EORA  #S55
      EORB  #S03

S22  BRCLR  DAT+2,$04,S32
      EORA  #S13
      EORB  #S03

S32  BRCLR  DAT+2,$08,S42
      EORA  #S9F
      EORB  #S03

S42  BRCLR  DAT+2,$10,S52
      EORA  #S87
      EORB  #S02

S52  BRCLR  DAT+2,$20,S62
      EORA  #SB7

S62  BRCLR  DAT+2,$40,S72
      EORA  #S6E
      EORB  #S01

S72  BRCLR  DAT+2,$80,FIN
      EORA  #SDC
      EORB  #S02

FIN  STAB   SWN
      STAA  SWN+1

```

```

*****
*                                     *
*      Check for syndromes A, B, C & C'. *
*                                     *
*****

```

```

      LDAA  LEV
      CMPA  #3
      BEQ   TRYD
      CMPA  #2
      BEQ   TRYC
      CMPA  #1
      BEQ   TRYB
      CLR   LEV

TRYA  LDAA  SWN+1
      CMPA  #SD8
      BNE  NOTV
      LDAA  SWN
      CMPA  #S03
      BNE  NOTV
      BRA  VALID

      LDAA  SWN+1
      CMPA  #SD4
      BNE  NOTV
      LDAA  SWN
      CMPA  #S03
      BNE  NOTV
      BRA  VALID

TRYC  BRSET  TMPGRP+2,$08,TRYCD
      LDAA  SWN+1
      CMPA  #SSC
      BNE  NOTV
      LDAA  SWN
      CMPA  #S02
      BRA  VC

TRYCD LDAA  SWN+1
      CMPA  #SOC
      BNE  NOTV
      LDAA  SWN
      CMPA  #S03
      BEQ   VALID

VC

```

```

260
261
262
263
264
265
266
267 000001bb >7f0000
268 000001be >9600
269 000001c0 8129
270 000001c2 2410
271 000001c4 >150001
272 000001c7 810a
273 000001c9 230d
274 000001cb >7a0000
275 000001ce 2607
276 000001d0 861a
277 000001d2 >9700
278 000001d4 >7a0000
279 000001d7 3b
280 000001d8 >140010
281 000001db 3b
282
283 000001dc >9601
284 000001de 8158
285 000001e0 26d9
286 000001e2 >9600
287 000001e4 8102
288 000001e6 26d3
289 000001e8 >140002
290 000001eb >150010
291
292 000001ee >12000107
293 000001f2 8626
294 000001f4 >9700
295 000001f6 >140001
296 000001f9 >9600
297 000001fb 8138
298 000001fd 2204
299 000001ff 8b04
300 00000201 >9700
301 00000203 ce0000
302 00000206 >d600
303 00000208 59
304 00000209 3a
305 0000020a >7c0000
306 0000020d 861a
307 0000020f >9700
308 00000211 >760000
309 00000214 >760001
310 00000217 >760002
311 0000021a >760000
312 0000021d >760001
313 00000220 >760002
314 00000223 >9602
315 00000225 >a701
316 00000227 >9601
317 00000229 >a700
318 0000022b >130002ac
319 0000022f ce0008
320 00000232 >a6ff
321 00000234 >a7ff
322 00000236 09
323 00000237 26f9
324
325
326
327
328
329
330
331
332
333 00000239 >9600
334 0000023b >9100
335 0000023d 2606
336 0000023f >9601
337 00000241 >9101
338 00000243 270e
339 00000245 >9600
340 00000247 >9700
341 00000249 >9601
342 0000024b >9701
343 0000024d >b30000
344 00000250 >140010
345
346
347
348
349
350
351
352
353 00000253 >9602
354 00000255 >9700
355 00000257 >13000405
356 0000025b >140008
357 0000025e 2003
358 00000260 >150008
359 00000263 >9603
360 00000265 >760000
361 00000268 46
362 00000269 44
363 0000026a 44
364 0000026b 44
365 0000026c 44
366 0000026d >9700

```

```

*****
*
*   Invalid syndrome handling, check for
*   block 4 and save group data if valid.
*
*****

NOTV  CLR  LEV          RESTART AT BLOCK 1
      LDAA  CNF
      CMPA  #41        CONFIDENCE 41 OR GREATER ?
      BHS   DECC        BIT BY BIT SYNDROME CHECK
      BCLR  STAT2,$01
      CMPA  #10        CONFIDENCE 10 OR LESS ?
      BLS   SKPDC
      DEC   BIT
      BNE   NNOW        USE BIT COUNTER TO SLOW CONFIDENCE
      LDAA  #26        DROP DURING BIT BY BIT ATTEMPT TO
      STAA  BIT         RE-SYNCRONISE
      DECC  DEC  CNF
      NNOW  RTI
      SKPDC BSET  STAT2,$10  10 OR LESS, INITIALISE DISPLAY
      NOT4  RTI

TRYD  LDAA  SYN+1
      CMPA  #558
      BNE  NOTV
      LDAA  SYN
      CMPA  #502
      BNE  NOTV
      BSET  STAT2,$02      GROUP COMPLETE
      BCLR  STAT,$10      RE-ENABLE RDS DATA CLEARING

VALID  BRSET  STAT2,$01,VLD
      LDAA  #38
      STAA  CNF
      BSET  STAT2,$01
VLD    LDAA  CNF
      CMPA  #56
      BHI   NMR
      ADDA  #4
      STAA  CNF
NMR    LDX   #0
      LDAB  LEV
      ROLB
      ARX
      INC   LEV
      LDAA  #26
      STAA  BIT
      ROR   DAT
      ROR   DAT+1
      ROR   DAT+2
      ROR   DAT
      ROR   DAT+1
      ROR   DAT+2
      LDAA  DAT+2
      STAA  TMPGRP+1,X
      LDAA  DAT+1
      STAA  TMPGRP,X
      BRCLR STAT2,$02,NOT4  GROUP COMPLETE ?
XFER  LDX   #8
TXLP  LDAA  TMPGRP-1,X
      STAA  GROUP-1,X
      DEX
      BNE   TXLP
      RTL

*****
*
*   Update PI code, initialise if changed.
*   All block 1s used, block 3s not used.
*
*****

PROC  LDAA  GROUP        COMPARE PI WITH PREVIOUS
      CMPA  PI
      BNE  INDX
      LDAA  GROUP+1
      CMPA  PI+1
      BE2  PTYL
INDX  LDAA  GROUP
      STAA  PI
      LDAA  GROUP+1
      STAA  PI+1
      JSR  CLREON
      BSET  STAT2,$10      INITIALISE DISPLAY DATA

*****
*
*   Update PTY and TP.
*   All block 2s used, not block 4 (grp 15B).
*
*****

PTYL  LDAA  GROUP+2
      STAA  TMP1
      BRCLR TMP1,$04,TPL1  TP HIGH ?
      BSET  STAT3,$08      YES, FLAG HIGH
      BRA   TPL
TPL1  BCLR  STAT3,$08
      LDAA  GROUP+3
      ROR   TMP1
      RORA
      LSRA
      LSRA
      LSRA
      LSRA
      STAA  PTY

```

```

368
369
370
371
372
373
374
375
376
377
378
379
380
381
382
383
384
385
386
387
388 0000026f >9602
389 00000271 84f8
390 00000273 270a
391 00000275 8108
392 00000277 2706
393
394 00000279 81f8
395 0000027b 2713
396 0000027d 2077
397
398 0000027f >d603
399 00000281 c403
400 00000283 58
401 00000284 >ce0000
402 00000287 3a
403 00000288 >9606
404 0000028a a700
405 0000028c >9607
406 0000028e a701
407
408 00000290 >7f0000
409 00000293 >12031014
410 00000297 >150004
411 0000029a >13080010
412 0000029e >1200010c
413 000002a2 >150080
414 000002a5 8604
415 000002a7 >9700
416 000002a9 2003
417 000002ab >140004
418
419
420
421
422
423
424
425 000002ae >d603
426 000002b0 c403
427 000002b2 >9603
428 000002b4 8404
429 000002b6 5d
430 000002b7 2609
431 000002b9 >150008
432 000002bc 4d
433 000002bd 2703
434 000002bf >140008
435 000002c2 c101
436 000002c4 2609
437 000002c6 >150004
438 000002c9 4d
439 000002ca 2703
440 000002cc >140004
441 000002cf c102
442 000002d1 2609
443 000002d3 >150002
444 000002d6 4d
445 000002d7 2703
446 000002d9 >140002
447 000002dc c103
448 000002de 2609
449 000002e0 >150001
450 000002e3 4d
451 000002e4 2703
452 000002e6 >140001
453
454 000002e9 >150008
455 000002ec >13030803
456 000002f0 >140008
457 000002f3 >7e0000

```

```

*****
*
*      Groups handled.
*
*      All          PI, PTY & TP
*      0 A & B      TA, PS, DI, M/S (not AF)
*      1 A & B      PIN
*      2 A          RT
*      4 A          CT
*      14 A & B     ECN (TP only)
*      15 B         TA
*
*****
*
*      Process groups 0 & 15B (PS & TA).
*
*****
*
*      LDAA      GROUP=2
*      ANDA      #5F8
*      BEQ      GRP0          GROUP 0A
*      CMPA      #508          GROUP 0B
*      BEQ      GRP0
*
*      TGRP15    CMPA      #5F8          GROUP 15B
*                  BEQ      TACK
*                  BRA      PROC1
*
*      GRP0      LDAB      GROUP=3          GROUP 0 - PS & TA
*                  ANDB      #503
*                  LSLB
*                  LDX      #PSN
*                  ABX
*                  LDAA      GROUP=6
*                  STAA      0,X
*                  LDAA      GROUP=7
*                  STAA      1,X
*
*      TACK      CLR      ROSTO          RDS OK, RESET TIME-OUT
*                  BRSET   GROUP=3,S10,TAH    TA HIGH ?
*                  BCLR    STAT3,S04          NO, TA FLAG LOW
*                  BRCLR   STAT2,S08,NTD      SWITCHED TO TA ?
*                  BRSET   STAT4,S01,NTD      3s LOCKOUT TIMEOUT FINISHED ?
*                  BCLR    STAT4,S80          YES, SWITCH BACK TO NORMAL PROG.
*                  LDAA      #4
*                  STAA      RDSRET
*                  BRA      NTD
*      TAH      BSET      STAT3,S04          YES, TA FLAG HIGH
*
*****
*
*      Process groups 0 & 15B (DI & M/S).
*
*****
*
*      NTD      LDAB      GROUP=3          DI
*                  ANDB      #3
*                  LDAA      GROUP=3
*                  ANDA      #504
*                  TSTB
*                  BNE      NOT0
*                  BCLR    DI,#8
*                  TSTA
*                  BEQ      NOT0
*                  BSET    DI,#8
*      NOT0     CMPB      #1
*                  BNE      NOT1
*                  BCLR    DI,#4
*                  TSTA
*                  BEQ      NOT1
*                  BSET    DI,#4
*      NOT1     CMPB      #2
*                  BNE      NOT2
*                  BCLR    DI,#2
*                  TSTA
*                  BEQ      NOT2
*                  BSET    DI,#2
*      NOT2     CMPB      #3
*                  BNE      NOT3
*                  BCLR    DI,#1
*                  TSTA
*                  BEQ      NOT3
*                  BSET    DI,#1
*
*      NOT3     BCLR      STAT5,S08          N/S
*                  BRCLR   GROUP=3,#508,MSZ
*                  BSET     STAT5,S08
*      MSZ      JMP      OUT1

```

```

260
261
262
263
264
265
266
267 000001bb >7f0000
268 000001be >9600
269 000001c0 8129
270 000001c2 2410
271 000001c4 >150001
272 000001c7 810a
273 000001c9 230d
274 000001cb >7a0000
275 000001ce 2607
276 000001d0 861a
277 000001d2 >9700
278 000001d4 >7a0000
279 000001d7 3b
280 000001d8 >140010
281 000001db 3b
282
283 000001dc >9601
284 000001de 8158
285 000001e0 26d9
286 000001e2 >9600
287 000001e4 8102
288 000001e6 26d3
289 000001e8 >140002
290 000001eb >150010
291
292 000001ee >12000107
293 000001f2 8626
294 000001f4 >9700
295 000001f6 >140001
296 000001f9 >9600
297 000001fb 8138
298 000001fd 2204
299 000001ff 8b04
300 00000201 >9700
301 00000203 ce0000
302 00000206 >d600
303 00000208 59
304 00000209 3a
305 0000020a >7c0000
306 0000020d 861a
307 0000020f >9700
308 00000211 >760000
309 00000214 >760001
310 00000217 >760002
311 0000021a >760000
312 0000021d >760001
313 00000220 >760002
314 00000223 >9602
315 00000225 >a701
316 00000227 >9601
317 00000229 >a700
318 0000022b >130002ac
319 0000022f ce0008
320 00000232 >a6ff
321 00000234 >a7ff
322 00000236 09
323 00000237 26f9
324
325
326
327
328
329
330
331
332
333 00000239 >9600
334 0000023b >9100
335 0000023d 2606
336 0000023f >9601
337 00000241 >9101
338 00000243 270e
339 00000245 >9600
340 00000247 >9700
341 00000249 >9601
342 0000024b >9701
343 0000024d >b30000
344 00000250 >140010
345
346
347
348
349
350
351
352
353 00000253 >9602
354 00000255 >9700
355 00000257 >13000405
356 0000025b >140008
357 0000025e 2003
358 00000260 >150008
359 00000263 >9603
360 00000265 >760000
361 00000268 46
362 00000269 44
363 0000026a 44
364 0000026b 44
365 0000026c 44
366 0000026d >9700

```

```

*****
*
*   Invalid syndrome handling, check for
*   block 4 and save group data if valid.
*
*****

NOTV  CLR  LEV          RESTART AT BLOCK 1
      LDAA CONF
      CMPA #41         CONFIDENCE 41 OR GREATER ?
      BHS  DECC        BIT BY BIT SYNDROME CHECK
      BCLR STAT2,$01
      CMPA #10         CONFIDENCE 10 OR LESS ?
      BLS  SKPDC
      DEC  BIT
      BNE NNOW         USE BIT COUNTER TO SLOW CONFIDENCE
      LDAA #26         DROP DURING BIT BY BIT ATTEMPT TO
      STAA BIT         RE-SYNCHRONISE
      DECC DEC         CNF
      NNOW RTI
      SKPDC BSET      STAT2,$10      10 OR LESS, INITIALISE DISPLAY
      NOT4 RTI

TRYD  LDAA SYN+1
      CMPA #558
      BNE NOTV
      LDAA SYN
      CMPA #502
      BNE NOTV
      BSET STAT2,$02      GROUP COMPLETE
      BCLR STAT,$10      RE-ENABLE RDS DATA CLEARING

VALID BRSET STAT2,$01,VLD
      LDAA #38
      STAA CNF
      BSET STAT2,$01
VLD   LDAA CNF
      CMPA #56
      BHI NMR
      ADDA #4
      STAA CNF
NMR   LDX  #0
      LDAB LEV
      ROLB
      ARX
      INC  LEV
      LDAA #26
      STAA BIT
      ROR  DAT
      ROR  DAT+1
      ROR  DAT+2
      ROR  DAT
      ROR  DAT+1
      ROR  DAT+2
      LDAA DAT+2
      STAA TMPGRP+1,X
      LDAA DAT+1
      STAA TMPGRP,X
      BRCLR STAT2,$02,NOT4      GROUP COMPLETE ?
XFER  LDX  #8
TXLP  LDAA TMPGRP-1,X
      STAA GROUP-1,X
      DEX
      BNE TXLP
      RTL

*****
*
*   Update PI code, initialise if changed.
*   All block 1s used, block 3s not used.
*
*****

PROC  LDAA GROUP          COMPARE PI WITH PREVIOUS
      CMPA PI
      BNE INDX
      LDAA GROUP+1
      CMPA PI+1
      BE2 PTYL
      INDX LDAA GROUP
      STAA PI              DIFFERENT, SAVE NEW PI
      LDAA GROUP+1
      STAA PI+1
      JSR  CLREON
      BSET STAT2,$10      INITIALISE DISPLAY DATA

*****
*
*   Update PTY and TP.
*   All block 2s used, not block 4 (grp 15B).
*
*****

PTYL  LDAA GROUP+2
      STAA ITMP1
      BRCLR ITMP1,$04,TP1      TP HIGH ?
      BSET STAT3,$08          YES, FLAG HIGH
      BRA  TPL
      BCLR STAT3,$08          NO, FLAG LOW
      TPL  LDAA GROUP+3
      ROR  ITMP1
      RORA
      LSRA
      LSRA
      LSRA
      LSRA
      STAA PTY

```

```

368
369
370
371
372
373
374
375
376
377
378
379
380
381
382
383
384
385
386
387
388 0000026f >9602
389 00000271 94f8
390 00000273 270a
391 00000275 8108
392 00000277 2706
393
394 00000279 81f8
395 0000027b 2713
396 0000027d 2077
397
398 0000027f >d603
399 00000281 c403
400 00000283 58
401 00000284 >ce0000
402 00000287 3a
403 00000288 >9606
404 0000028a a700
405 0000028c >9607
406 0000028e a701
407
408 00000290 >7f0000
409 00000293 >12031014
410 00000297 >150004
411 0000029a >13008010
412 0000029e >1200010c
413 000002a2 >150080
414 000002a5 8604
415 000002a7 >9700
416 000002a9 2003
417 000002ab >140004
418
419
420
421
422
423
424
425 000002ae >d603
426 000002b0 c403
427 000002b2 >9603
428 000002b4 8404
429 000002b6 5d
430 000002b7 2609
431 000002b9 >150008
432 000002bc 4d
433 000002bd 2703
434 000002bf >140008
435 000002c2 c101
436 000002c4 2609
437 000002c6 >150004
438 000002c9 4d
439 000002ca 2703
440 000002cc >140004
441 000002cf c102
442 000002d1 2609
443 000002d3 >150002
444 000002d6 4d
445 000002d7 2703
446 000002d9 >140002
447 000002db c103
448 000002de 2609
449 000002e0 >150001
450 000002e3 4d
451 000002e4 2703
452 000002e6 >140001
453
454 000002e9 >150008
455 000002ec >13030803
456 000002f0 >140008
457 000002f3 >7e0000

```

```

.....
*
* Groups handled.
*
* All PI, PTY & TP
* 0 A & B TA, PS, DI, M/S (not AF)
* 1 A & B PIN
* 2 A RT
* 4 A CT
* 14 A & B ECM (TP only)
* 15 B TA
*
.....
*
* Process groups 0 & 15B (PS & TA).
*
.....
*
* LDAA GROUP=2
* ANDA #SF8
* BEQ GRP0 GROUP 0A
* CMPA #S08 GROUP 0B
* BEQ GRP0
*
* TGRP15 CMPA #SF8 GROUP 15B
* BEQ TACK
* BRA PROCI
*
* GRP0 LDAB GROUP=3 GROUP 0 - PS & TA
* ANDB #S03
* LSLB
* LDX #PSN
* AEX
* LDAA GROUP=6
* STAA 0.X
* LDAA GROUP=7
* STAA 1.X
*
* TACK CLR ROSTD RDS OK, RESET TIME-OUT
* BRSET GROUP=3,S10,TAH TA HIGH ?
* BCLR STAT3,S04 NO, TA FLAG LOW
* BRCLR STAT2,S08,NTD SWITCHED TO TA ?
* BRSET STAT4,S01,NTD 3s LOCKOUT TIMEOUT FINISHED ?
* BCLR STAT4,S08 YES, SWITCH BACK TO NORMAL PROG.
* LDAA #4
* STAA RARET
* BRA NTD
* BSET STAT3,S04 YES, TA FLAG HIGH
*
*
*
* Process groups 0 & 15B (DI & M/S).
*
*
*
* NTD LDAB GROUP=3 DI
* ANDB #3
* LDAA GROUP=3
* ANDA #S04
* TSTB
* BNE NOT0
* BCLR DI,#8
* TSTA
* BEQ NOT0
* BSET DI,#8
* NOT0 CMPB #1
* BNE NOT1
* BCLR DI,#4
* TSTA
* BEQ NOT1
* BSET DI,#4
* NOT1 CMPB #2
* BNE NOT2
* BCLR DI,#2
* TSTA
* BEQ NOT2
* BSET DI,#2
* NOT2 CMPB #3
* BNE NOT3
* BCLR DI,#1
* TSTA
* BEQ NOT3
* BSET DI,#1
*
* NOT3 BCLR STAT3,S08 N/S
* BRCLR GROUP=3,#S08,MSZ
* BSET STAT3,S08
* MSZ JMP OUT1

```



459
460
461
462
463
464
465 000002f6 8110
466 000002f8 2704
467 000002fa 8118
468 000002fc 260b
469
470 000002fe >9606
471 00000300 >9700
472 00000302 >9607
473 00000304 >9701
474
475 00000306 >7e0000
476
477
478
479
480
481
482
483
484 00000309 8120
485 0000030b 2634
486
487 0000030d >12031009
488 00000311 >1200020f
489 00000315 >140002
490 00000318 2007
491 0000031a >13000206
492 0000031e >150002
493 00000321 >bd0000
494
495 00000324 >d603
496 00000326 c40f
497 00000328 58
498 00000329 58
499 0000032a >ce0000
500 0000032d 3a
501 0000032e >9604
502 00000330 a705
503 00000332 >9605
504 00000334 a706
505 00000336 >9606
506 00000338 a707
507 0000033a >9607
508 0000033c a708
509 0000033e >7e0000

```
*****
*                                     *
*      Process group 1 (PIN) .      *
*                                     *
*****

PROC1  CMPA  #S10          GROUP 1A
        BEQ  GRP1
        CMPA  #S18          GROUP 1B
        BNE  PROC2

GRP1    LDAA  GROUP+6
        STAA  PIN
        LDAA  GROUP+7
        STAA  PIN+1

        JMP  OUT1

*****
*                                     *
*      Process group 2A (RT) .      *
*      Group 2B not handled.      *
*                                     *
*****

PROC2  CMPA  #S20          GROUP 2A
        BNE  PROC4

GRP2    BRSET GROUP+3,S10,TEXTB
TEXTA   BRSET STAT3,S02,NCH
        BSET  STAT3,S02
        BRA   LCDINI
TEXTB   BRCLR  STAT3,S02,NCH
        BCLR  STAT3,S02
LCDINI  JSR    INITR          CLEAR RT

NCH     LDAB  GROUP+5          GROUP 2A - RT
        ANDB  #S0F
        LSLB
        LSLB
        LDX   #RT
        LDAA  GROUP+4
        STAA  5,X
        LDAA  GROUP+5
        STAA  6,X
        LDAA  GROUP+6
        STAA  7,X
        LDAA  GROUP+7
        STAA  8,X
        JMP  OUT1
```

```

511
512
513
514
515
516
517 00000341 8140
518 00000343 2703
519 00000345 >7e0000
520
521 00000348 >9603
522 0000034a 46
523 0000034b 8401
524 0000034d >9700
525 0000034f >9604
526 00000351 46
527 00000352 >9701
528
529 00000354 >9606
530 00000356 >7e0005
531 00000359 46
532 0000035a 44
533 0000035b 44
534 0000035c 44
535 0000035d >9700
536
537 0000035f >9605
538 00000361 >9702
539
540 00000363 >9606
541 00000365 >780007
542 00000368 49
543 00000369 >780007
544 0000036c 49
545 0000036d 843f
546 0000036f >9700
547 00000371 >7f0000
548 00000374 >7f0000
549 00000377 >140040
550
551
552
553
554
555
556
557 0000037a >d607
558 0000037c 58
559 0000037d 2764
560 0000037f 2436
561
562 00000381 54
563 00000382 54
564 00000383 54
565 00000384 54
566 00000385 240d
567 00000387 >9600
568 00000389 801e
569 0000038b 2a05
570 0000038d 8b3c
571 0000038f >7a0000
572 00000392 >9700
573
574 00000394 >d000
575 00000396 53
576 00000397 4c
577 00000398 2a19
578 0000039a c218
579 0000039c >d700
580
581 0000039e >7d0002
582 000003a1 260b
583 000003a3 >7d0001
584 000003a6 2603
585 000003a8 >7a0000
586 000003ab >7a0001
587 000003ae >7a0002
588 000003b1 2030
589
590 000003b3 >d700
591 000003b5 202c
592
593 000003b7 54
594 000003b8 54
595 000003b9 54
596 000003ba 54
597 000003bb 240f
598 000003bd 861e
599 000003bf >9b00
600 000003c1 813b
601 000003c3 2305
602 000003c5 803c
603 000003c7 >7e0000
604 000003ca >9700
605
606 000003cc >db00
607 000003ce c117
608 000003d0 230f
609 000003d2 c018
610 000003d4 >7e0002
611 000003d7 2608
612 000003d9 >7e0001
613 000003dc 2603
614 000003de >7e0000
615 000003e1 >d700
616 000003e3 >150002
617 000003e6 3b

```

```

*****
*
*   Process group 4A (CT) .
*
*****

PROC4  CHPA  #540          GROUP 4A - CT
      BEQ  GRP4
      JMP  PROC14

GRP4   LDAA  GROUP+3
      RORA
      ANDA  #501
      STAA  BMJD          MJD MS BIT
      LDAA  GROUP+4
      RORA
      STAA  BMJD+1        MJD MSD
      LDAA  GROUP+6
      RORA
      STAA  GROUP+5
      LSR   GROUP+5
      LSR   GROUP+5
      LSR   GROUP+5
      LSR   GROUP+5
      LSR   GROUP+5
      LSR   GROUP+5
      STAA  OUR
      LDAA  GROUP+5
      STAA  BMJD+2        MJD LSD
      LDAA  GROUP+6
      LSL   GROUP+7
      ROLA  GROUP+7
      LSL   GROUP+7
      ROLA  GROUP+7
      ANDA  #53F
      STAA  MIN
      CLR   SEC
      CLR   TH8
      BSET  STAT3,540     UPDATE MJD

*****
*
*   Local time difference adjustment.
*
*****

LOCAL  LDAB  GROUP+7
      LSLB
      BEQ  OUT1
      BCC  POS
      NEG  LSRB
      LSRB
      LSRB
      LSRB
      BCC  NOTHN
      LDAA  MIN
      SUBA  #30
      BPL  LT60
      ADDA  #60
      DEC  OUR
      LT60  STAA  MIN
      NOTHN  SUBB  OUR
      CQCB
      INCA
      BPL  ZCH
      AIDB  #24
      STAB  OUR
      TST  BMJD+2
      BNE  TT2
      TST  BMJD+1
      BNE  TT1
      DEC  BMJD
      TT1  DEC  BMJD+1
      TT2  DEC  BMJD+2
      BRA  OUT1
      ZCH  STAB  OUR
      BRA  OUT1
      POS  LSRB
      LSRB
      LSRB
      BCC  NOTHP
      LDAA  #30
      ADDA  MIN
      CHPA  #59
      BLS  HDON
      SUBA  #60
      INC  OUR
      HDON  STAA  MIN
      NOTHP  AIDB  OUR
      CHPB  #23
      BLS  ADDCN
      SUBB  #24
      INC  BMJD+2
      BNE  ADDCN
      INC  BMJD+1
      BNE  ADDCN
      INC  BMJD
      ADDCN  INC  BMJD
      OUT1  BCLR  STAT2,502
      RTI

ADJUSTMENT ?
YES, POSITIVE ?
NO, NEGATIVE
HOURS IN B
1/2 HOUR ?
YES
SUBTRACT 30 MINUTES
UNDERFLOW ?
YES, ADD 60 MINUTES
AND SUBTRACT 1 HOUR
NEGATIVE HOUR OFFSET, MINUS UTC HOURS
WRONG WAY ROUND SO COMPLEMENT
AND INCREMENT
UNDERFLOW ?
YES, ADD 24 HOURS
AND SUBTRACT A DAY
LSB WILL UNDERFLOW ?
YES
MSB WILL UNDERFLOW ?
YES DECREMENT MS BIT
DECREMENT MSB
DECREMENT LSB
POSITIVE ADJUSTMENT
HOURS IN B
HALF HOUR ?
YES, ADD 30 MINUTES
OVERFLOW ?
YES, SUBTRACT 60 MINUTES
AND ADD AN HOUR
HOUR OFFSET, ADD UTC HOURS
OVERFLOW ?
YES, SUBTRACT 24 HOURS
AND ADD A DAY
GROUP HANDLED, CLEAR FLAG

```

619					
620					
621					
622					
623					
624					
625	000003e7	81e0	PROC14	CMFA	#SE0
626	000003e9	2703		BEQ	GRP14A
627	000003eb	>7e0000		JMP	PR14B
628					
629	000003ee	>13031003	GRP14A	BRCLR	GROUP+3, \$10, TPLO
630				JMP	OUT2
631	000003f2	01		NOP	
632	000003f3	01		NOP	
633	000003f4	01		NOP	
634	000003f5	>7f0000	TPLO	CLR	ITMP1
635	000003f8	>d600	LPIL	LDAB	ITMP1
636	000003fa	>ce0000		LDX	#EON
637	000003fd	3a		AEK	
638	000003fe	a600		LDAA	0, X
639	00000400	>9106		CMFA	GROUP+6
640	00000402	2663		BNE	NOTH
641	00000404	a601		LDAA	1, X
642	00000406	>9107		CMFA	GROUP+7
643	00000408	265d		BNE	NOTH
644					
645				LDAA	GROUP+3
646				ANDA	#S10
647				STAA	SB, X
648					
649	0000040a	>d603		LDAB	GROUP+3
650	0000040c	c40f		ANDB	#S0F
651	0000040e	c104		CMFB	#4
652	00000410	2411		BHS	NPS
653	00000412	58		LSLB	
654	00000413	>db00		ADDB	ITMP1
655	00000415	>ce0000		LDX	#EON
656	00000418	3a		AEK	
657	00000419	>9604		LDAA	GROUP+4
658	0000041b	a702		STAA	2, X
659	0000041d	>9605		LDAA	GROUP+5
660	0000041f	a703		STAA	3, X
661	00000421	20c0		BRA	OUT1
662					
663	00000423	c104	NPS	CMFB	#4
664	00000425	262c		BNE	TRYPIN
665					
666	00000427	>9604		LDAA	GROUP+4
667					
668	00000429	81fa		CMFA	#250
669	0000042b	2616		BNE	NMLW
670	0000042d	a60c		LDAA	SC, X
671	0000042f	81ff		CMFA	#SFF
672	00000431	2777		BEQ	OUT2
673	00000433	a60e		LDAA	SE, X
674	00000435	81ff		CMFA	#SFF
675	00000437	2671		BNE	OUT2
676	00000439	86fa		LDAA	#250
677	0000043b	a70e		STAA	SE, X
678	0000043d	>9605		LDAA	GROUP+5
679	0000043f	a70f		STAA	SF, X
680	00000441	2067		BRA	OUT2
681					
682	00000443	81e0	NMLW	CMFA	#224
683	00000445	250a		BLO	TOOLS
684	00000447	81ff		CMFA	#249
685	00000449	2206		BHI	TOOLS
686	0000044b	a70c		STAA	SC, X
687	0000044d	>9605		LDAA	GROUP+5
688	0000044f	a70d		STAA	SD, X
689	00000451	2057		BRA	OUT2
690					
691			*TRYPTY	CMFB	#S0D
692				BNE	TRYPIN
693				LDAA	GROUP+4
694				LSRA	
695				LSRA	
696				LSRA	
697				LDAB	ITMP1
698				LDX	#EON
699				AEK	
700				STAA	SA, X
701				BRA	OUT2
702					
703	00000453	c10e	TRYPIN	CMFB	#S0E
704	00000455	2653		BNE	OUT2
705	00000457	>d600		LDAB	ITMP1
706	00000459	>ce0000		LDX	#EON
707	0000045c	3a		AEK	
708	0000045d	>9604		LDAA	GROUP+4
709	0000045f	a70a		STAA	SA, X
710	00000461	>9605		LDAA	GROUP+5
711	00000463	a70b		STAA	SB, X
712	00000465	2043		BRA	OUT2
713					
714	00000467	81ff	NOTH	CMFA	#SFF
715	00000469	260a		BNE	NOTH1
716	0000046b	>9606		LDAA	GROUP+6
717	0000046d	a700		STAA	0, X
718	0000046f	>9607		LDAA	GROUP+7
719	00000471	a701		STAA	1, X
720	00000473	2035		BRA	OUT2
721					
722	00000475	>9600	NOTH1	LDAA	ITMP1
723	00000477	8b10		ADCA	#16
724	00000479	>9700		STAA	ITMP1
725	0000047b	272d		BEQ	OUT2
726	0000047d	>7e0000		JMP	LPIL
727					
728	00000480	81e8	PR14B	CMFA	#CEH
729	00000482	2626		BNE	OUT2

```

731
732
733
734
735
736
737 00000484 >9600
738 00000486 840c
739 00000488 8104
740 0000048a 261e
741 0000048c 181e082019
742 00000491 >13030815
743 00000495 >13031011
744 00000499 >dc06
745 0000049b >dd00
746 0000049d >12000406
747 000004a1 8601
748 000004a3 >9700
749 000004a5 2003
750 000004a7 >140080
751 000004aa >150002
752 000004ad 3b
753
754
755
756
757
758
759
760 000004ae >d600
761 000004b0 c110
762 000004b2 2501
763 000004b4 5f
764 000004b5 8610
765 000004b7 3d
766 000004b8 >ce0000
767 000004bb 3a
768 000004bc >18ce0000
769 000004c0 a600
770 000004c2 18a700
771 000004c5 1808
772 000004c7 08
773 000004c8 >188c0010
774 000004cc 25f2
775 000004ce 39
776
777
778
779
780
781
782
783 000004cf >9602
784 000004d1 >9702
785 000004d3 >9601
786 000004d5 >9701
787 000004d7 >9600
788 000004d9 >9700
789 000004db >ce0000
790 000004de >df00
791 000004e0 >bd0000
792 000004e3 >7c0008
793 000004e6 >ce0000
794 000004e9 >bd0000
795 000004ec 8611
796 000004ee >9700
797 000004f0 >740000
798 000004f3 >760001
799 000004f6 >760002
800 000004f9 2408
801 000004fb >ce0000
802 000004fe >df00
803 00000500 >bd0000
804 00000503 >ce0000
805 00000506 >df00
806 00000508 >bd0000
807 0000050b >7a0000
808 0000050e 26e0
809 00000510 >150040
810 00000513 >7e0000
811
812
813
814
815
816
817
818 00000516 >bd0000
819 00000519 86a8
820 0000051b >bd0000
821 0000051e >ce0000
822 00000521 >bd0000
823 00000524 181c0420
824 00000528 a608
825 0000052a 81ff
826 0000052c 2602
827 0000052e 862d
828 00000530 >bd0000
829 00000533 08
830 00000534 >8c0008
831 00000537 26e8
832 00000539 39

```

```

*****
*
* Process group 148 (switch to TA).
*
*****
GRP14B LDAA STAT3
      ANDA #S0C
      CMPA #S04
      BNE OUT2
      BRSET PORTD,Y,S20,OUT2
      BRCLR GROUP+3,S08,OUT2
      BRCLR GROUP+3,S10,OUT2
TAOH LD LD GROUP+6
      STD PION
      BRSET STAT4,S04,OUT3
      LDAA #1
      STAA REARET
      BRA OUT2
OUT3 BSET STAT4,S80
OUT2 BCLR STAT2,S02
      RTI

```

```

*****
*
* Update PTY display mode.
*
*****
PTD LDAB PTY
      CMPB #16
      BLO BOK
      CLRB
BOK LDAA #16
      MUL
      LDX #PTYT
      AEX
      LDY #DISP
LCD3 LDAA 0,X
      STAA 0,Y
      INY
      INX
      CPY #DISP+16
      BLO LCD3
      RTS

```

```

*****
*
* Extract MJD and convert to decimal.
*
*****
MJDRT LDAA BMJD+2
      STAA YR+2
      LDAA BMJD+1
      STAA YR+1
      LDAA BMJD
      STAA YR
      LDX #R
      STX NUM1
      JSR CLRAS
      INC R+ND-1
      LDX #MJD
      JSR CLRAS
      LDAA #17
      STAA W6
      LSR YR
      ROR YR+1
      ROR YR+2
      BCC NXTJ
      LDX #MJD
      STX NUM2
      JSR ADD
      LDX #R
      STX NUM2
      JSR ADD
      DEC W6
      BNE LOORJ
      BCLR STAT3,S4C
      JMP MJDC

```

```

*****
*
* Bytes for /16 LCD modules.
*
*****
LCD401 JSR WAIT
      LDAA #SA8
      JSR CLOCK
      LDX #DISP
      JSR WAIT
      BSET PORTB,Y,S20
      LDAA 8,X
      CMPA #SFF
      BNE COK2
      LDAA #S2D
      JSR CLOCK
      INX
      CPX #DISP+8
      BNE LCD41
      RTS

```

```

834
835
836
837
838
839
840 0000053a >13001006 MOD BRCLR STAT2,S10,NOCL
841 0000053e >bd0000 JSR INITD CLEAR PIN, DI, M/S, PTY & RT
842 00000541 >150010 NOCL BRCLR STAT2,S10 PERHAPS EVEN PS
843 00000544 181f08200d BRCLR PORTD,Y,S20,TRRT STANDBY ?
844 00000549 >1235046e BRSET STAT5,S04,SLPD YES, SLEEP DISPLAY ?
845 0000054d >1200087c BRSET STAT4,S08,ALRMJ NO, ALARM DISPLAY ?
846 00000551 >bd0000 JSR STBYD NO, NORMAL STANDBY DISPLAY
847 00000554 207a BRA ROW1
848 00000556 >13000258 TRRT BRCLR STAT5,S02,RTTTS RDS DISPLAY ?
849 0000055a >9600 LDAA RTDIS
850 0000055c 8101 CMPA #1
851 0000055e 2605 BNE NPTY
852 00000560 >bd0000 JSR PTYD PTY DISPLAY
853 00000563 206b BRA ROW1
854 00000565 8102 CMPA #2
855 00000567 2605 BNE NPI
856 00000569 >bd0000 JSR DIP1 PI
857 0000056c 2062 BRA ROW1
858 0000056e 8103 CMPA #3
859 00000570 2605 BNE NTAP
860 00000572 >bd0000 JSR DITAP TA & PT
861 00000575 2059 BRA ROW1
862 00000577 8104 CMPA #4
863 00000579 2605 BNE NPIN1
864 0000057b >bd0000 JSR DPIN1 PIN - HEX
865 0000057e 2050 BRA ROW1
866 00000580 8105 CMPA #5
867 00000582 2605 BNE NPIN2
868 00000584 >bd0000 JSR DPIN2 PIN - DAY AND TIME
869 00000587 2047 BRA ROW1
870 00000589 8106 CMPA #6
871 0000058b 2605 BNE NMUD
872 0000058d >bd0000 JSR LMUD MUD
873 00000590 203e BRA ROW1
874 00000592 8107 CMPA #7
875 00000594 2605 BNE NMSD
876 00000596 >bd0000 JSR DMSC M/S & DI
877 00000599 2035 BRA ROW1
878 0000059b 8108 CMPA #8
879 0000059d 2605 BNE NONPI
880 0000059f >bd0000 JSR DONPI LAST TA PI
881 000005a2 202c BRA ROW1
882 000005a4 8109 CMPA #9
883 000005a6 2605 BNE NRET
884 000005a8 >bd0000 JSR DRET TA RETURN
885 000005ab 2023 BRA ROW1
886 000005ad >bd0000 JSR DEON
887 000005b0 201e BRA ROW1
888
889
890
891
892
893
894
895 000005b2 >13000405 RTTTS BRCLR STAT2,S04,SLPD RT DISPLAY ?
896 000005b6 >bd0000 JSR RTDSP
897 000005b9 2015 BRA ROW1
898 000005bb >13000405 SLPD BRCLR STAT5,S04,NRMD SLEEP TIMER DISPLAY ?
899 000005bf >bd0000 JSR SLEEPD
900 000005c2 200c BRA ROW1
901 000005c4 >12000805 NRMD BRSET STAT4,S08,ALRMJ ALARM DISPLAY ?
902 000005c8 >bd0000 JSR NORMD
903 000005cb 2003 BRA ROW1
904 000005cd >bd0000 JSR ALRMD
905
906 000005d0 >ce0000 ROW1 LDX #DISP
907 000005d3 a600 CLOP2 LDAA 0,X
908 000005d5 a110 CMPA 16,X
909 000005d7 2607 BNE DIFF HAS CHARACTER CHANGED ?
910 000005da 08 INX
911 000005dc >8c0010 CPX #DISP+16
912 000005de 26f4 BNE CLOP2
913 000005df 39 RTS YES, NO NEED TO UPDATE DISPLAYS
914
915 000005e0 18ce1000 DIFF LDX #S1000
916 000005e4 181f0a8049 BRCLR PORTB,Y,S80,VFD LCD OR VFD ?
917 000005e9 >bd0000 JSR WAIT LCD
918 000005ec 860c LDAA #S0C SWITCH DISPLAY ON
919 000005ee >bd0000 JSR CLOCK LATCH IT
920 000005f1 >bd0000 JSR WAIT
921 000005f4 8630 LDAA #S30 /8 DISPLAY
922 000005f6 181f008002 BRCLR PORTA,Y,S80,M8A /16 DISPLAY
923 000005fb 8638 LDAA #S38 LATCH IT
924 000005fd >bd0000 JSR CLOCK
925 00000600 >bd0000 JSR WAIT
926 00000603 8680 LDAA #S80 ADDRESS DISPLAY RAM
927 00000605 >bd0000 JSR CLOCK LATCH IT
928
929 00000608 >ce0000 LDX #DISP
930 0000060b 18ce1000 LDY #C1000
931 0000060f >bd0000 JSR WAIT
932 00000612 181c0420 BSET PORTB,Y,S20 WRITE DATA
933 00000616 a600 LDAA 0,X GET A BYTE
934 00000618 a710 STRA 16,X SAVE IT IN "CURRENT" BUFFER
935 0000061a 81ff CMPA #SFF
936 0000061c 2602 BNE COK
937 0000061e 862d LDAA #S2D
938 00000620 >bd0000 JSR CLOCK SEND IT TO MODULE
939 00000623 08 INX
940 00000624 >8c0010 CPX #DISP+16
941 00000627 26e6 BNE LCD DONE ?
942 00000629 181f008003 BRCLR PORTA,Y,S80,MUX8
943 0000062e >bd0000 JSR LCD401
944 00000631 39 MUX8 RTS

```

```

946
947
948
949
950
951
952 00000632 18ce1000
953 00000636 181d0402
954 0000063a 181c0401
955 0000063e 181d0404
956
957 00000642 >ce0000
958 00000645 a600
959 00000647 >d1f0
960 00000649 8d2c
961 0000064b >8c0007
962 0000064e 26f5
963
964 00000650 >ce0000
965 00000653 >d1f0
966 00000655 e600
967 00000657 e710
968 00000659 c1ff
969 0000065b 2602
970 0000065d c52d
971 0000065f c47f
972 00000661 >ce0000
973 00000664 3a
974 00000665 a600
975 00000667 8d0e
976 00000669 >8c0010
977 0000066c 26e5
978
979 0000066e 181c0404
980 00000672 181d0401
981 00000676 39
982
983 00000677 c608
984 00000679 44
985 0000067a 2404
986 0000067c 181c0402
987 00000680 181d0401
988 00000684 181c0401
989 00000688 181d0402
990 0000068c 5a
991 0000068d 26ea
992 0000068f c654
993 00000691 5a
994 00000692 26fd
995 00000694 >de00
996 00000696 08
997 00000697 39
998
999 00000698 a00fb000800090
1000
1001
1002
1003
1004
1005
1006
1007
1008 0000069f 8620
1009 000006a1 >b70001
1010 000006a4 >b7000a
1011 000006a7 862e
1012 000006a9 >13000203
1013
1014 000006ad >b70001
1015 000006b0 >12000403
1016 000006b4 >b7000a
1017
1018 000006b7 >13004047
1019 000006bb >150040
1020 000006be 86ff
1021 000006c0 >18ce0000
1022 000006c4 18a700
1023 000006c7 1808
1024 000006c9 >188c0007
1025 000006cd 23f5
1026 000006cf >1300012f
1027 000006d3 >8601
1028 000006d5 2602
1029 000006d7 c6f0
1030 000006d9 c830
1031 000006db >f70003
1032 000006de >9602
1033 000006e0 2606
1034 000006e2 c120
1035 000006e4 2602
1036 000006e6 86f0
1037 000006e8 8b30
1038 000006ea >b70004
1039 000006ed >9603
1040 000006ef 8b30
1041 000006f1 >b70005
1042 000006f4 >9604
1043 000006f6 8b30
1044 000006f8 >b70006
1045 000006fb >9605
1046 000006fd 8b30
1047 000006ff >b70007
1048
1049 00000702 >18ce0000
1050 00000706 >ce0000
1051 00000709 18a600
1052 0000070c a702
1053 0000070e 08
1054 0000070f 1808
1055 00000711 >188c0007
1056 00000715 23f2
1057 00000717 >9600
1058 00000719 8b30
1059 0000071b >b70000
1060 0000071e >13004009

```

```

*****
*
*      VFD.
*
*****

VFD  LDY  #S1000
      BCLR PORTB,Y,$02      DATA LOW ?
      BSET PORTB,Y,$01      CLOCK HIGH ?
      BCLR PORTB,Y,$04      ENABLE LOW

DISS  LDX  #INITF           SEND VFD SET-UP BYTES
      LDAA 0,X
      STX  W7              SAVE INDEX
      BSR  VFDL
      CPX  #INITF+7
      BNE  DISS            LAST BYTE ?

VFD3  LDX  #DISP           SEND 16 CHARACTER BYTES
      STX  W7              SAVE INDEX
      LDAB 0,X
      STAB 16,X            SAVE IT IN "CURRENT" BUFFER
      CMPB #SFF
      BNE  NOTFF
      LDAB #S2D
      ANDB #S7F
      LDX  #VIDAB
      ABX
      LDAA 0,X
      BSR  VFDL
      CPX  #DISP+16
      BNE  VFD3            LAST BYTE ?

      BSET PORTB,Y,$04      ENABLE HIGH
      BCLR PORTB,Y,$01      CLOCK LOW ?
      RTS

VFDL  LDAB #8
DIS3  LSR 4
      BCC  DIS4
      BSET PORTB,Y,$02      DATA HIGH
      BCLR PORTB,Y,$01      CLOCK
      BSET PORTB,Y,$01      IT
      BCLR PORTB,Y,$02      CLEAR DATA
      DECB
      BNE  DIS3            COMPLETE ?
      LDAB #84
      BNE  DIS3            NO
      DECB
      BNE  DEL
      LDX  W7
      INX
      RTS                WAIT 200uS
                        RESTORE INDEX

INITF  FCB  $A0,$0F,$B0,$00,$80,$00,$90

*****
*
*      Normal display (PS and time or freq).
*
*****

NORMD  LDAA  #S20
      STAA DISP+1
      STAA DISP+10
      LDAA  #S2E
      BRCLR STAT4,$02,NOTP    DP TO INDICATE SLEEP TIMER RUNNING ?
      BRCLR TRB,$04,NOTP      FLASH IT
      STAA DISP+1
      BSET  STAT4,$04,TYP1    DP TO INDICATE TRAFFIC SWITCH DISABLED ?
      STAA DISP+10

TYP1  BRCLR STAT2,$40,TYP2    CLEAR PS NAME ?
      BCLR  STAT2,$40          YES, CLEAR FLAG
      LDAA  #SFF
      LDY  #PSN
      CPS  0,Y
      CPY  #PSN+7
      BLS  CPS
      BRCLR STAT,$01,TYP2    FREQUENCY MODE ?
      LDAB RQ+1
      BNE  NZ1B
      LDAB #SF0
      LDAB #S30
      STAB PSN+3
      LDAA RQ+2
      BNE  NZ2B
      CMPB #S20
      BNE  NZ2B
      LDAA #SF0
      NZ2B  ADDA #S30
      STAA PSN+4
      LDAA RQ+3
      ADDA #S30
      STAA PSN+5
      LDAA RQ+4
      ADDA #S30
      STAA PSN+6
      LDAA RQ+5
      ADDA #S30
      STAA PSN+7

TYP2  LDY  #PSN
      LDX  #DISP
      MPS  LDAA 0,Y
      SQNG  STAA 2,X          GET PS NAME
                        AND PUT INTO DISPLAY RAM

      INY
      INY
      CPY  #PSN+7
      BLS  MPS
      LDAA LED
      ADDA #S30
      STAA DISP
      BRCLR STAT5,$40,NTSCN2    GET PROGRAM NUMBER
                        STORE MODE ?

```

```

1061 00000722 >13000405 BRCLR TH8,C04,NTSON2 YES, FLASH ?
1062 00000726 8620 LDAA #S20 YES
1063 00000728 >b70000 STAA DISP
1064
1065 0000072b >a600 NTSON2 LDAB PSNP
1066 0000072d 2718 BEQ CJ EDITING ?
1067 0000072f >13000702 BRCLR TH8,S07,NCJ YES, FLASH
1068 00000733 2012 BRA CJ
1069 00000735 >ce0001 NCJ LDX #DISP+1
1070 00000738 3a ABX
1071 00000739 a600 LDAA 0,X
1072 0000073b 8120 CMPA #S20 GET CHARACTER TO FLASH
1073 0000073d 2704 BEQ SPCE SPACE ?
1074 0000073f 8620 LDAA #S20 NO, REPLACE WITH SPACE
1075 00000741 2002 BRA CJP
1076 00000743 862d SPCE LDAA #S2D YES, REPLACE WITH -
1077 00000745 a700 CJP STAA 0,X
1078
1079 00000747 >12008007 CJ BRSET STAT2,S80,TYPE3 TA SWITCH ?
1080 0000074b >13000103 BRCLR STAT,S01,TYPE3 NO, FREQUENCY MODE ?
1081 0000074f >7e0000 JMP PRGMD NO, DISPLAY TIME
1082 00000752 18ce1000 LDY #S1000
1083 00000756 181e000251 BRSET PORTA,Y,S02,AMD YES DISPLAY FREQUENCY, AM BAND ?
1084 0000075b >a601 FMD LDAB RQ+1 NO, FM
1085 0000075d 2602 BNE NZ1
1086 0000075f c6f0 LDAB #SF0
1087 00000761 cb30 NZ1 ADDB #S30
1088 00000763 >f7000a STAB DISP+10
1089 00000766 >9602 LDAA RQ+2
1090 00000768 2606 BNE NZ2
1091 0000076a c120 CMPB #S20
1092 0000076c 2602 BNE NZ2
1093 0000076e 86f0 LDAA #SF0
1094 00000770 8b30 NZ2 ADDA #S30
1095 00000772 >b7000b STAA DISP+11
1096 00000775 >9603 LDAA RQ+3
1097 00000777 8b30 ADDA #S30
1098 00000779 >b7000c STAA DISP+12
1099 0000077c 862e LDAA #S2E
1100 0000077e >12000206 BRSET STAT5,S02,SKCL RDS (BON) DISPLAY ?
1101 00000782 >13000202 BRCLR STAT5,S20,SKCL
1102 00000786 862d LDAA #S2D
1103 00000788 >b7000d SKCL STAA DISP+13
1104 0000078b >9604 LDAA RQ+4
1105 0000078d 8b30 ADDA #S30
1106 0000078f >b7000e STAA DISP+14
1107 00000792 >9605 LDAA RQ+5
1108 00000794 8b30 ADDA #S30
1109 00000796 >1200020e BRSET STAT5,S02,SKCL2 RDS (BON) DISPLAY ?
1110 0000079a >1200020a BRSET STAT5,S20,SKCL2 NO, INHIBITED ?
1111 0000079e >13000806 BRCLR STAT3,S08,SKCL2 NO, TP FLAG SET ?
1112 000007a2 >13000402 BRCLR TH8,S04,SKCL2 YES, FLASH ?
1113 000007a5 862e LDAA #S2E
1114 000007a8 >b7000f SKCL2 STAA DISP+15
1115 000007ab 39 RTS
1116
1117 *****
1118 * Normal display (cont...) *
1119 * *****
1120
1121
1122
1123 000007ac c620 AMD LDAB #S20 YES, AM
1124 000007ae >f7000a STAB DISP+10
1125 000007b1 >a601 LDAB RQ+1
1126 000007b3 2602 BNE NZ1A
1127 000007b5 c6f0 LDAB #SF0
1128 000007b7 cb30 NZ1A ADDB #S30
1129 000007b9 >f7000b STAB DISP+11
1130 000007bc >9602 LDAA RQ+2
1131 000007be 2606 BNE NZ2A
1132 000007c0 c120 CMPB #S20
1133 000007c2 2602 BNE NZ2A
1134 000007c4 86f0 LDAA #SF0
1135 000007c6 8b30 NZ2A ADDA #S30
1136 000007c8 >b7000c STAA DISP+12
1137 000007cb >9603 LDAA RQ+3
1138 000007cd 8b30 ADDA #S30
1139 000007cf >b7000d STAA DISP+13
1140 000007d2 >9604 LDAA RQ+4
1141 000007d4 8b30 ADDA #S30
1142 000007d6 >b7000e STAA DISP+14
1143 000007d9 >9605 LDAA RQ+5
1144 000007db 8b30 ADDA #S30
1145 000007dd >b7000f STAA DISP+15
1146 000007e0 39 RTS
1147
1148 000007e1 >9600 PRGMD LDAA OUR GET TIME
1149 000007e3 >b30000 JSR CBED
1150 000007e6 8130 CMPA #S30 LEADING ZERO ?
1151 000007e8 2602 BNE TNZ
1152 000007ea 8620 LDAA #S20 YES, MAKE IT A SPACE
1153 000007ec >f8000b TNZ STD DISP+11
1154 000007ef >9600 CHIN LDAA MIN
1155 000007f1 >b30000 JSR CBED
1156 000007f4 >f8000e STD DISP+14
1157 000007f7 863a CSEC LDAA #S3A
1158 000007f9 >12002006 BRSET STAT6,S20,DDC FLASHING ENABLED ?
1159 000007fd >13000402 BRCLR TH8,S04,DDC YES, TIME TO FLASH ?
1160 00000801 8630 LDAA #S20 YES, 0.5 Hz FLASHING COLON
1161 00000803 >b7000d STAA DISP+13
1162 00000806 39 RTS

```

```

1164
1165
1166
1167
1168
1169
1170
1171
1172 00000807 >9600
1173 00000809 261b
1174 0000080b >7e0000
1175 0000080e >d600
1176 00000810 >ceffff
1177 00000813 3a
1178 00000814 a600
1179 00000816 8120
1180 00000818 2617
1181 0000081a >1300200e
1182 0000081e >7c0000
1183 00000821 >7c0000
1184 00000824 >9600
1185 00000826 8145
1186 00000828 221e
1187 0000082a 20e2
1188 0000082c >140020
1189 0000082f 2003
1190 00000831 >150020
1191 00000834 >9700
1192 00000836 >ce0000
1193 00000839 a601
1194 0000083b a700
1195 0000083d 08
1196 0000083e >8c000f
1197 00000841 26f6
1198 00000843 >9600
1199 00000845 >b7000f
1200 00000848 39
1201
1202 00000849 >ce0000
1203 0000084c >18ce0000
1204 00000850 a600
1205 00000852 18a700
1206 00000855 08
1207 00000856 1808
1208 00000858 >8c000f
1209 0000085b 23f3
1210 0000085d >9600
1211 0000085f 270e
1212 00000861 >bd0000
1213 00000864 >fd000b
1214 00000867 >9601
1215 00000869 >bd0000
1216 0000086c >fd000d
1217 0000086f 39
1218
1219
1220
1221
1222
1223
1224
1225 00000870 >12001058
1226 00000874 >d600
1227 00000876 58
1228 00000877 >db0000
1229 00000879 >ce0000
1230 0000087c 3a
1231 0000087d a600
1232 0000087f >b70000
1233 00000882 a601
1234 00000884 >b70001
1235 00000887 a602
1236 00000889 >b70002
1237 0000088c 8620
1238 0000088e >b70003
1239 00000891 >b70006
1240 00000894 >b7000a
1241 00000897 >9601
1242 00000899 8b30
1243 0000089b >b70005
1244 0000089e >9600
1245 000008a0 2702
1246 000008a2 8b10
1247 000008a4 8b20
1248 000008a6 >b70004
1249 000008a9 >9601
1250 000008ab >9600
1251 000008ad 2702
1252 000008af cb0a
1253 000008b1 >d700
1254 000008b3 58
1255 000008b4 >db00
1256 000008b6 >ceffff
1257 000008b9 3a
1258 000008ba a600
1259 000008bc >b70007
1260 000008bf a601
1261 000008c1 >b70008
1262 000008c4 a602
1263 000008c6 >b70009
1264 000008c9 >7e0000

```

```

*****
*
*   RT display mode.
*
*   Last TA PI display.
*
*****

RTDSP  LDAA  DISP2
      BNE  SKP1
      JMP  PTMD
NXT     LDAB  DISP2
      LDX  #RT-1
      AEX
      LDAA  0,X
      CMPA  #S20
      BNE  NOTSP
      BRCLR STAT2,S20,FSP
      INC  DISP1
      INC  DISP2
      LDAA  DISP2
SKP1    CMPA  #69
      BHI  LCD4
      BRA  NXT
FSP     BSET  STAT2,S20
      BRA  CONT
NOTSP   BCLR  STAT2,S20
CONT    STAA  W7
      LDX  #DISP
      LDAA  1,X
      STAA  0,X
      INX
      CPX  #DISP+15
      BNE  ILP1
      LDAA  W7
      STAA  DISP+15
      LCD4  RTS
      DONTPI LDX  #CONTPIST
      LDY  #DISP
      DLOPO LDAA  0,X
      STAA  0,Y
      INX
      INY
      CPX  #CONTPIST+15
      BLS  DLOPO
      LDAA  PION
      BEQ  PINVO
      JSR  SPLIT
      STD  DISP+11
      LDAA  PION+1
      JSR  SPLIT
      STD  DISP+13
PINVO   RTS

*****
*
*   Standby display.
*
*****

STBYD  BRSET  STAT4,S10,ALARM
NOTDAY LDAB  DOW
      LSLB
      ALDB  DOW
      LDX  #DNAME
      AEX
      LDAA  0,X
      STAA  DISP
      LDAA  1,X
      STAA  DISP+1
      LDAA  2,X
      STAA  DISP+2
      LDAA  #S20
      STAA  DISP+3
      STAA  DISP+6
      STAA  DISP+10
      LDAA  DOM+1
      ADDA  #S30
      STAA  DISP+5
      LDAA  DOM
      BEQ  ADD20
      ADDA  #S10
      ADDA  #S20
      STAA  DISP+4
      LDAB  DISP
      LDAA  MYTH+1
      BEQ  MYHZ
      ALDB  #10
      STAB  W7
      LSLB
      ALDB  W7
      LDX  #MNAME-3
      AEX
      LDAA  0,X
      STAA  DISP+7
      LDAA  1,X
      STAA  DISP+8
      LDAA  2,X
      STAA  DISP+9
      JMP  PKGMD

ALARM ARMED ?
NO, GET DAY OF WEEK

DATE
IF ZERO USE A SPACE
IF NOT MAKE ASCII

MONTH, LSD
MONTH, MSD

```

```

1266
1267
1268
1269
1270
1271
1272 000008cc >13008006
1273 000008d0 >9600
1274 000008d2 8104
1275 000008d4 229e
1276 000008d6 >9600
1277 000008d8 >bd0000
1278 000008da >fd0000
1279 000008dc >9600
1280 000008de >bd0000
1281 000008e0 >fd0000
1282 000008e2 >fd0000
1283 000008e4 >bd0000
1284 000008e6 a601
1285 000008ef 18a704
1286 000008f0 08
1287 000008f2 1808
1288 000008f4 >8c0006
1289 000008f6 23f3
1290 000008f8 >7e0000
1291
1292
1293
1294
1295
1296
1297
1298 000008fd >ce0000
1299 00000900 >18ce0000
1300 00000904 a600
1301 00000906 18a700
1302 00000908 08
1303 0000090a 1808
1304 0000090c >8c000f
1305 0000090e 23f3
1306 00000910 >9600
1307 00000912 270e
1308 00000914 >bd0000
1309 00000916 >fd0000
1310 00000918 >9601
1311 0000091a >bd0000
1312 0000091c >fd000d
1313 0000091e 39
1314
1315
1316
1317
1318
1319
1320
1321 00000924 >ce0000
1322 00000926 >13001003
1323 00000928 >ce0000
1324 0000092a >18ce0000
1325 0000092c a600
1326 0000092e 18a700
1327 00000930 08
1328 00000932 1808
1329 00000934 >18ce000f
1330 00000936 23f2
1331 00000938 >13001037
1332 0000093a >13008005
1333 0000093c 8635
1334 0000093e >700000
1335 00000940 >9600
1336 00000942 >bd0000
1337 00000944 >fd000c
1338 00000946 >9600
1339 00000948 >bd0000
1340 0000094a >fd000e
1341 0000094c >1300201a
1342 0000094e >13000702
1343 00000950 2014
1344 00000952 8620
1345 00000954 >12004008
1346 00000956 >70000e
1347 00000958 >70000f
1348 0000095a 2006
1349 0000095c >70000c
1350 0000095e >70000d
1351 00000960 39
1352
1353
1354
1355
1356
1357
1358
1359 0000097c >ce0000
1360 0000097e >18ce0000
1361 00000980 a600
1362 00000982 18a700
1363 00000984 08
1364 00000986 1808
1365 00000988 >8c000f
1366 0000098a 23f3
1367 0000098c 8631
1368 0000098e >13008003
1369 00000990 >700006
1370 00000992 >13000403
1371 00000994 >70000e
1372 00000996 39

```

```

*****
*
*      Standby (alarm armed) & PI displays.
*
*****

```

```

ALRMA  BRCLR  STAT5,$80,D7      ARMED, BUT IS IT 5-DAY ?
        LDAA   LOW              YES
        CMPA   #4              SAT OR SUN ?
        BHI    NOTODAY         IF SO, THEN NORMAL STANDBY DISPLAY
D7      LDAA   ACUR              GET ALARM HOURS
        JSR    CBCT
        STD    DISP
        LDAA   AMIN
        JSR    CBCT
        STD    DISP+2
        LDX    #ALARMF
        LDY    #DISP
ALOP2   LDAA   1,X
        STAA   4,Y
        INX
        INY
        CPX    #ALARMF+6
        BLS    ALOP2
        JMP    PRGMD

```

```

*****
*
*      PI display.
*
*****

```

```

DIPI    LDX    #PIST
        LDY    #DISP
DLOP    LDAA   0,X
        STAA   0,Y
        INX
        INY
        CPX    #PIST+15
        BLS    DLOP
        LDAA   PI
        BEQ    PINV
        JSR    SPLIT
        STD    DISP+11
        LDAA   PI+1
        JSR    SPLIT
        STD    DISP+13
PINV    RTS

```

```

*****
*
*      Alarm display.
*
*****

```

```

ALRMD    LDX    #ALARMF
        BRCLR  STAT4,$10,ALOFD  ARMED ?
        LDX    #ALARMN         YES
        LDY    #DISP          NO
ALOFD    LDAA   0,X
ALOP      STAA   0,Y
        INX
        INY
        CPY    #DISP+15
        BLS    ALOP
        BRCLR  STAT4,$10,ALOF  ALARM ARMED ?
        BRCLR  STAT5,$80,NSD   YES, WEEKDAY ONLY ?
        LDAA   #535            TES, REPLACE 7 WITH 5
        STAA   DISP
        LDAA   ACUR              GET ALARM HOURS
        JSR    CBCT
        STD    DISP+12
        LDAA   AMIN
        JSR    CBCT
        STD    DISP+14
        BRCLR  STAT4,$20,ALOF  SET-UP ?
        BRCLR  TH6,$07,NALOF
        BRA    ALOF
NALOF    LDAA   #520
        BRSET  STAT4,$40,FH    HOURS ?
        STAA   DISP+14         NO, FLASH MINUTES
        STAA   DISP+15
        BRA    ALOF
        LDAA   DISP+12
        STAA   DISP+13
        STAA   DISP+13
ALOF     RTS

```

```

*****
*
*      TA/TP display.
*
*****

```

```

DITAP    LDX    #TAPST
        LDY    #DISP
BLOP     LDAA   0,X
        STAA   0,Y
        INX
        INY
        CPX    #TAPST+15
        BLS    BLOP
        LDAA   #531
        BRCLR  STAT3,$08,TPLOW
        STAA   DISP+6
        BRCLR  STAT3,$04,TALOW
        STAA   DISP+14
        STAA   DISP+14
TALOW    RTS

```

```

1374
1375
1376
1377
1378
1379
1380 000009a1 >ce0000
1381 000009a4 >18ce0000
1382 000009a8 a600
1383 000009aa 18a700
1384 000009ad 08
1385 000009ae 1808
1386 000009b0 >188c000f
1387 000009b4 23f2
1388 000009b6 >9600
1389 000009b8 270e
1390 000009ba >bd0000
1391 000009bd >fd000b
1392 000009c0 >9601
1393 000009c2 >bd0000
1394 000009c5 >fd000d
1395 000009c8 39
1396
1397 000009c9 >ce0000
1398 000009cc >18ce0000
1399 000009d0 a600
1400 000009d2 18a700
1401 000009d5 08
1402 000009d6 1808
1403 000009d8 >188c000f
1404 000009dc 23f2
1405 000009de >9600
1406 000009e0 270e
1407 000009e2 44
1408 000009e3 44
1409 000009e4 44
1410 000009e5 >bd0000
1411 000009e8 8130
1412 000009ea 2602
1413 000009ec 8620
1414 000009ee >fd0002
1415 000009f1 8131
1416 000009f3 272a
1417 000009f5 c131
1418 000009f7 260a
1419 000009f9 8673
1420 000009fb >b70004
1421 000009fe 8674
1422 00000a00 >b70005
1423 00000a03 c132
1424 00000a05 260a
1425 00000a07 866e
1426 00000a09 >b70004
1427 00000a0c 8664
1428 00000a0e >b70005
1429 00000a11 c133
1430 00000a13 260a
1431 00000a15 8672
1432 00000a17 >b70004
1433 00000a1a 8664
1434 00000a1c >b70005
1435 00000a1f >9600
1436 00000a21 8407
1437 00000a23 >d601
1438 00000a25 58
1439 00000a26 49
1440 00000a27 58
1441 00000a28 49
1442 00000a29 >bd0000
1443 00000a2c >fd000a
1444 00000a2f >9601
1445 00000a31 843f
1446 00000a33 >bd0000
1447 00000a36 >fd000d
1448 00000a39 39
1449
1450
1451
1452
1453
1454
1455
1456 00000a3a 8d26
1457 00000a3c >9600
1458 00000a3e 2721
1459 00000a40 8b30
1460 00000a42 >b7000a
1461 00000a45 >9601
1462 00000a47 8b30
1463 00000a49 >b7000b
1464 00000a4c >9602
1465 00000a4e 8b30
1466 00000a50 >b7000c
1467 00000a53 >9603
1468 00000a55 8b30
1469 00000a57 >b7000d
1470 00000a5a >9604
1471 00000a5c 8b30
1472 00000a5e >b7000e
1473 00000a61 39
1474
1475 00000a62 >ce0000
1476 00000a65 >18ce0000
1477 00000a69 a600
1478 00000a6b 18a700
1479 00000a6e 08
1480 00000a6f 1808
1481 00000a71 >188c000f
1482 00000a75 23f2
1483 00000a77 39

```

```

.....
*
*   PIN displays.
*
*
.....

```

```

DPIN1  LDX    #PINST1
        LDY    #DISP
PLOP   LDAA   0,X
        STAA   0,Y
        INX
        INY
        CPY    #DISP+15
        BLS    PLOP
        LDAA   PIN
        BEQ    PINNV
        JSR    SPLIT
        STD    DISP+11
        LDAA   PIN+1
        JSR    SPLIT
        STD    DISP+13
PINNV   RTS

DPIN2  LDX    #PINST2
        LDY    #DISP
PLOP2  LDAA   0,X
        STAA   0,Y
        INX
        INY
        CPY    #DISP+15
        BLS    PLOP2
        LDAA   PIN
        BEQ    PINNV
        JSR    LSRA
        JSR    LSRA
        JSR    CBDD
        CMPA   #S3C
        BNE    DTNO
        LDAA   #S20
        STD    DISP+2
        CMPA   #S31
        BEQ    NOTRD
        CMPB   #S31
        BNE    NOTST
        LDAA   #'s
        STAA   DISP+4
        LDAA   #'c
        STAA   DISP+5
        NOTST  CMPB   #S32
        BNE    NOTIND
        LDAA   #'n
        STAA   DISP+4
        LDAA   #'d
        STAA   DISP+5
        NOTIND CMPB   #S33
        BNE    NOTRD
        LDAA   #'r
        STAA   DISP+4
        LDAA   #'d
        STAA   DISP+5
        NOTRD  LDAA   PIN
        ANDB   #7
        LDAB   PIN+1
        ASLB
        ROLA
        JSR    CBDD
        STD    DISP+10
        LDAA   PIN+1
        ANDB   #S3F
        JSR    CBDD
        STD    DISP+13
        RTS

```

DATE

HOURS

MINUTES

```

.....
*
*   MJD display.
*
*
.....

```

```

DMJD   BSR    SMJD
        LDAA   MJD
        BEQ    MJTNV
        ALDA   #S30
        STAA   DISP+10
        LDAA   MJD+1
        ALDA   #S30
        STAA   DISP+11
        LDAA   MJD+2
        ALDA   #S30
        STAA   DISP+12
        LDAA   MJD+3
        ALDA   #S30
        STAA   DISP+13
        LDAA   MJD+4
        ALDA   #S30
        STAA   DISP+14
MJTNV   RTS

SMJD   LDX    #MJDST
        LDY    #DISP
MLOP   LDAA   0,X
        STAA   0,Y
        INX
        INY
        CPY    #DISP+15
        BLS    MLOP
        RTS

```

```

1485
1486
1487
1488
1489
1490
1491 00000a78 >9600
1492 00000a7a c610
1493 00000a7c 3d
1494 00000a7d >ce0000
1495 00000a80 3a
1496 00000a81 >18ce0000
1497 00000a85 a600
1498 00000a87 18a700
1499 00000a8a 08
1500 00000a8b 1808
1501 00000a8d >188c000f
1502 00000a91 23f2
1503 00000a93 39
1504
1505
1506
1507
1508
1509
1510
1511 00000a94 >ce0000
1512 00000a97 >18ce0000
1513 00000a9b a600
1514 00000a9d 18a700
1515 00000aa0 08
1516 00000aa1 1808
1517 00000aa3 >188c000f
1518 00000aa7 23f2
1519 00000aa9 >9600
1520 00000aab >bd0000
1521 00000aae >fd0008
1522 00000ab1 39
1523
1524 00000ab2 >ce0000
1525 00000ab5 >18ce0000
1526 00000ab9 a600
1527 00000abb 18a700
1528 00000abe 08
1529 00000abf 1808
1530 00000ac1 >188c000f
1531 00000ac5 23f2
1532 00000ac7 >13000805
1533 00000acb 864d
1534 00000acd >b70006
1535 00000ad0 >9600
1536 00000ad2 >bd0000
1537 00000ad5 >fd000d
1538 00000ad8 39
1539
1540
1541
1542
1543
1544
1545
1546 00000ad9 >bd0000
1547
1548 00000adc >9600
1549 00000ade 800a
1550 00000ae0 c610
1551 00000ae2 3d
1552 00000ae3 >ce0000
1553 00000ae6 3a
1554
1555 00000ae7 8620
1556 00000ae9 >b70008
1557 00000aec >b70009
1558 00000aef a602
1559 00000af1 >b70000
1560 00000af4 a603
1561 00000af6 >b70001
1562 00000af9 a604
1563 00000afb >b70002
1564 00000afe a605
1565 00000b00 >b70003
1566 00000b03 a606
1567 00000b05 >b70004
1568 00000b08 a607
1569 00000b0a >b70005
1570 00000b0d a608
1571 00000b0f >b70006
1572 00000b12 a609
1573 00000b14 >b70007
1574
1575 00000b17 a60d
1576
1577 00000b19 81cd
1578 00000b1b 2603
1579 00000b1d 08
1580 00000b1e a60d
1581 00000b20 81fa
1582 00000b22 2718
1583
1584 00000b24 81cc
1585 00000b26 22b0
1586 00000b28 c60a
1587 00000b2a 3d
1588 00000b2b c62e
1589 00000b2d >d700
1590 00000b2f 892d
1591 00000b31 >9700
1592 00000b33 >bd0000
1593 00000b36 >bd0000
1594 00000b39 >7e0000

```

```

*****
*
*      TA return display.
*
*****

DRET  LDAA  REARET
      LDAB  #16
      MUL
      LDX  #DARET
      ABX
      LDY  #DISP
RLOP  LDAA  0,X
      STAA  0,Y
      INX
      INY
      CPY  #DISP+15
      BLS  RLOP
      RTS

*****
*
*      Sleep and M/S & DI displays.
*
*****

SLEEPD LDX  #SLPGT
      LDY  #DISP
SLOP  LDAA  0,X
      STAA  0,Y
      INX
      INY
      CPY  #DISP+15
      BLS  SLOP
      LDAA  SLEPT
      JSR  CBDD
      STD  DISP+8
      RTS

DMSD  LDX  #MSDST
      LDY  #DISP
ILOP  LDAA  0,X
      STAA  0,Y
      INX
      INY
      CPY  #DISP+15
      BLS  ILOP
      BRCLR .,STAT5,$08,MSM
      LDAA  #'M
      STAA  DISP+6
      LDAA  DI
      JSR  CBDD
      STD  DISP+13
      FNOK  RTS

*****
*
*      BDN display.
*
*****

DEON  JSR  SMJD          CLEAR FREQUENCY CHARACTERS
      LDAA  RTDIS
      SUBA  #10
      LDAB  #16
      MUL
      LDX  #BDN
      ABX
      LDAA  #S20
      STAA  DISP+8
      STAA  DISP+9
      LDAA  2,X
      STAA  DISP
      LDAA  3,X
      STAA  DISP+1
      LDAA  4,X
      STAA  DISP+2
      LDAA  5,X
      STAA  DISP+3
      LDAA  6,X
      STAA  DISP+4
      LDAA  7,X
      STAA  DISP+5
      LDAA  8,X
      STAA  DISP+6
      LDAA  9,X
      STAA  DISP+7
      LDAA  13,X
      CMPA  #205          FILLER ?
      BNE  NFIL
      INX
      LDAA  13,X          YES, TRY AGAIN
      CMPA  #250          MEDIUM/LONG ?
      BEQ  MLWF
      CMPA  #204          FREQUENCY OK ?
      BHI  FNOK
      LDAB  #10
      MUL
      ADDB  #S2E
      STAB  W1
      ADCA  #S22
      STAA  W2
      JSR  LOON2
      JSR  TYPE3
      JMP  NEW          RESTORE Q

```

```

1596
1597
1598
1599
1600
1601
1602 00000b3c 08
1603 00000b3d a60d
1604 00000b3f 810f
1605 00000b41 2302
1606 00000b43 8b1b
1607 00000b45 8b10
1608 00000b47 c609
1609 00000b49 3d
1610 00000b4a >d700
1611 00000b4c >9700
1612 00000b4e >bd0000
1613 00000b51 >9602
1614 00000b53 2602
1615 00000b55 86f0
1616 00000b57 8b30
1617 00000b59 >b70009
1618 00000b5c >9603
1619 00000b5e 8b30
1620 00000b60 >b7000a
1621 00000b63 >9604
1622 00000b65 8b30
1623 00000b67 >b7000b
1624 00000b6a >9605
1625 00000b6c 8b30
1626 00000b6e >b7000c
1627 00000b71 866b
1628 00000b73 >b7000d
1629 00000b75 8648
1630 00000b78 >b7000e
1631 00000b7b 867a
1632 00000b7d >b7000f
1633 00000b80 >7e0000
1634
1635
1636
1637
1638
1639
1640
1641 00000b83 >ce0000
1642 00000b86 86ff
1643 00000b88 a700
1644 00000b8a 08
1645 00000b8b >8c0008
1646 00000b8e 26f8
1647 00000b90 39
1648
1649
1650
1651
1652
1653
1654
1655 00000b91 16
1656 00000b92 0d
1657 00000b93 46
1658 00000b94 0d
1659 00000b95 46
1660 00000b96 44
1661 00000b97 44
1662 00000b98 8139
1663 00000b9a 2302
1664 00000b9c 8b07
1665 00000b9e c40f
1666 00000ba0 cb30
1667 00000ba2 c139
1668 00000ba4 2302
1669 00000ba6 cb07
1670 00000ba8 39
1671
1672
1673
1674
1675
1676
1677
1678
1679
1680 00000ba9 18c01000
1681 00000bad 18a703
1682 00000bb0 181c0480
1683 00000bb4 181d0480
1684 00000bb8 39
1685
1686 00000bb9 18c01000
1687 00000bbd 181d04a0
1688 00000bc1 181c0440
1689 00000bc5 186f07
1690 00000bc8 181c0480
1691 00000bcc 01
1692 00000bcd 18a603
1693 00000bd0 181d0480
1694 00000bd4 >9700
1695
1696 00000bd6 >120080ee
1697 00000bda 186307
1698 00000bdb 181d0440
1699 00000be1 39

```

```

*****
*
*   Medium/long-wave EON display.
*
*****

MILWF INX          DISPLAY M/L EON FREQUENCY
      LDA 13,X
      CMA #15
      BLS LONG
      ADDA #27      MW OFFSET
LONG  ADDA #16      M/L OFFSET
      LDAB #9
      MUL
      STAB W1
      STAB W2
      JSR CCN2
      LDA RQ-2
      BNE NZ3
      LDA #SF0
      ADDA #S30
NZ3   STAB DISP+9
      LDA RQ-3
      ADDA #S30
      STAB DISP+10
      LDA RQ-4
      ADDA #S30
      STAB DISP+11
      LDA RQ-5
      ADDA #S30
      STAB DISP+12
      LDA #'K
      STAB DISP+13
      LDA #'H
      STAB DISP+14
      LDA #'Z
      STAB DISP+15
      JMP NEW      RESTORE Q

*****
*
*   Clear PS-name after confidence loss.
*
*****

CLRPS LDY #PSN          CLEAR PS-NAME
      LDA #SFF          TO '-'s
PLOP3 STAB 0,X
      INX
      CPX #PSN+8
      BNE PLOP3
      RTS

*****
*
*   Split A nibbles into B (LS) and A (MS).
*
*****

SPLIT TAB              MSD INTO A, LSD INTO B
      SEC
      RORA
      SEC
      RORA
      LSRA
      CMA #S39
      CMA XCK
      BLS #7
      ANDB #S0F
XCK   ANDB #S3C
      ANDB #S39
      BLS ACK
      ANDB #7
ACK   RTS

*****
*
*   Send and clock data to LCD module.
*
*   Check to see if LCD module is busy.
*
*****

CLOCK LDY #S1000
      STAB PORTC,Y
      BSET PORTB,Y,$80
      BCLR PORTB,Y,$80      CLOCK IT
      RTS

WAIT  LDY #S1000
      BCLR PORTB,Y,$A0      READ LCD MODULE BUSY FLAG
      BSET PORTB,Y,$40
      CLR PORTC,Y
WLOOP BSET PORTB,Y,$80      INPUT ON PORTC
      NOP                  CLOCK HIGH
      LDA PORTC,Y
      BCLR PORTB,Y,$80      READ MODULE
      STAB W7              CLOCK LOW
      BRSET PORTD,Y,$02,NOTEST **TEST**
      BRSET W7,$80,WLOOP     BUSY ?
NOTEST CMA PORTC,Y
      BCLR PORTB,Y,$40      OUTPUT ON PORTC
      RTS

```

```

1701
1702
1703
1704
1705
1706
1707 00000be2 16
1708 00000be3 840f
1709 00000be5 c4f0
1710 00000be7 8b00
1711 00000be9 19
1712 00000bea c010
1713 00000bec 2505
1714 00000bee 8b16
1715 00000bf0 19
1716 00000bf1 20f7
1717 00000bf3 >7e0000
1718
1719
1720
1721
1722
1723
1724
1725 00000bf6 6e6f2070726f6772
1726 00000c06 2020202020204e65
1727 00000c16 43757272656e7420
1728 00000c26 2020496e666f726d
1729 00000c36 20202020202053706f
1730 00000c46 2020204564756361
1731 00000c56 2020202020447261
1732 00000c66 2020202043756c74
1733 00000c76 2020202053636965
1734 00000c86 2020202020566172
1735 00000c96 202020506f70206d
1736 00000ca6 202020526f636b20
1737 00000cb6 2045617379206c69
1738 00000cc6 204c696768740263
1739 00000cd6 536572696f757320
1740 00000ce6 20204f7468657220
1741
1742
1743
1744
1745
1746
1747
1748 00000cf6 >ce0000
1749 00000cf9 86ff
1750 00000cfb a700
1751 00000cfd 08
1752 00000cfe >4c0100
1753 00000d01 26f8
1754 00000d03 39
1755
1756
1757
1758
1759
1760
1761
1762 00000d04 >7f0000
1763 00000d07 >7f0001
1764 00000d0a >7f000c
1765 00000d0d >150008
1766 00000d10 >7f0000
1767 00000d13 >150008
1768 00000d16 >13000103
1769 00000d1a >4d000c
1770
1771 00000d1d 86a0
1772 00000d1f >b70000
1773 00000d22 >b70001
1774 00000d25 >b70003
1775 00000d28 >b70004
1776 00000d2b 862d
1777 00000d2d >b70002
1778 00000d30 8620
1779 00000d32 >ce0000
1780 00000d35 a705
1781 00000d37 08
1782 00000d38 >4c0040
1783 00000d3b 26f8
1784 00000d3d >7f0000
1785 00000d40 >7f0000
1786 00000d43 >150004
1787 00000d46 39
1788
1789 00000d47 2020416c61726d20
1790 00000d57 372044617920416c
1791 00000d67 20504920636f6465
1792 00000d77 6c61737420544120
1793 00000d87 205450202d203020
1794 00000d97 2050494e206e6f2e
1795 00000da7 2020202074682061
1796 00000db7 204d4a2064617920
1797 00000dc7 20536c6565702020
1798 00000dd7 204d4f5320205320
1799 00000de7 20313020536e6f6f
1800
1801 00000df7 5441207274726e3a
1802 00000e07 544120696e68623a
1803 00000e17 5441207274726e3a
1804 00000e27 5441207274726e3a
1805 00000e37 5441207274726e3a
1806 00000e47 5441207274726e3a
1807 00000e57 544120696e68623a
1808 00000e67 544120696e68623a
1809 00000e77 544120696e68623a
1810 00000e87 5441207274726e3a

```

```

*****
*
*   Hex->BCD conversion (& decimal adjust).
*
*****

```

CBCTD	TAB		HEX IN A & B
	ANDA	#50F	LSB IN A
	ANDB	#5F0	MSB (x16) IN B
	ALDA	#0	
	DAA		
MOREB	SUBB	#510	DECREMENT MSB
	BCC	BDONE	TOO FAR ?
	ALDA	#516	NO, ADD 16 TO A,
	DAA		ADJUST.
	BRA	MOREB	AND TRY AGAIN
BDONE	JMP	SPLIT	

```

*****
*
*   Programme Type (PTY) Codes.
*
*****

```

PTYT	FCC	'no program type'	0
	FCC	'News'	1
	FCC	'Current affairs'	2
	FCC	'Information'	3
	FCC	'Sport'	4
	FCC	'Education'	5
	FCC	'Drama'	6
	FCC	'Culture'	7
	FCC	'Science'	8
	FCC	'Varied'	9
	FCC	'Pop music'	10
	FCC	'Rock music'	11
	FCC	'Easy listening'	12
	FCC	'Light classics'	13
	FCC	'Serious classics'	14
	FCC	'Other music'	15

```

*****
*
*   Clear EON data.
*
*****

```

CLREON	LIX	#EON	
	LDRA	#5FF	
ELOP	STRA	0,X	EON RAM CLEAR
	INX		
	CPX	#EON+256	
	BNE	ELOP	
	RTS		

```

*****
*
*   LCD initialisation.
*
*****

```

INITD	CLR	PIN	CLEAR
	CLR	PIN+1	PIN.
	CLR	DI	DI.
	BCLR	STAT5,\$08	M/S BIT.
	CLR	PTY	PTY.
	BCLR	STAT3,\$08	AND TP FLAG
	BRCLR	STAT3,\$01,INITR	OFF STATION ?
	JSR	CLRPS	YES, CLEAR PS-NAME
INITR	LDAA	#5AC	CLEAR RT
	STAA	RT	INITIALISE SPACES BEFORE RT
	STAA	RT+1	
	STAA	RT+3	
	STAA	RT+4	
	LDAA	#52D	
	STAA	RT+2	AND -
	LDAA	#520	INITIALISE RADIOTEXT TO SPACES
	LDX	#RT	AFTER CONFIDENCE LOSS OR TEXT A/B CHANGE
CLOP	STAA	5,X	
	LDX		
	CPX	#RT+64	
	BNE	CLOP	
	CLR	DISP1	INITIALISE SCROLLING POINTERS
	CLR	DISP2	
	BCLR	STAT2,\$04	CANCEL RT DISPLAY
	RTS		
ALARMF	FCC	'Alarm - OFF'	
ALARMN	FCC	'7 Day Alarm ----'	
PIST	FCC	'PI code -'	
ONPIST	FCC	'last TA PI'	
TAPST	FCC	'TP - 0 TA - 0'	
PINST1	FCC	'PIN no. -'	
PINST2	FCC	'th at ----'	
MJDST	FCC	'Mj day -'	
SLPST	FCC	'Sleep 0 min.'	
MSDST	FCC	'M/S S DI 0'	
SNOZ	FCC	'10 Snooze ----'	NOT USED
TARET	FCC	'TA rtn:	
	FCC	'TA inh:	flag 1
	FCC	'TA rtn:	signal 2 NOT USED
	FCC	'TA rtn:	PI code 3
	FCC	'TA rtn:	TA low 4
	FCC	'TA rtn:	TP low 5
	FCC	'TA inh:	ill frq 6 NOT USED
	FCC	'TA inh:	EON PI 7
	FCC	'TA inh:	NVM 8
	FCC	'TA rtn:	manual 9

```

1812
1813
1814
1815
1816
1817
1818
1819
1820
1821
1822
1823
1824
1825
1826 00000ee9 7e7e7e7e
1827 00000eeb 7e7e7e7e
1828 00000ee1 7e7e7e7e
1829 00000ea3 7e7e7e7e
1830
1831 00000ea7 7e7e7e7e
1832 00000ea9 7e7e7e7e
1833 00000eaf 7e7e7e7e
1834 00000eb3 7e7e7e7e
1835
1836 00000eb7 7e7b7a7e
1837 00000ebb 7e7e7e7a
1838 00000ebf 7e7e7e7e
1839 00000ec3 3f7d3e7d
1840
1841 00000ec7 00010203
1842 00000ecb 04050607
1843 00000ecf 08097d7e
1844 00000ed3 7e7e7e7c
1845
1846 00000ed7 7e0a0b0c
1847 00000edb 0d0e0f10
1848 00000edf 11213134
1849 00000ee3 15161718
1850
1851 00000ee7 191a1b1c
1852 00000eeb 1d1ef220
1853 00000ee1 2122237e
1854 00000ee3 7e7e7e7d
1855
1856 00000ef7 7a242526
1857 00000efb 2728292a
1858 00000eff 2b2c2d2e
1859 00000f03 2f203132
1860
1861 00000f07 33343536
1862 00000f0b 3738393a
1863 00000f0f 3b3c3d3e
1864 00000f13 7e7e7e7e
1865
1866

```

Section synopsis

```

1 000000ae (      174) .RAM1
2 00000100 (      256) .RAM2
3 0000006d (      109) .RAM3
4 00000117 (    3863) .ROM3

```

Symbol table

.RAM1	1	00000000	DIS3	4	00000679	LT60	4	00000392	NFIN1	4	00000580	RDSTO	1	00000049		
.RAM2	2	00000000	DIS4	4	00000680	M8A	4	0000051d	NFIN2	4	00000589	REARET	1	000000a2		
.RAM3	3	00000000	DIS5	4	00000645	MIN	1	00000070	NPS	4	00000423	RLOP	4	00000a85		
.ROM3	4	00000000	DISP	3	00000000	MJD	1	00000030	NPTY	4	00000565	ROW1	4	000005d0		
ADD	I	0	00000000	DISP1	1	00000074	MJDAT	E	4	000004cf	NRET	4	000005ad	RP	1	0000007c
ADD20	4	000008a4	DISP2	1	00000075	MJDC	I	0	00000000	NRMD	4	000005c4	RQ	1	00000076	
ADDON	4	000003e1	DISPP	3	00000010	MJDNV	4	00000a61	NTAP	4	00000577	RR	1	00000082		
ALARMF	4	00000d47	DIST	1	00000047	MJDST	4	00000db7	NTD	4	000002ae	RT	3	00000028		
ALARHN	4	00000d57	DITAP	4	0000097c	MLOP	4	00000a69	NTSCN2	4	0000072b	RTDIS	1	000000a3		
ALOF	4	0000097b	DLOP	4	00000904	MLWF	4	00000b3c	NUM1	1	0000009b	RTDSP	4	00000807		
ALOFD	4	0000092e	DLOPO	4	00000850	MNAME	I	0	00000000	NUM2	1	0000009d	RTITS	4	000005b2	
ALOP	4	00000932	DMJD	4	00000a3a	MNTH	1	00000042	NWR	4	00000025	S02	4	00000130		
ALOP2	4	000008ed	DMSD	4	00000ab2	MOD	E	4	0000053a	NWR2	4	0000002c	S03	4	000000f2	
ALRMA	4	000008cc	DNAME	I	0	00000000	MOREB	4	00000bea	NXT	4	0000080e	S12	4	00000138	
ALRMD	4	00000924	DNDX	4	00000245	MPS	4	00000709	NXTJ	4	00000503	S13	4	000000fa		
ALRMJ	4	000005cd	DOM	1	00000044	MSDST	4	00000dd7	NZ1	4	00000761	S22	4	00000140		
AMD	4	000007ac	DONPI	4	00000849	MSM	4	00000ad0	NZ1A	4	000007b7	S23	4	00000102		
AMIN	1	00000072	DOW	1	00000046	MSZ	4	000002f3	NZ1B	4	000006d9	S32	4	00000148		
AOK	4	00000ba8	DPIN1	4	000009a1	MTHZ	4	000008b1	NZ2	4	00000770	S33	4	0000010a		
AOUR	1	00000073	DPIN2	4	000009c9	MUX8	4	00000631	NZ2A	4	000007c6	S42	4	00000150		
BCTO	1	000000ac	DRET	4	00000a78	N14B	4	00000053	NZ2B	4	000006e8	S43	4	00000110		
BDONE	4	00000bf3	DTN0	4	000009ee	N5D	4	0000094d	NZ3	4	00000b57	S52	4	00000158		
BIT	1	00000068	ELOP	4	00000cfb	NALOF	4	00000967	ONPIST	4	00000d77	S53	4	00000118		
BLOP	4	00000983	EON	2	00000000	NCH	4	00000324	OUR	1	00000071	S62	4	0000015e		
BMJD	1	00000000	FH	4	00000975	NCJ	4	00000735	OUT1	4	000003e3	S63	4	00000120		
BOK	4	000004b5	FIN	4	0000016e	NEG	4	00000381	OUT2	4	000004aa	S72	4	00000166		
CARRY	1	00000099	FMD	4	0000075b	NEW	I	0	00000000	OUT3	4	000004a7	S73	4	00000128	
CBGD	E	4	00000be2	FNOK	4	00000ad8	NFIL	4	00000b20	P	1	00000015	SCHAN	1	000000a5	
CCBH	4	000000b3	FOK2	4	00000b28	NMJD	4	00000592	PI	1	00000061	SCNG	4	0000070e		
CJ	4	00000747	FOUR	4	00000012	NMLW	4	00000443	PIN	1	00000065	SCNT	1	0000004d		
CJP	4	00000745	FSP	4	0000082c	NMR	4	00000203	PINNV	4	000009c8	SDATA	E	4	000000b7	
CLKC	4	00000037	GROUP	1	00000057	NMSD	4	0000059b	PINST1	4	00000497	SEC	1	0000006f		
CLOCK	E	4	00000ba9	GRP0	4	0000027f	NNOW	4	000001d7	PINST2	4	00000da7	SHAFT	I	0	00000000
CLOP	4	00000d35	GRP1	4	000002fe	NOCL	4	00000544	PINV	4	00000923	SKCL	4	00000788		
CLOP2	4	000005d3	GRP14A	4	000003ee	NONPI	4	000005a4	PINVO	4	0000086f	SKCL2	4	000007a8		
CLRAS	I	0	00000000	GRP14B	4	00000484	NORMD	4	0000069f	PION	1	00000063	SKP1	4	00000826	
CLREON	E	4	00000cf6	GRP2	4	0000030d	NOT0	4	000002c2	PIST	4	00000d67	SKPDC	4	000001d8	
CLRPS	4	00000b83	GRP4	4	00000348	NOT1	4	000002cf	PLOP	4	000009a8	SLEEPD	4	00000a94		
CLTR	I	0	00000000	HDON	4	000003ca	NOT2	4	000002dc	PLOP2	4	000009d0	SLEPT	1	00000048	
CMIN	4	000007ef	ILOP	4	00000ab9	NOT3	4	000002e9	PLOP3	4	00000b88	SLOP	4	00000a9b		
COK	4	00000620	ILP1	4	00000839	NOT4	4	000001db	POS	4	000003b7	SLPD	4	000005bb		
COK2	4	00000530	INITD	E	4	00000d04	NOT5	4	0000007b	PR14B	4	00000480	SLPST	4	00000dc7	
CONF	1	0000006c	INITF	4	00000698	NOTC	4	000000aa	PRGMD	4	000007e1	SMEM	1	000000a0		
CONT	4	00000834	INITR	4	00000d1d	NOTD	4	000000a7	PROC	E	4	00000239	SMJD	4	00000a62	
COUNT	1	0000009a	ITMP1	1	00000069	NOTEST	4	00000bda	PROC1	4	00000216	SNOZ	4	00000de7		
CPS	4	000006c4	KEY	1	00000096	NOTFF	4	0000065f	PROC14	4	000003e7	SPCE	4	00000743		
CSEC	4	000007f7	KOUNT	1	00000097	NOTFM	4	000000e7	PROC2	4	00000309	SPLIT	4	00000b91		
D7	4	000008d6	LCD	4	0000060f	NOTH	4	00000467	PROC4	4	00000341	STAT	1	000000a6		
DAT	1	0000004b	LCD3	4	000004c0	NOTH1	4	00000475	PSN	3	00000020	STAT2	1	000000a7		
DCON2	I	0	00000000	LCD4	4	00000848	NOTHN	4	00000394	PSNP	1	0000004a	STAT3	1	000000a8	
DDC	4	00000803	LCD401	4	00000516	NOTHP	4	000003cc	PTY	1	0000005f	STAT4	1	000000a9		
DECC	4	000001d4	LCD41	4	00000521	NOTND	4	00000a11	PTYCMP	1	00000060	STAT5	1	000000aa		
DEL	4	00000691	LCDINI	4	00000321	NOTODAY	4	00000874	PTYD	4	000004ae	STAT6	1	000000ab		
DEON	4	00000ad9	LED	1	0000009f	NOTP	4	000006b0	PTYL	4	00000253	STBYD	4	00000870		
DHIGH	4	000000d2	LEV	1	00000067	NOTRD	4	00000a1f	PTYT	4	00000bf6	SYN	1	0000006a		
DI	1	0000000a	LOCAL	4	0000037a	NOTSP	4	00000831	Q	1	00000003	TACK	4	00000290		
DIFF	4	000005e0	LONG	4	00000b45	NOTST	4	00000a03	R	1	00000027	TAH	4	000002ab		
DIG2	1	00000098	LOOPJ	4	000004f0	NOTV	4	000001bb	RDSO	4	000000c7	TALOW	4	000009a0		
DIP1	4	000008fd	LPIL	4	000003f8	NPI	4	0000056e	RDSOK	4	00000066	TAOH	4	00000499		
TAPST	4	00000d87	THQ	1	0000000c	TRYC	4	0000019f	TYPE3	4	00000752	W3	1	0000008c		
TARET	4	00000df7	TNZ	4	000007ec	TRYCD	4	000001af	VALID	4	000001ee	W4	1	0000008e		
TEXTA	4	00000311	TOOL9	4	00000451	TRYD	4	000001dc	VC	4	000001b9	W5	1	00000090		
TEXTB	4	0000031a	TPL	4	00000263	TRYPIN	4	00000453	VFD	4	00000632	W6	1	00000092		
TFCC	E	4	000000ab	TPL1	4	00000260	TRYRT	4	00000556	VFD3	4	00000653	W7	1	00000094	
TGRP15	4	00000279	TPLO	4	000003f5	TT1	4	000003ab	VFDL	4	00000677	WAIT	E	4	00000bb9	
TH32	1	0000006d	TPLOW	4	00000999	TT2	4	000003ae	VLD	4	000001f9	WLOOP	4	00000bc8		
TH8	1	0000006e	TRY1	4	000000e8	TXLP	4	00000232	VTAB	4	00000e97	XFER	4	0000022f		
TINTB	E	4	00000000	TRY2	4	000000ec	TYP1	4	000006b7	W1	1	00000088	XOK	4	00000b9e	
TMP	1	0000001e	TRYA	4	00000183	TYP2	4	00000702	W2	1	0000008a	YR	1	00000039		
TMPGRP	1	0000004f	TRYB	4	00000191	TYPE3	4	00000752								

Appendix 2

```

3      *
4      *      MC68HC11 functions.      *
5      *
6      *      Add, Subtract, Multiply, Divide,      *
7      *      MJD -> day, date, month and year      *
8      *
9      *      Used with RADE.S11, RDSE.S11 and RDRAME.S11.      *
10     *
11     *      P. Topping      6th March '93      *
12     *
13     *
14     *
15     *      EXPORT TRA,ADD,SUB,MULT,DIV,CLO,CLRAS,ADD1,MJDC,DNAME,MNAME
16     *
17     *      LIB      RDRAME.S11
17
17     00000000      PORTA      EQU      $00      PORT A ADDRESS
17     00000004      PORTB      EQU      $04      * B *
17     00000008      PORTC      EQU      $08      * C *
17     0000000C      PORTD      EQU      $0C      * D *
17     00000010      PORTE      EQU      $0A      * E *
17     00000014      PORTCDD      EQU      $07      PORT C DATA DIRECTION REG.
17     00000018      PORTIDD      EQU      $09      * D *
17     0000001C      TMSK2      EQU      $24
17     00000020      RACTL      EQU      $26
17     00000024      OPTION      EQU      $39
17     00000028      INIT       EQU      $3D
17
17     00001000      RBO         EQU      $1000      REGISTER BLOCK OFFSET
17     0000003B      PPRG       EQU      $3B      EEPROM CONTROL REGISTER
17     00000009      ND         EQU      9          No. DIGITS
17
17     *      SECTION.S .RAM1.COMM
17
17     00000000      BMJD       RMB      3          BINARY MJD
17     00000003      Q          RMB      9          WORKING NUMBER 1 - RDS
17     00000006      TMQ        RMB      9          SCRATCH
17     00000009      P          RMB      9          WORKING NUMBER 2 - RDS
17     0000000C      TMP        RMB      9          MULT. OVER. OR DIV. REMAINDER
17     0000000F      R          RMB      9          WORKING NUMBER 3 - RDS
17     00000012      MJD        RMB      9          MODIFIED JULIAN DAY NUMBER
17     00000015      YR         RMB      9          YEAR
17     00000018      MONTH      RMB      2          MONTH
17     0000001B      DOM        RMB      2          DATE
17     0000001E      DOW        RMB      1          DAY OF WEEK
17
17     *
17     *      RAM allocation, RDS & radio.      *
17     *
17     *
17
17     00000047      DIST       RMB      1          TRANSIENT DISPLAY TIMEOUT COUNTER
17     00000048      SLEPT      RMB      1          SLEEP TIMER MINUTES COUNTER
17     00000049      RDSTO      RMB      1          RDS TIMEOUT COUNTER
17     0000004A      PSNP       RMB      1          PS DISPLAY POINTER
17     0000004B      DAT        RMB      4          SERIAL DATA BUFFER
17     0000004F      TMPGRP     RMB      8          TEMPORARY GROUP DATA
17     00000057      GROUP      RMB      8          COMPLETE GROUP DATA
17     0000005F      PTY        RMB      1          PROGRAM-TYPE CODE (CURRENT)
17     00000060      PTYCHP     RMB      1          PROGRAM-TYPE CODE (PTY SCAN)
17     00000061      PI         RMB      2          PROGRAM IDENTIFICATION CODE
17     00000063      PION       RMB      2          PROGRAM IDENTIFICATION CODE (EON)
17     00000065      PIN        RMB      2          PROGRAM ITEM NUMBER
17     00000067      LEV        RMB      1          VALID BLOCK LEVEL
17     00000068      STY        RMB      1          BIT LEVEL
17     00000069      TMPI1      RMB      1          TEMP BYTE FOR USE IN IRQ
17     0000006A      SYN        RMB      2          SYNDROME
17     0000006C      CNF        RMB      1          SYNDROME CONFIDENCE
17     0000006D      TH32       RMB      1          TICS (SECONDS/32)
17     0000006E      TH8        RMB      1          EIGHTHS OF SECONDS
17     0000006F      SEC        RMB      1          SECONDS
17     00000070      MIN        RMB      1          MINUTES
17     00000071      OUR        RMB      1          HOURS
17     00000072      AMIN       RMB      1          ALARM MINUTES
17     00000073      AOUR       RMB      1          ALARM HOURS
17     00000074      DISP1      RMB      1          RT DISPLAY POINTER #1
17     00000075      DISP2      RMB      1          RT DISPLAY POINTER #2
17
17     00000076      RQ         RMB      6          WORKING BCD NUMBER 1 RADIO
17     00000077      RP         RMB      6          * 2 *
17     00000082      RR         RMB      6          * 3 *
17     00000088      W1         RMB      2          W
17     0000008A      W2         RMB      2          O
17     0000008C      W3         RMB      2          R
17     0000008E      W4         RMB      2          K
17     00000090      W5         RMB      2          I
17     00000092      W6         RMB      2          N
17     00000094      W7         RMB      2          G
17     00000096      KEY        RMB      1          CODE OF PRESSED KEY
17     00000097      KOUNT      RMB      1          KEYBOARD COUNTER
17     00000098      DIG2       RMB      1          2nd DIGIT TIMEOUT COUNTER
17     00000099      CARRY       RMB      1          BCD CARRY
17     0000009A      COUNT      RMB      1          LOOP COUNTER
17     0000009B      NUM1        RMB      2          1st No. POINTER (ADD & SUBTRACT)
17     0000009C      NUM2        RMB      2          2nd No. POINTER (ADD & SUBTRACT)
17     0000009F      LED         RMB      1          STATION NUMBER
17     000000A0      SMEN       RMB      2          CURRENT FREQUENCY
17     000000A2      REARET     RMB      1          LAST TA REASON FOR RETURN
17     000000A3      RTDIS      RMB      1          RDS DISPLAY TYPE
17     000000A4      DI         RMB      1          DECODER IDENTIFICATION
17     000000A5      SCHAN      RMB      1          SCAN CHANNEL

```

```

17      *          Flags, & pages 1-2.          *
17      *                                     *
17      .....,.....                          .....
17
17 000000a6       STAT   RMB     1             0: MODE 1: STATION, 0: FREQ
17              *                             1: STEP 1: 50KHz, 0: 10KHz
17              *                             2: CLRQ 1: CLEAR IF NO. KEYED
17              *                             3: TIMER MS BIT TOGGLE (64 Hz)
17              *                             4: RDS DATA CLEARNING ENABLE
17              *                             5: KEY FUNCTION PERFORMED
17              *                             6: KEY REPEATING
17              *                             7: NOT JUST POWERED UP
17 000000a7       STAT2  RMB     1            0: VALID SINEWAVE
17              *                             1: VALID GROUP
17              *                             2: RT DISPLAY
17              *                             3: UPDATE DISPLAY
17              *                             4: CLEAR DISPLAY
17              *                             5: SPACE FLAG
17              *                             6: NOT ON PROGRAM (AM)
17              *                             7: TA RETUNE DONE
17 000000a8       STAT3  RMB     1            0: NOT ON PROGRAM (FM)
17              *                             1: TEXTA/TEXTB BIT (RT)
17              *                             2: TA FLAG
17              *                             3: TP FLAG
17              *                             4: SHAFT DIRECTION
17              *                             5: SHAFT ROTATION
17              *                             6: UPDATE DATE
17              *                             7: SHAFI INTERRUPTS
17 000000a9       STAT4  RMB     1            0: DISPLAY (OR TA SWITCH) TRANSIENT
17              *                             1: SLEEP TIMER RUNNING
17              *                             2: TRAFFIC ENABLED
17              *                             3: ALARM DISPLAY
17              *                             4: ALARM ARMED
17              *                             5: ALARM SET-UP
17              *                             6: ALARM HOURS (SET-UP)
17              *                             7: VALID GROUP 14B RECEIVED
17 000000aa       STAT5  RMB     1            0: BAND CHANGE TIMEOUT
17              *                             1: RDS DISPLAYS
17              *                             2: SLEEP DISPLAY
17              *                             3: M/S 0: M, 1: S
17              *                             4: RETUNE FLAG (FREQUENCY MODE)
17              *                             5: TA INHIBIT FLAG (NMV)
17              *                             6: STORE MODE
17              *                             7: WEEKDAY ONLY ALARM
17
17 000000ab       STATE6 RMB     1            BAND/BANK (PM IF, . , A1,A0,,E6;
17 BCTO           RMB     1            BAND CHANGE TIMEOUT
17 SCNT           RMB     1            SHAFT DETENT COUNTER
17
17              SECTION .RANG.COMM
17 EON             RMB    256                EON DATA (16 NETWORKS)
17              SECTION .RANG.COMM
17 DISP           RMB    16                 LCD MODULE BUFFER
17 DISPP          RMB    16                 CURRENT LCD MODULE CONTENTS
17 PSN            RMB     8
17 RT             RMB    65                 RADIIOTEXT
17
17              SECTION .RM2
17
17      .....
17      * Transfer of BCD numbers.         *
17      *                                *
17      * (X) <- (NUM1), X preserved        *
17      *                                *
17      .....
17
17 00000000 >dff0 TRA STX NUM2 CLEAR DESTINATION
17 00000002 >bcd0000 JSR CLRAS AND ADD IT TO No. AT NUM1
17
17      .....
17      * Addition of BCD numbers.         *
17      *                                *
17      * (X) <- (NUM1) + (NUM2), X preserved *
17      *                                *
17      .....
17
17 00000005 >7f0000 ADD CLR CARRY
17 00000008 >dff0 STX W7
17 0000000a >dff0 AD STX W5 ANSWER POINTER
17 0000000c c609 LDAB #ND
17 0000000e >de00 LDX NUM1 1st No. POINTER
17 00000010 >dff0 STX W3
17 00000012 >de00 LDX NUM2 2nd No. POINTER
17 00000014 >dff0 STX W4
17 00000016 >de00 LDX W5
17 00000018 af08 LDA ND-1.X
17 0000001a 09 DEX
17 0000001b >dff0 STX W3
17 0000001d >de00 LIX W4
17 0000001f ab08 ADA ND-1.X ADD
17 00000021 09 DEX
17 00000022 >dff0 STX W4
17 00000024 >9b00 ADA CARRY SET ON ADDITION OVERFLOW
17 00000026 >7f0000 CLR CARRY OR POS. RESULT SUBTRACTION
17 00000029 8dl2 BSR ADJ DECIMAL ADJUST
17 0000002b >de00 LDX W5
17 0000002d af08 STAA ND-1.X SAVE ANSWER
17 0000002f 07 DEX
17 00000030 >dffc STX W5
17 00000032 5a DECB LOOP
17 00000033 26el BNE LPF DONE ?
17 00000035 >de00 LDX W7
17 00000037 39 RTS
17
17 00000038 800a AJ SUBA #10 YES, SUBTRACT 10
17 0000003a >7fc000 INC CARRY AND RECORD CARRY
17 0000003d 810a ADJ CMPLA #1C
17 0000003f 24f7 BHS A? 10 OR MORE ?
17 00000041 39 RTS NC

```

```

74
75
76
77
78
79
80
81
82
83
84 00000042 >df00
85 00000044 8d08
86 00000046 >7f0000
87 00000049 >7c0000
88 0000004c 8db0c
89
90 0000004e >de00
91 00000050 8d03
92 00000052 >de00
93 00000054 39
94
95 00000055 c609
96 00000057 8609
97 00000059 a008
98 0000005b a708
99 0000005d 09
100 0000005e 5a
101 0000005f 26f6
102 00000061 39
103
104 00000062 8df1
105 00000064 c609
106 00000066 6c11
107 00000068 a611
108 0000006a 810a
109 0000006c 2508
110 0000006e 800a
111 00000070 a711
112 00000072 09
113 00000073 5a
114 00000074 26f0
115 00000076 39
116
117
118
119
120
121
122
123 00000077 >ce0000
124 0000007a >bd0000
125 0000007d >ce0000
126 00000080 >bd0000
127 00000083 ce0012
128 00000086 >df00
129 00000088 ce0009
130 0000008b >a6ff
131 0000008d >df00
132 0000008f >9700
133
134 00000091 ce0009
135 00000094 >a6ff
136 00000096 >9700
137 00000098 2733
138 0000009a >9600
139 0000009c >9700
140 0000009e 4f
141 0000009f >740000
142 000000a2 2402
143 000000a4 >9b00
144 000000a6 >7d0000
145 000000a9 2705
146 000000ab >780000
147 000000ae 20ef
148 000000b0 09
149 000000b1 >df00
150 000000b3 >de00
151 000000b5 >abf6
152 000000b7 >bd0000
153 000000ba >a7f6
154 000000bc >9600
155 000000be >abf5
156 000000c0 >a7f5
157 000000c2 >9600
158 000000c4 >9700
159 000000c6 09
160 000000c7 >df00
161 000000c9 >de00
162 000000cb 2004
163 000000cd >7a0001
164 000000d0 09
165 000000d1 28c1
166 000000d3 >9601
167 000000d5 8b08
168 000000d7 >9701
169
170 000000d9 >de00
171 000000db 09
172 000000dc 26ad
173 000000de >ce0000
174 000000e1 39

```

```

*****
*
* Subtraction, complementing and incre-
* menting (X=REG-ND) of BCD numbers.
*
* (X) <- (NUM1) - (NUM2), X preserved.
* (X and NUM2 should not be equal)
*
*****

```

```

SUB STX W6 ANSWER POINTER
BSR COM2 9S COMP. SECOND NUMBER
CLR CARRY SET CARRY TO ONE
INC CARRY BEFORE ADDING
BSR AD ADD FIRST NUMBER

COM2 LDX NUM2 9S COMPLIMENT
BSR COMP SECOND NUMBER
LDX W6 RESTORE ANSWER POINTER
RTS

```

```

COMP LDAB #ND 9S COMPLIMENT
LOOP3 LDAA #S09
SUBA ND-1,X
STAA ND-1,X
DEX
DECB
BNE LOOP3
RTS

```

```

COM10 BSR COMP NINES COMPLIMENT THEN
ADD1 LDAB #ND ADD 1 FOR TENS COMPLIMENT
ADD2 INC 2*ND-1,X ENTER WITH X = REG-ND
LDAA 2*ND-1,X
CMPA #S0A
BLO RETURN
SUBA #10
STAA 2*ND-1,X
DEX
DECB
BNE ADD2
RETURN RTS

```

```

*****
*
* Mult., R <- P x Q, over. in TMP, X = #R.
*
*****

```

```

MULT LDX #R
JSR CLRAS
LDX #TMP
JSR CLRAS CLEAR RESULT
LDX #2*ND
STX W6 INIT. R POINTER
LDX #ND
LDAA P-1,X
STX W1 SAVE P POINTER
STAA CARRY SAVE P
* BEQ DCP
LDX #ND
LDAA Q-1,X
STX W4 INIT. Q POINTER
STAA W4 SAVE Q
BEQ TZ0 IF ZERO GOTO NEXT Q
LDAA CARRY RECALL P
STAA W3 SAVE P

PLY LSR CARRY RIGHT SHIFT INTO C
BCC SHF C = ZERO ?
ADDA W4 NO, A=A+Q
SHF TST CARRY ZERO ?
BEQ C4 YES, FINISHED WITH THIS Q
ASL W4 NO, LEFT SHIFT Q
BRA PLY

C4 DEX
W2 Q = Q + 1
STX W6 SAVE Q POINTER
LDX W6 R POINTER
ADDA R-ND-1,X WAS -(ND+1)
JSR ADJ ADJUST
STAA R-ND-1,X WAS -(ND+1)
LDAA CARRY
ADDA R-ND-2,X WAS -(ND+2)
STAA R-ND-2,X WAS -(ND+2)
LDAA W3 R-(ND+2) = R-(ND+2) + CARRY WAS -(ND+2)
STAA CARRY RECALL P
DEX SAVE IN CARRY
STX W6 SAVE R POINTER
LDX W2 Q POINTER
BRA C3

TZ0 DEC W6+1 DEC. R POINTER
DEX DEC. Q POINTER
BNE XTT

C3 LDAA W6+1 R POINTER
ADDA #ND-1
STAA W6+1 R = R + ND-1
*DCP DEX
LDX W1
DEX
P = P + 1
BNE STR IF NOT ZERO GOTO NEXT P
LDX #R
RTS

```

```

176
177
178
179
180
181
182
183
184
185 000000e2 >ce0000
186 000000e5 >bd0000
187
188 000000e8 >ce0000
189 000000eb >df00
190 000000ed >ce0000
191 000000f0 >bd0000
192 000000f3 >ce0000
193 000000f6 >df00
194 000000f8 >ce0000
195 000000fb >bd0000
196
197 000000fe c609
198 00000100 >ce0000
199 00000103 a600
200 00000105 2607
201 00000107 >bd0000
202 0000010a 26f4
203 0000010c 2035
204 0000010e >7f0000
205 00000111 >df01
206
207 00000113 >ce0000
208 00000116 >df00
209 00000118 >bd0000
210 0000011b >9600
211 0000011d 2706
212 0000011f >de00
213 00000121 >6c00
214 00000123 20ee
215 00000125 >ce0000
216 00000128 >bd0000
217 0000012b >ce0000
218 0000012e c608
219 00000130 a607
220 00000132 a708
221 00000134 09
222 00000135 5a
223 00000136 26f8
224 00000138 6f08
225 0000013a >7f0001
226 0000013d >9601
227 0000013f 810a
228 00000141 26d0
229 00000143 >ce0000
230 00000146 39
231
232
233
234
235
236
237
238
239
240
241 00000147 >ce0000
242 0000014a >bd0000
243 0000014d >ce0000
244 00000150 >bd0000
245
246 00000153 >cefff7
247 00000156 >bd0000
248 00000159 >cefff7
249 0000015c >bd0000
250 0000015f >ce0000
251 00000162 >bd0000
252 00000165 8607
253 00000167 >9708
254 00000169 >bd0000
255 0000016c >9608
256 0000016e >970c
257
258 00000170 >ce0000
259 00000173 >bd0000
260 00000176 >ce0000
261 00000179 >df00
262 0000017b >ce0000
263 0000017e >bd0000
264 00000181 >ce0000
265 00000184 >bd0000
266 00000187 >bd0000
267 0000018a >df00
268 0000018c >ce0000
269 0000018f >bd0000

```

```

*****
*
*   Division of BCD numbers.
*
*   R <- P / Q, remainder in TMP.
*   on exit X = #R, TMQ used.
*
*****

```

DIV	LDX	#R	CLEAR
	JSR	CLRAS	RESULT
	CLR	W2	SIGN
	LDX	#P	TRANSFER
	STX	NUM1	P TO
	LDX	#TMP	WORKING
	JSR	TRA	P (TMP)
	LDX	#Q	TRANSFER
	STX	NUM1	Q TO
	LDX	#TMQ	WORKING
	JSR	TRA	Q (TMQ)
POSS	LDAB	#ND	NUMBER DIGITS
LOOP6	LDX	#TMQ	FIND LEAST SIGNIFICANT
	LDAA	0,X	NON-ZERO DIGIT
	BNE	NOSH	ZERO ?
	JSR	SHIFT	YES, SHIFT Q
	BNE	LOOP6	UP ONE PLACE
ZQ	BRA	RTRN	Q WAS ZERO
NOSH	CLR	W1	SAVE
	STAB	W1+1	No. DIGITS - No. SHIFTS
SUBB	LDX	#TMP	SUBTRACT Q
	STX	NUM1	FROM
	JSR	SUB	P
	LDAA	CARRY	TOO FAR ?
	BEQ	NEXTD	IF YES, GO TO NEXT DIGIT
	LDX	W1	INCREMENT RELEVANT
	INC	R-1,X	DIGIT IN RESULT
	BRA	SUBB	ONCE AGAIN
NEXTD	LDX	#TMP	TOO FAR, ADD
	JSR	ADD	Q BACK ON
ROR	LDX	#TMQ	SET UP TO
	LDAB	#ND-1	SHIFT BACK WORKING Q
RRJ	LDAA	ND-2,X	MOVE ALL
	STAA	ND-1,X	DIGITS
	DEB		DOWN
	DECB		ONE PLACE
	BNE	RRJ	LOVE ?
	CLR	ND-1,X	CLEAR MS DIGIT
	INC	W1+1	INCREMENT POINTER
	LDAA	W1+1	
	CMPA	#ND+1	FINISHED ?
	BNE	SUBB	NO, NEXT DIGIT
RTRN	LDX	#R	
	RTS		

```

*****
*
*   MJD - day of week and year.
*
*   DOW = (MJD+2)MOD7 (= WD-1)          (DOW)
*   Y' = INT((MJD-15078.2)/3652500)      (YR)
*
*****

```

MJDC	LDX	#MJD	
	JSR	XFERP	P <- MJD
	LDX	#MJD	
	JSR	T10K	MJD <- MJD TIMES 10,000
DOFFW	LDX	#P-ND	
	JSR	ADD1	P <- MJD + 1
	LDX	#P-ND	
	JSR	ADD1	P <- MJD + 2
	LDX	#Q	
	JSR	CLRAS	
	LDAA	#7	
	STAA	Q+ND-1	Q <- 7
	JSR	DIV	R <- (MJD+2)/7
	LDAA	TMP+ND-1	REMAINDER (WD-1) IN TMP
	STAA	DOW	
YEAR	LDX	#CY	
	JSR	XFERQ	Q <- CY (150782000)
	LDX	#MJD	NUM2 <- (Q)
	STX	NUM1	
	LDX	#P	
	JSR	SUB	P <- 10K(MJD-15078.2)
	LDX	#DY	
	JSR	XFERQ	Q <- 3652500
	JSR	DIV	R <- Y' ((MJD-15078.2)/365.25)
	STX	NUM1	
	LDX	#YR	
	JSR	TRA	YR <- Y'

```

271
272
273
274
275
276
277
278
279
280 00000193 >bd0000
281 00000195 >ce0000
282 00000198 >bd0000
283 0000019b >ce0000
284 0000019e >df00
285 000001a0 >ce0000
286 000001a3 >bd0000
287 000001a6 >df00
288 000001a8 >ce0000
289 000001ab >df00
290 000001ad >ce0000
291 000001b0 >bd0000
292 000001b3 >ce0000
293 000001b6 >bd0000
294 000001b9 >bd0000
295 000001bc >bd0000
296 000001bf >9607
297 000001c1 >9700
298 000001c3 >9608
299 000001c5 >9701
300
301 000001c7 >bd0000
302 000001ca >df00
303 000001cc >ce0000
304 000001cf >bd0000
305 000001d2 >bd0000
306 000001d5 >df00
307 000001d7 >ce0000
308 000001da >df00
309 000001dc >bd0000
310 000001df >ce0000
311 000001e2 >bd0000
312 000001e5 >7f0005
313 000001e8 >ce0000
314 000001eb >df00
315 000001ed >ce0000
316 000001f0 >bd0000
317 000001f3 >df00
318 000001f5 >ce0000
319 000001f8 >df00
320 000001fa >ce0000
321 000001fd >bd0000
322 00000200 a604
323 00000202 >9701
324 00000204 a603
325 00000206 >9700
326
327
328
329
330
331
332
333
334
335
336
337 00000208 >9600
338 0000020a 2720
339 0000020c >9601
340 0000020e 2715
341 00000210 8104
342 00000212 2518
343 00000214 >ceff17
344 00000217 >bd0000
345 0000021a >7f0000
346 0000021d >7a0001
347 00000220 >7a0001
348 00000223 2007
349 00000225 860a
350 00000227 >9701
351 00000229 >7f0000
352 0000022c >7a0001
353 0000022f 39
354
355 00000230 >ce0000
356 00000233 8d58
357 00000235 >ce0000
358 00000238 8d5b
359 0000023a >bd0000
360 0000023d >7f0005
361 00000240 >7f0006
362 00000243 >7f0007
363 00000246 >7f0008
364 00000249 39
365
366 0000024a c605
367 0000024c a604
368 0000024e a700
369 00000250 08
370 00000251 5a
371 00000252 26f8
372 00000254 6f00
373 00000256 6f01
374 00000258 6f02
375 0000025a 6f03
376 0000025c 39

```

```

*****
*
*      MJD - month and day.
*
*      M' = INT((MJD-14956.1-INT(Y'*365.25))/306001)      (P)
*      D = MJD-14956-INT(Y'*365.25)-INT(M'*30.6001)      (Q(x10K))
*
*****

MONTH JSR INT      R <- 10K(INT(Y'*365.25))
      LDX #D01
      JSR XFERP      P <- 149561000
      LDX #MJD      NUM2 <- (P)
      STX NUM1
      LDX #Q
      JSR SUB      Q <- 10K(MJD-14956.1)
      STX NUM1
      LDX #R
      STX NUM2
      LDX #P
      JSR SUB      P <- 10K(MJD-14956.1-INT(Y'*365.25))
      LDX #DM
      JSR XFERQ      Q <- 306001
      JSR DIV      R <- M' ( MJD-14956.1-INT(Y'*365.25) )
      JSR XFERP      P <- M' INT ( ----- )
      LDA P-ND-2      SAVE M' ( 306001 )
      STAA MTH
      LDA P-ND-1
      STAA MTH-1

DAY JSR MULTI      R <- 10K(INT(M'*30.6001))
    STX NUM1
    LDX #TMO
    JSR TRA      TMO <- 10K(INT(M'*30.6001))
    JSR INT      R <- 10K(INT(Y'*365.25))
    STX NUM2
    LDX #TMO
    STX NUM1
    JSR ADD      TMO <- 10K(INT(Y'*365.25)+INT(M'*30.6001))
    LDX #D01
    JSR XFERP      P <- 149561000
    CLR P-ND-4      P <- 149560000
    LDX #TMO      NUM2 <- (P)
    STX NUM1
    LDX #R
    JSR ADD      R <- 10K(14956-INT(Y'*365.25)+INT(M'*30.6001))
    STX NUM2
    LDX #MJD
    STX NUM1
    LDX #Q
    JSR SUB      Q <- MJD-R (10K*DM)
    LDA ND-5,X
    STAA DM+1      MJD-14956-INT(Y'*365.25)-INT(M'*30.6001)
    LDA ND-6,X
    STAA DM

*****
*
*      MJD - final correction of year & month and subs.
*
*      If M' = 14 or 15, then K = 1, else K = 0
*      Y = Y' + K
*      M = M' - 1 - K*12
*
*****

ADJU LDA MTH      MONTH, MSD
    BEQ KE02      0 ?
    LDA MTH+1      NO, M' = 10 THRU 15
    BEQ KE01      0 ?
    CMPA #4      NO, M' = 11 THRU 15
    BLO KE02      LESS THAN 14
    LDX #YR-ND      NO, M' = 14 OR 15, K=1
    JSR ADD1      Y <- Y'+1
    CLR MTH      MONTH, MSD (-10)
    DEC MTH+1      DEC. MONTH
    DEC MTH+1      AND AGAIN (-2)
    BRA KE02      -12
    KE01 LDA #10      M' = 10
    STAA MTH+1      PUT 10 IN LSD
    CLR MTH      CLEAR MSD
    KE02 DEC MTH+1      9<-10, 1,2<-14,15, 3-<4-9, 10-12<-11-13
    RTS

INT LDX #YR
    BSR XFERP      P <- Y'
    LDX #DY
    BSR XFERQ      Q <- 10K*365.25
    JSR MULT      R <- 10K*Y'*365.25
    CLR R-ND-4
    CLR R-ND-3
    CLR R-ND-2
    CLR R-ND-1
    RTS      R <- 10K(INT(Y'*365.25))

T10K LDA B      TIMES 10,000
SLP LDA 4,X
    STAA 0,X
    INX
    DECB
    BNE SLP
    CLR 0,X
    CLR 1,X
    CLR 2,X
    CLR 3,X
    RTS

```

```

378
379
380
381
382
383
384 0000025d >ce0000      CLQ    LEX    #Q      CLEAR Q
385 00000260 >df00      CLRAS  STX    W5      CLEAR No. DIGITS STARTING AT X
386 00000262 c609      CR      LDAB   WND
387 00000264 6f00      CR      CLR    0,X
388 00000266 08      CR      INX
389 00000267 5a      CR      DECB
390 00000268 26fa      CR      BNE    CR      DONE ?
391 0000026a >de00      CR      LEX    W5
392 0000026c 39      CR      RTS
393
394 0000026d >9700      SHIFT  STAA   W3      W1: MSD, W2: LSD
395 0000026f 8d11      BSR    DR1
396 00000271 >de00      LEX    W1
397 00000273 a601      AGS    LDAA   1,X      MOVE ALL DIGITS
398 00000275 a700      STAA   0,X      UP ONE PLACE
399 00000277 08      CR      INX
400 00000278 >9c00      CPX    W2
401 0000027a 26f7      BNE    AGS      DONE ?
402 0000027c >9600      LDAA   W3      YES, RECOVER NEW DIGIT
403 0000027e a700      STAA   0,X      AND PUT IT IN LSD
404 00000280 5a      CR      DECB
405 00000281 39      CR      RTS
406
407 00000282 >df00      DR1     STX    W1      STORE POINTERS
408 00000284 8608      LDAA   #ND-1      (USED IN DIGIT AND DQ)
409 00000286 08      AXL     INX
410 00000287 4a      CR      DECA
411 00000288 26fc      BNE    AXL
412 0000028a >df00      STX    W2
413 0000028c 39      CR      RTS
414
415 0000028d >df00      XFERP   STX    NUM1
416 0000028f >ce0000      LEX    #P
417 00000292 >7e0000      JMP     TRA      NUM2 <- (P)
418
419 00000295 >df00      XFERQ   STX    NUM1
420 00000297 >ce0000      LEX    #Q
421 0000029a >7e0000      JMP     TRA      NUM2 <- (Q)
422
423 0000029d 4d5f6e5475655765  DNAME  FCC      'MonTueWedThuFriSatSun'
424 000002b2 696e76      FCC      'inv'
425 000002b5 4a616e4665624d61  MNAME  FCC      'JanFebMarAprMayJunJulAugSepOctNovDec'
426
427 000002d9 0105000708020000  CY      FCB      1,5,0,7,8,2,0,0,0,0
428 000002e2 0000030605020500  DY      FCB      0,0,3,6,5,2,5,0,0,0
429 000002eb 0104090506010000  DD1     FCB      1,4,9,5,6,1,0,0,0,0
430 000002f4 0000000300060000  DM      FCB      0,0,0,3,0,6,0,0,0,1
431
432
END

```

Section synopsis

```

1 000000ae ( 174) .RAM1
2 00000100 ( 256) .RAM2
3 0000006d ( 109) .RAM3
4 000002fd ( 765) .ROM2

```

Symbol table

.RAM1	1	00000000	COUNT	1	0000009a	KEY	1	00000096	PSNP	1	0000004a	STAT4	1	000000a9			
.RAM2	2	00000000	CR	4	00000264	KOUNT	1	00000097	PTY	1	0000005f	STAT5	1	000000aa			
.RAM3	3	00000000	CY	4	000002d9	LED	1	0000009f	PTYCMP	1	00000060	STAT6	1	000000ab			
.ROM2	4	00000000	DAT	1	0000004b	LEV	1	00000067	Q	1	00000003	STR	4	000000hb			
AD	4	0000000a	DAY	4	000001c7	LOOP	4	00000016	R	1	00000027	SUB	4	00000042			
ADD	E	4	00000005	DI	1	000000a4	LOOP3	4	00000057	RDESTO	1	00000049	SUBB	4	00000113		
ADD1	E	4	00000064	DIG	4	00000098	LOOP6	4	00000100	REARET	1	000000a2	SYN	1	0000006a		
ADD2	4	00000066	DISP	3	00000000	MIN	1	00000070	RETURN	4	00000076	T10K	4	0000024a			
ADJ	4	0000003d	DISP1	1	00000074	MJD	1	00000030	ROR	4	0000012b	THS2	1	0000006d			
ADJU	4	00000208	DISP2	1	00000075	MJDC	E	4	00000147	RP	1	0000007c	TH8	1	0000006e		
AGS	4	00000273	DISPP	3	00000010	MNAME	E	4	000002b5	RQ	1	00000076	TMP	1	0000001e		
AJ	4	00000038	DIST	1	00000047	MNTH	1	00000042	RR	1	00000082	TMPGRP	1	0000004f			
AMIN	1	00000072	DIV	E	4	000000e2	MONTH	4	00000192	RRJ	4	00000130	TMQ	1	0000000c		
ACUR	1	00000073	DM	E	4	00000214	MULT	E	4	00000077	RT	3	00000028	TRA	E	4	00000000
AXL	4	00000286	DNAME	E	4	0000029d	MULTI	4	0000023a	RTDIS	1	000000a3	TZC	4	000000cd		
BCTO	1	000000ac	DD1	4	000002eb	NEXTD	4	00000125	RTRN	4	00000143	W1	1	00000086			
BIT	1	00000068	DDFFW	4	00000153	NOSH	4	0000010e	SCHAN	1	000000a5	W2	1	0000008a			
BMUD	1	00000000	DDM	1	00000044	NUM1	1	0000009b	SCNT	1	000000ad	W3	1	0000008c			
C2	4	000000ba	DDW	1	00000046	NUM2	1	0000009d	SEC	1	0000006f	W4	1	0000008e			
C3	4	000000d1	DR1	4	00000282	OUR	1	00000071	SHF	4	000000a6	W5	1	00000090			
C4	4	000000b0	DY	4	000002e2	P	1	00000015	SHIFT	4	0000026d	W6	1	00000092			
CARRY	1	00000099	ECN	2	00000000	PI	1	00000061	SLEPT	1	00000048	W7	1	00000094			
CLQ	E	4	0000025d	GROUP	1	00000057	PIN	1	00000065	SLP	4	0000024c	XFERP	4	0000028d		
CLRAS	E	4	00000260	INT	4	00000230	PLCN	1	00000063	SMB	1	000000a0	XFERQ	4	00000295		
CMK10	4	00000062	ITMP1	1	00000069	PLX	4	0000009f	STAT	1	000000a6	XTT	4	0000009a			
CMK2	4	0000004e	KE01	4	00000225	PCSS	4	0000000e	STAT2	1	000000a7	YEAR	4	00000170			
COMP	4	00000055	KE02	4	0000022c	PSN	3	00000020	STAT3	1	000000a8	YR	1	00000039			
CONF	1	0000006c	KE1	4	00000214	PSNF	1	0000004a									

How to Reach Us:

Home Page:

www.freescal.com

E-mail:

support@freescal.com

USA/Europe or Locations Not Listed:

Freescal Semiconductor
Technical Information Center, CH370
1300 N. Alma School Road
Chandler, Arizona 85224
+1-800-521-6274 or +1-480-768-2130
support@freescal.com

Europe, Middle East, and Africa:

Freescal Halbleiter Deutschland GmbH
Technical Information Center
Schatzbogen 7
81829 Muenchen, Germany
+44 1296 380 456 (English)
+46 8 52200080 (English)
+49 89 92103 559 (German)
+33 1 69 35 48 48 (French)
support@freescal.com

Japan:

Freescal Semiconductor Japan Ltd.
Headquarters
ARCO Tower 15F
1-8-1, Shimo-Meguro, Meguro-ku,
Tokyo 153-0064
Japan
0120 191014 or +81 3 5437 9125
support.japan@freescal.com

Asia/Pacific:

Freescal Semiconductor Hong Kong Ltd.
Technical Information Center
2 Dai King Street
Tai Po Industrial Estate
Tai Po, N.T., Hong Kong
+800 2666 8080
support.asia@freescal.com

For Literature Requests Only:

Freescal Semiconductor Literature Distribution Center
P.O. Box 5405
Denver, Colorado 80217
1-800-441-2447 or 303-675-2140
Fax: 303-675-2150
LDCForFreescalSemiconductor@hibbertgroup.com

Information in this document is provided solely to enable system and software implementers to use Freescal Semiconductor products. There are no express or implied copyright licenses granted hereunder to design or fabricate any integrated circuits or integrated circuits based on the information in this document.

Freescal Semiconductor reserves the right to make changes without further notice to any products herein. Freescal Semiconductor makes no warranty, representation or guarantee regarding the suitability of its products for any particular purpose, nor does Freescal Semiconductor assume any liability arising out of the application or use of any product or circuit, and specifically disclaims any and all liability, including without limitation consequential or incidental damages. "Typical" parameters which may be provided in Freescal Semiconductor data sheets and/or specifications can and do vary in different applications and actual performance may vary over time. All operating parameters, including "Typicals" must be validated for each customer application by customer's technical experts. Freescal Semiconductor does not convey any license under its patent rights nor the rights of others. Freescal Semiconductor products are not designed, intended, or authorized for use as components in systems intended for surgical implant into the body, or other applications intended to support or sustain life, or for any other application in which the failure of the Freescal Semiconductor product could create a situation where personal injury or death may occur. Should Buyer purchase or use Freescal Semiconductor products for any such unintended or unauthorized application, Buyer shall indemnify and hold Freescal Semiconductor and its officers, employees, subsidiaries, affiliates, and distributors harmless against all claims, costs, damages, and expenses, and reasonable attorney fees arising out of, directly or indirectly, any claim of personal injury or death associated with such unintended or unauthorized use, even if such claim alleges that Freescal Semiconductor was negligent regarding the design or manufacture of the part.