An HC11-Controlled Multiband RDS Radio

By Peter Topping
Motorola, Ltd.
East Kilbride, Scotland

This application note describes the software and hardware features of the microcontroller (MCU) of a synthesized multiband radio which includes RDS (radio data system) decoding (FM, band II). It uses an MC68HC(7)11 MCU whose program can be on-chip or contained in an external EPROM (erasable programmable read-only memory). ROM versions are available.

Both LCD (liquid crystal display) and VFD (vacuum fluorescent display) 16-character dot matrix display modules can be used to display RDS and tuning information. Traffic messages, initiated by the reception of EON (enhanced other networks) data (group 14B) or TA = TP = 1 (traffic announcement = traffic program = 1) on the current frequency, are handled. The station carrying the TA is tuned for the duration of the message, followed by a return to the original frequency. A tuning knob employing an incremental encoder is supported.

Introduction

Figure 1 shows a block diagram of the application. The controller hardware and software are described in detail here. The other hardware is not covered to the same depth, because that varies between different implementations, the intention being to describe a controller which could
be added to an existing radio or to one which includes only one or two of the possible bands.

Separate FM and AM PLLs (phase-locked loop) are shown. This is not essential, but it reduces the amount of band switching necessary and simplifies hardware fault finding. The illustrated configuration corresponds to that used by the author for software development and debugging.

**Figure 1. Main Block Diagram**

The MCU used is the MC68HC(7)11. The MC68HC711K4 (K4) [and similar chips such as the MC68HC711P2 (P2) and MC68HC711PH8 (PH8)] can be used in expanded mode, but this application has been included in the ROM of an E32 and a PH8.

To use the ROMed parts in this application, the first three bytes of EEPROM (electrically erasable programmable ROM) should contain an extended jump to the appropriate start address. The E32 (ZC403311) requires $7E, $90, and $00 at addresses $B600, $B601, and $B602, while the PH8 (ZC428200 or ZC428202) requires $7E, $40, and $00 at
addresses $0D00, $0D01, and $0D02. This can be done using either PCbug11 or the BUFFALO (bit users fast friendly aid to logical operation) monitor (see reference 5). The E32 version uses all the input/output (I/O) and can, therefore, be used only in single-chip mode. The circuit diagram of the HC11E controller is shown in Figure 3 and the circuit diagram of the K4/PH8 in Figure 4. The K4/PH8 version shows the additional hardware (within the dotted line) used to develop and debug the software on a K4 using PCbug11. This implementation uses two of the K4’s chip selects to enable external memories allowing debug to be done with the code in RAM and the PCbug11 talker in an EPROM. This arrangement requires a further four I/O (input/output) lines, leaving 30 for use in the application. The description of the application, and the listed software, corresponds to the E32 ROMed version (ZC403311). Later sections list the port allocation and functional differences which apply to the PH8 ROMed versions (ZC428200 and ZC428202).

Forty programs (10 on FM, 10 on MW and 20 on SW) can be stored using the HC11E’s on-chip EEPROM (the PH8 has 20 additional SW (shortwave) programs). Each contains frequency, an 8-character name [PS (program service) name on a station with RDS] and, on FM only, PI (program identification) code and a TA inhibit bit. For stations with no RDS (for example, all AM stations), the saved name can be manually entered. Programs saved with no name use their frequency instead. The SW banks are selected by an I/O line (two for the PH8). When the MCU is reset, or any of the band or memory select inputs are changed, the last used program in the selected band is tuned. This feature does not require that the MCU is permanently powered up, as this information is also stored in non-volatile EEPROM.

The keyboard uses an MC14028 decoder to minimize the number of I/O lines used. Either LCD or VFD 16-digit dot matrix displays can be used. The VFD display driver supported is the MSC7128, and the LCD driver the HD44780. This driver on its own provides a 16-way multiplexed LCD. In conjunction with an HD44100, it can facilitate an 8-way multiplexed higher contrast display. The input level on a port pin selects the appropriate type of multiplexing to match the display in use. To minimize the I/O activity, only one display is driven, the choice between LCD and VFD again being determined by an I/O line.
MC145170 and MC145157 PLLs are supported, using the same data and clock lines as the VFD driver, along with dedicated chip selects. The MC145157 requires an external prescaler for frequencies above 20 MHz, but the MC145170 has an on-chip 160-MHz capability.

A tuning knob can be included by using an incremental encoder. This can utilize either IRQ or XIRQ. As IRQ is used for the RDS clock, XIRQ is most appropriate for the tuning function. The possibility of using IRQ (see information described later) has been included to facilitate debug with PCbug11, which can employ XIRQ for its communication with the PC. Edges detected on the encoder execute the PS edit and alarm setup functions of the +/- (plus/minus) keys, depending on the direction of rotation. This provides a quick and convenient method of editing the PS name and changing the alarm time. A difference in function between the encoder and the +/- keys applies in normal mode. The program number is not affected by the tuning knob. In this mode, when the +/- keys control the program number, the tuning knob increments or decrements the frequency.

Two I/O lines are used to select the band. These lines are regularly monitored; if they change, the radio is retuned to the last used station in the selected band. Table 1 shows the bands which are available.

- Band 2 is intended for single-conversion (low IF, intermediate frequency) MW or SW radios. The large step size of 9 or 10 kHz is suitable for MW rather than SW, but the small step size of 1 kHz is suitable for either SW or MW.
- Band 3 is for dual-conversion (10.7-MHz first IF) SW designs. The FM IF offset is selected as + or –, according to the level on port A, bit 2 (high: LO high; low: LO low).
- Bands 0 and 1 both are intended for VHF/FM, the difference between them being in the use of the M68HC11’s IRQ pin. It is possible to use IRQ interrupts for both RDS and the tuning knob, as the two functions are not required simultaneously. To facilitate this, the band-select inputs affect the function performed when an edge is detected in the IRQ pin. When band 0 is selected, an RDS bit is read, but in any other band the incremental encoder function is performed. This enables automatic selection of function if bit 0
on port A is taken high when movement is detected from the shaft encoder. This facility can be disabled (RDS function only) by holding bit 3 of port A low. This should be done if XIRQ is being used for the tuning knob. As XIRQ is level-sensitive, some additional components are required to interface it with the incremental encoder. Figure 2 shows a simple circuit which can be used for this purpose.

Table 1. Available Bands

<table>
<thead>
<tr>
<th>Band</th>
<th>PA1</th>
<th>PA0</th>
<th>IF Offset</th>
<th>Step</th>
<th>Memory</th>
<th>Use</th>
<th>Prescaler MC145157 Only</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>+/-10,700</td>
<td>50, 10</td>
<td>10</td>
<td>VHF</td>
<td>10</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>1</td>
<td>+/-10,700</td>
<td>50, 10</td>
<td>10</td>
<td>VHF</td>
<td>10</td>
</tr>
<tr>
<td>2</td>
<td>1</td>
<td>0</td>
<td>455</td>
<td>9 (or 10), 1</td>
<td>10</td>
<td>MW/SW</td>
<td>—</td>
</tr>
<tr>
<td>3</td>
<td>1</td>
<td>1</td>
<td>10,700</td>
<td>5</td>
<td>20/40</td>
<td>SW</td>
<td>5</td>
</tr>
</tbody>
</table>

Figure 2. Incremental Encoder Interface Circuit
Figure 3. MC68HC11E32 Circuit
Synthesis of the local oscillator (LO) in a superheterodyne radio provides many advantages over mechanical tuning. The main benefits are:

- Tuning accuracy
- Stability
- Storing of often-used frequencies.

The accuracy and stability result from the fact that the LO is phase-locked to a crystal oscillator. In conjunction with RDS, frequency synthesis provides the additional facility of allowing the radio to retune itself to a traffic announcement or news bulletin. A synthesizer can be retrofitted to most radios by replacing the tuning capacitor with a varicap diode. The voltage biasing the varicap is supplied by the synthesizer and also can be used to provide RF (radio frequency) tuning. Alternatively, manual preselector or no RF tuning can be employed.

Motorola’s MC145157 and MC145170 synthesizers are two of a series offering a variety of options including serial or parallel interfacing and single or dual modulus prescaling. The MC145157 requires a prescaler for frequencies above 20 MHz but the MC145170 can handle input frequencies up to 160 MHz. The MC145157 has been included to retain compatibility with hardware developed for use with the MC68HC05B4 synthesizer described in ANE416 (reference 1).

Figure 5 shows the block diagram of the MC145170. It uses the Motorola bitgrabber system, whereby the number of bits sent determines the register which is written to. There is, therefore, no need for the control bit which is required by the MC145157.
The reference counter divides the 8-MHz crystal oscillator (10 MHz for the MC145157) down to the reference frequency (in this case, 1 kHz for the MC145157 and 10 kHz for the MC145170) at which the comparison is made with the (also divided down) local oscillator. The filtered output of the phase comparator supplies the tuning voltage to the local oscillator. The numbers chosen as the divide ratios determine the frequency at which this oscillator stabilizes. The equation that follows shows the relationship between the various frequencies where P is the LO prescaler (MC145157 only). The received frequency can be changed by altering the LO divide ratio. The MCU takes care of the decimal-to-binary conversion, IF offset, and the other arithmetic required.

\[
\text{LO frequency} = \text{RF} + \text{IF} = P \times \left( \frac{\text{Xtal frequency}}{\text{(ref. divide ratio)}} \right) \times \text{LO divide ratio}
\]

The MC145157 is specified to operate up to 20 MHz, so prescaling is required on FM and SW (10.7-MHz IF). For this SW band, divide-by-5...
prescaling is used; for FM, divide-by-10 is used. This increases the minimum step size to 10 kHz of FM, which is ideal for this band, and to 5 kHz on SW, which is suitable for almost all broadcast stations. The MC145170 does not require any prescaling even on the FM band and can use this to advantage by allowing the use of a higher reference frequency, making the low-pass filter design less critical.

An important part of any PLL is the loop filter. The filter in Figure 6 is an active filter using the double-ended phase detector outputs from the MC145170 feeding a CA3460 operational amplifier. This dual op-amp allows the simple double-ended low-pass filter to be followed by a second order Sallen and Key filter. An active filter has the added advantage of increasing the available voltage swing beyond the supply rail of the MC145170/MC145157.

Figure 6. MC145170 Circuit
The combination of active filter and double-ended phase detector outputs makes it simple to select the correct relationship between voltage and frequency. Usually, the fixed side of the varicap diode is grounded, so increased voltage increases the frequency of the oscillator; in some oscillator designs, the fixed side may be taken to the supply rail, and increasing the voltage will decrease the frequency. With the filter design shown here, the choice can be made simply by swapping the phase detector outputs from the PLLs.

Radio Data System

The radio data system (RDS) 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, see reference 2). An MC68HC05E0 implementation of RDS is described in AN460, (reference 5). It monitors the RDS activity on the MPX signal of a VHF radio but is not able to tune the radio and, therefore, cannot, use AF (alternative frequencies) or EON data. This application can tune the radio and uses EON data to retune the radio when a traffic announcement is taking place on another frequency. An announcement is initiated by a packet 14B and the radio retunes if TAs are enabled. At the end of the TA, 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 subset of the RDS features; a list of all currently defined features is shown in Table 2.
The retrieval of data is carried out by demodulation hardware, which generates clock and data signals that can be used by the MCU. Suitable devices which can perform this function include SAA6579, SAA7579T (plus an external filter), TDA7330, LA2231, and RDS hybrids.

### Table 2. RDS Features

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

This application supports PI, PTY, PS, RT, CT, TA, TP, 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 5).

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 appropriate only 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. PI is a 2-byte number which identifies the
country, coverage area, and service. It can be used by the control MCU but is not normally intended for display. A change in PI code causes the initialization of all RDS data as it indicates that the radio has been retuned. This application facilitates the display of the current PI code.

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 MCU. In this example, PTY can be displayed on request; Table 3 shows the display used for each PTY code.

PS is the 8-character name of the station and is permanently displayed (except in standby mode). In the absence of RDS (for example, AM bands), the name can be entered manually. If none is entered, then the frequency is used as the station name when the program is stored in EEPROM.

Radiotext (RT) constitutes a string of up to 64 characters which give additional information regarding the service or program currently being transmitted. In this application, RT is displayed on request on the 16-digit dot matrix displays, using scrolling. The data often contains extra spaces to center the text on a 2 x 32 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.

CT (clock time and date) data is transmitted every minute on the minute and provides a very accurate clock, traceable to national standards. The (modified Julian) date and local time variation are also transmitted. Time is permanently displayed. In standby mode (see information later), the date is displayed instead of the PS name. The MJD number, which is the form in which the date is received, can also be displayed. The MCU converts this number into day-of-week, day-of-month, month and year.

AF would be used by a car radio to retune to the strongest signal carrying the selected service. AF data, along with TDC (transparent data channel) and INH (in-house data), is not used in this application.

TA and TP are flags. TP is set if the transmitter normally carries traffic information and TA is set if a traffic announcement is in progress. The
combination — TA = 1 and 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 (for instance, cassette when a TA occurs).

### Table 3. PTY Types

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

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.
Decoder information (DI) constitutes four bits indicating the type of transmission (mono, stereo, binaural, etc.). Currently, it is not in use in the United Kingdom, but it can be displayed as a number between 1 and 15.

Program item number (PIN) is used to identify the program 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 program. PIN can be displayed as four hexadecimal digits or fully decoded to day of month and time.

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. Type 14A groups are used to send information about other networks. The PS name and principal frequency of up to 16 other networks can be displayed. Type 14B groups are used to switch to traffic announcements; they include the PI code of the station carrying the announcement. This PI code is searched for in NVM, and the required station is 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.

**Keyboard**

The keyboard has 23 keys. Table 4 shows the layout and Table 5 contains a summary of key functions against mode.

### Table 4. Keyboard Layout

<table>
<thead>
<tr>
<th>PE0</th>
<th>PE1</th>
<th>PE2</th>
<th>PE3</th>
</tr>
</thead>
<tbody>
<tr>
<td>Q6</td>
<td>RDS</td>
<td>Traffic</td>
<td>MW step</td>
</tr>
<tr>
<td>Q5</td>
<td>Time colon</td>
<td>Sleep</td>
<td>—</td>
</tr>
<tr>
<td>Q4</td>
<td>7</td>
<td>8</td>
<td>9</td>
</tr>
<tr>
<td>Q3</td>
<td>4</td>
<td>5</td>
<td>6</td>
</tr>
<tr>
<td>Q2</td>
<td>1</td>
<td>2</td>
<td>3</td>
</tr>
<tr>
<td>Q1</td>
<td>0</td>
<td>Manual</td>
<td>Clear/Step</td>
</tr>
</tbody>
</table>
Application Note

The following functions are available.

**On/Off**

This key is intended as an on/off control for the radio. It sets a port line low for on and high for standby and can be used to control the power supply to the radio. Its status affects the behavior of other keys as described later in this application note.

**Sleep**

When pressed, the 1-hour sleep timer starts, leaving the standby line low (radio on) until the sleep time has elapsed. At this time, the line is switched to the standby mode (high). In the normal display mode, the sleep timer running causes the decimal point to appear on the display modules' first character. The sleep timer can be cancelled by pressing ON/OFF. The sleep time can be reduced in increments of five minutes by repressing or holding down the SLEEP key.

**Alarm**

The alarm key selects the alarm display mode and toggles the alarm armed status. When the alarm is not armed, the legend ALARM-OFF is displayed. When it is armed, the alarm time is shown and adjustment of the alarm setup can be done by selecting the field (5/7 day, hours, or minutes) with the STORE/SET key. The selected field (hours or minutes) flashes and can be adjusted with the +/- keys or the tuning knob. The alarm setup display returns to normal three seconds after the last adjustment. If the radio is in standby mode and the alarm is set, the alarm time is displayed instead of the date. The radio will come fully on (standby line low) at the alarm time. After a 500-ms delay to allow power supplies to stabilize, the program which was tuned when the radio was last used is retuned. When set to the 5-day alarm, the alarm will not occur on Saturdays and Sundays.

**Plus (+) and Minus (–)**

Pressing + or –, while in normal mode, increments or decrements the program number. The program number wraps round at 0 and 9. The mute line is set high before retuning and returned low 100 ms after the new frequency has been sent to the PLL. Changing the tuned program using the +/- keys (or the 0–9 keys) disables PS name clearing if RDS information is absent or contains multiple errors.
In PS-edit mode (see entry that follows), the + and – keys are used to change the character at the cursor position. This function is duplicated on the tuning knob incremental encoder. In the alarm setup mode, the + and – keys are used to change the alarm time as described earlier. The field which is currently selected for adjustment (using the STORE key) flashes. This function is duplicated on the tuning knob also.

In manual mode, these keys increment and decrement the current frequency in steps of 10 kHz or 50 kHz (FM) as selected by the CLEAR/STEP key. The default is 10 kHz. On the SW band, 1-kHz (455 kHz IF only) or 5-kHz steps are available; on the MW/LW band, 1- or 9-kHz steps are available. In the U.S.A., 10 kHz is appropriate instead of 9 kHz; this can be selected with a special key (see entry that follows). This function is duplicated on the tuning knob both in this mode and in normal mode. Use of the +/- keys (or the incremental encoder) to adjust the frequency enables PS name clearing if RDS information is absent or contains multiple errors. In normal mode, on the AM bands, use of the tuning knob displays the frequency in the PS name field, facilitating simultaneous display of frequency and time.

**Store/Set**

In normal modes (not manual or alarm), the store key selects the PS-edit mode in which the first character of the displayed PS-name flashes and can be changed by the + and – keys or the tuning knob. Subsequent presses of STORE move to the next character. A space is shown as a hyphen (–). This mode returns to the normal display mode 10 seconds after the last key press. This mode can be used to give a name to a station with no RDS PS name (all AM stations or an FM station with no RDS or with RDS or unusable quality). See the entry that follows for the method of saving this name in EEPROM. Entry of a PS name in this way requires that PS name clearing is disabled. This is achieved by changing the program number (by using the +/- or 0−9 keys). Fine tuning enables PS name clearing (see +/- key description). Direct frequency entry does not affect the PS name clearing status.

In the alarm setup mode, STORE selects what will be changed when the + or – keys or the tuning knob are used (5/7 day, hours, or minutes). Hours or minutes flash when they are selected.
In manual mode, STORE enters a special manual store mode in which the 9–0 keys save, rather than recall, a program. After pressing STORE, the program number flashes to indicate this change of function. Alternatively, a second press of STORE saves the current tuning information into the current program number. The current frequency, PI code (FM), PS name, and TA inhibit flag (FM) are saved in EEPROM. The TA inhibit status can be changed using the TRAFFIC key (see entry that follows). If the PS edit mode has been used, then manual store mode should be used to save the entered PS name.

**Manual**

Select manual entry of frequency; a second press returns to normal mode if the tuned frequency has not been changed. If it has been changed, the second press retunes to the new frequency and an additional press is required to return to the normal mode. In manual mode, frequency is displayed instead of the time; the + and – keys or the tuning knob enable incrementing and decrementing of the current frequency. Direct entry of frequency can be made using 0–9 keys. In this mode, the STORE key enters the manual store mode in which the program number flashes, allowing storing of the tuned program and PS name into the current, or a different, program number. A second press of STORE saves the current frequency, PS name, PI code, and TA inhibit bit (FM) in EEPROM.

In manual mode, the TRAFFIC (TA) key controls the TA inhibit bit, which can be stored with each program. If the current station has its TP flag high, the least significant digit of the frequency will alternate with a decimal point. Pressing TP toggles the NVM inhibit bit. When inhibited, the decimal point between the MHz and kHz becomes a "—". A subsequent press of STORE saves this bit in NVM along with the frequency, PI code, and PS name.

**0 to 9**

These keys are used both for direct frequency entry and for recalling the 40 available programs. In all modes, except standby and manual, when a 0–9 key is pressed, the selected program is tuned. Changing the tuned program using the 0–9 keys (or the +/- keys) disables PS name clearing if RDS information is absent or contains multiple errors. In manual mode, these keys are used for the direct entry of frequency. After entering the required frequency, pressing MANUAL retunes to the new frequency.
The mute line is set high before retuning and returned low 100 ms after the new frequency has been sent to the PLL. In manual store mode, the program number flashes and the 0–9 keys save the tuned program into the selected program number in EEPROM.

**RDS**

The first press displays scrolling RT data. Subsequent presses display PTY code, PI code, TA and TP, PIN code (two formats), MJD, MS and DI, last TA PI code, the reason for returning from last TA and EON (up to 16 networks with their principal frequency). See Table 5 for the display formats. The RDS key is operational in all modes except standby.

**Traffic**

Enable/disable traffic switching. When disabled, this is indicated by a decimal point in the 11th character of the dot matrix displays. Default at power-up is enabled. The TRAFFIC key works in all modes except standby. During manual mode and manual store mode, it toggles the TA inhibit status, which can subsequently be saved in NVM.

**Clear/Step**

Toggles between 10-kHz and 50-kHz steps on the FM band or between 1 and 9 kHz (or 10 kHz) on the MW band. There is no indication on the dot matrix displays. In manual mode, the displayed frequency is cleared to facilitate the entry of a new frequency. If the clear is followed by use of the + or – keys or the tuning knob, the original frequency is retained, allowing a change of step size only. In PS edit mode, the clear key clears the current PS name.

**TA Test**

Pressing TA test simulates the arrival of a group 14B. The PI code of the other network is embedded in the code (C5B1, Radio Clyde in the ROMed version).

**Time Colon**

This key enables or disables the flashing colon in the time display. This can be used to prevent unnecessary I/O activity thus reducing RFI. Disabling the colon prevents 1-Hz updating, as the display modules are only updated if the data to be displayed has changed.
This optional key selects 9- or 10-kHz steps on MW. Nine kHz is appropriate in Europe and 10 kHz in the United States. The default is 9 kHz, and the key need not be implemented if 10 kHz will never be required.

### Table 5. Key Function by Mode

<table>
<thead>
<tr>
<th>Mode</th>
<th>On/Off</th>
<th>Sleep</th>
<th>Alarm</th>
<th>+/-</th>
<th>Store</th>
<th>Manual</th>
<th>TP</th>
<th>RDS</th>
<th>0–9</th>
<th>Clear</th>
</tr>
</thead>
<tbody>
<tr>
<td>Standby</td>
<td>mode</td>
<td>mode</td>
<td>mode</td>
<td>--</td>
<td>--</td>
<td>--</td>
<td>--</td>
<td>--</td>
<td>--</td>
<td>--</td>
</tr>
<tr>
<td>(OFF)</td>
<td>normal</td>
<td>sleep</td>
<td>alarm</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Normal</td>
<td>mode</td>
<td>mode</td>
<td>+/- prog.</td>
<td>mode</td>
<td>manual</td>
<td>traffic</td>
<td>flag</td>
<td>display</td>
<td>tune</td>
<td>toggle</td>
</tr>
<tr>
<td>(ON)</td>
<td>standby</td>
<td>(ON)</td>
<td>PS-edit</td>
<td>PS-edit</td>
<td>manual</td>
<td>flag</td>
<td></td>
<td>display</td>
<td>prog.</td>
<td>10/50</td>
</tr>
</tbody>
</table>

### Legend

- **MW Step**: This optional key selects 9- or 10-kHz steps on MW. Nine kHz is appropriate in Europe and 10 kHz in the United States. The default is 9 kHz, and the key need not be implemented if 10 kHz will never be required.
Circuit

The circuit is in two distinct parts. The circuit for the MC145170 synthesizer is shown in Figure 6. The synthesizer board is the only part of the synthesizer controller which actually needs to be in (or close to) the radio. A local oscillator signal to supply the synthesizer should be taken from a low-impedance point so that the oscillator is not significantly loaded. Pulling of the oscillator frequency is not a problem as the PLL circuitry will compensate, but loading the tuned circuit itself is not recommended unless a high-impedance buffer is included. This prevents affecting the tuning range or the "Q" of the oscillator.

The MC145157 requires a divide-by-10 prescaler for FM and divide-by-5 for band 3. The MC145170 does not require prescaling. The standard LP1186 FM tuner does not have an LO take-off but a signal can be taken, without other modification, from the emitter of the oscillator BF195 (near the center of the PCB).

The Mullard LP1186 is unusual in having its local oscillator low. More recent tuners, for instance, the Larsholt 7254/55, almost always have their local oscillator above the tuned frequency. This selection can be made using port A, bit 2.

A 16-digit LCD (parallel) or VFD (serial) dot-matrix display module can be driven. The two display modules show the same data (within the limitations of their character ROMs). The VFD display driver supported is the MSC7128 and the LCD driver, the HD44780. On its own, this driver can be used to provide a 16-way multiplexed display, but an 8-way multiplexed higher contrast display is possible if the module also incorporates an HD44100. In an application which drives an LCD module (for instance, a ROMed PH8) and the module is not connected, a 10-k pullup resistor should be added to bit 7 of port A. This prevents the software hanging up waiting for the busy line to go low.

Figure 3 and Figure 4 show the circuit diagrams of the controllers. Figure 3 gives the pin numbers for the 52-pin PLCC HC11E with the numbers for the 56-pin SDIP (if different) in brackets. With the E32, the display in use can be selected by the level on port E, bit 7 (high for LCD and low for VFD) and the LCD multiplexing by port A, bit 7 (high for
divide-by-16, low for divide-by-8). The SW bank is selected by the level on port E, bit 6.

**Figure 4** shows pin numbers for the 84-pin PLCC K4, with the differences for the PH8 in brackets. Debug on the K4 using PCbug11 (reference 5) requires some additional hardware (within the dotted line) and port D bits 0 and 1 (SCI), port G bits 1 and 7 (XA14 and R/W), and port H bits 5 and 7 (CSGP1 and CSPROG), leaving 30 input/output (I/O) lines for use in the application. The display selections are not available on the PH8 ROMed versions, but there are four SW banks of 10 program memories; they are selected by port E, bits 6 and 7.

Since different demodulator devices can be used, the circuitry for the demodulator is not shown. The clock from the demodulator interrupts the microprocessor on each positive edge. At this time, a data bit is available and is read on bit 5 of port E.

### Software

An assembled listing of part of the HC11E32 ROMed version (ZC403311) of the application is included. The software is in three modules and was assembled and linked using the Introl re-locatable assembler and linker. The first module is listed. It contains all the main control routines, including the main loop and keyboard scanning, and the function to be performed by each key.

The second module contains the RDS and display functions, while the third module is the 4-function, 9-digit integer BCD arithmetic required for the MJD date calculations.

The second and third modules are described and listed in AN495 (reference 4). EB419/D (reference 5) describes and lists additional debug code contained in the ROMed parts.

The code which is executed only on startup (power-on or reset) begins at the label `START` on the third page of the first module’s listing, while the main loop starts at the label `IDLE` on the next page. The idle loop is quite long, as many functions and checks have to be carried out.
These include:

- Pacing the loop using the main timer
- Checking to see if the display needs updating or if a transient display has timed out
- Checking if alarm is armed and, if so, comparing its time with the current time
- Sleep timer operation
- Traffic announcement timing and return
- Keyboard scanning and selected function execution
- Incremental encoder execution
- Checking for changes in the band and memory selection inputs
- Timing band changes
- Updating TA = TP = 1 and M/S outputs

The keyboard subroutine (KBD) is executed at 64 Hz from the idle loop and checks to see if a key is pressed. If the same key is pressed on three consecutive tries, its function is performed. The remainder of the first module constitutes the subroutines performed by each key and the arithmetic and serial activity required to tune the synthesizers. The batch files used for linking the modules are shown as comments at the end of the listing, along with the pseudo-vectors required by PCbug11 during debug.

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; if there has been a change, the display update routine is executed. This is done to minimize 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 initialized, indicating no valid RDS data. The dot-matrix modules are then updated, if necessary, with new data. Each time, before anything 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
initialization 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. The LCD
and VFD routines are in the second software module (see reference 4).

<table>
<thead>
<tr>
<th>Display Mode</th>
<th>Format</th>
</tr>
</thead>
<tbody>
<tr>
<td>Standby Off</td>
<td>Alarm off</td>
</tr>
<tr>
<td></td>
<td>Alarm off, no CT</td>
</tr>
<tr>
<td></td>
<td>Alarm on</td>
</tr>
<tr>
<td>Normal On</td>
<td>With RDS PS name</td>
</tr>
<tr>
<td></td>
<td>Without RDS</td>
</tr>
<tr>
<td></td>
<td>Auto name</td>
</tr>
<tr>
<td></td>
<td>Tuning knob (AM)</td>
</tr>
<tr>
<td>Alarm</td>
<td>Alarm off</td>
</tr>
<tr>
<td></td>
<td>Alarm on/setup</td>
</tr>
<tr>
<td>Sleep</td>
<td></td>
</tr>
</tbody>
</table>

Table 6. Display Formats
Traffic Announcements

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 here. 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 rtrn/inhb message to:
  \textit{TA inhb: flag} — \textit{Traffic key inhibit flag (d.p. at the 11th character position)}

- Search for TA PI code in NVM; if found, proceed; otherwise, set TA rtrn/inhb message to:
  \textit{TA inhb: EON PI} — \textit{The PI code given in 14B is not in the NVM.}

- Check station TA inhibit flag in NVM; if clear, proceed; otherwise, set TA rtrn/inhb message to:
  \textit{TA inhb: NVM} — \textit{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 rtrn/inhb message to:
  \textit{TA rtrn: TP low} — \textit{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 rtrn/inhb message to:
  \textit{TA rtrn: PI code} — \textit{PI code of TP station was not as expected.}

- After an additional two seconds, start to monitor the TA flag; if high, remain on current frequency, if low, return to original frequency and set TA rtrn/inhb message to:
  \textit{TA rtrn: TA low} — \textit{TA flag of TP station low. This is the normal return method.}

- If, during a TA, the radio is manually retuned, the TA rtrn/inhb message is set to:
  \textit{TA rtrn: manual} — \textit{User-initiated manual return}
### Table 7. MCU I/O

<table>
<thead>
<tr>
<th>K4 and PH8</th>
<th>Function</th>
<th>E32</th>
</tr>
</thead>
<tbody>
<tr>
<td>Port A bits 0–7</td>
<td>LCD module data bus</td>
<td>Port C bits 0–7</td>
</tr>
<tr>
<td>Port B bits 0–7</td>
<td>High-order addresses (K4)</td>
<td>N/A</td>
</tr>
<tr>
<td>Port C bits 0–7</td>
<td>Data bus (K4)</td>
<td>N/A</td>
</tr>
<tr>
<td>Port D bits 0–1</td>
<td>Debug (PCbug11 or BUFFALO)</td>
<td>Port D bits 0–1</td>
</tr>
<tr>
<td>Port E bits 2–4</td>
<td>Keyboard rows (via 14028 encoder)</td>
<td>Port D bits 2–4</td>
</tr>
<tr>
<td>Port E bits 5</td>
<td>Standby (high:standby, low:on)</td>
<td>5</td>
</tr>
<tr>
<td>Port E bits 0–3</td>
<td>Keyboard columns</td>
<td>Port E bits 0–3</td>
</tr>
<tr>
<td>Port E bits 4</td>
<td>Shaft direction (XIRQ)</td>
<td>4</td>
</tr>
<tr>
<td>Port E bits 5</td>
<td>RDS data in or shaft direction (IRQ)</td>
<td>5</td>
</tr>
<tr>
<td>Port E bits 6</td>
<td>Short-wave memory select 1</td>
<td>6</td>
</tr>
<tr>
<td>Port E bits 7</td>
<td>Short-wave memory select 2 (PH8 only)</td>
<td>N/A</td>
</tr>
<tr>
<td>Port F bits 0–7</td>
<td>Low-order addresses</td>
<td>N/A</td>
</tr>
<tr>
<td>Port G bits 0</td>
<td>Mute</td>
<td>Port A bit 4</td>
</tr>
<tr>
<td>Port G bits 1</td>
<td>XA14 (K4 only)</td>
<td>N/A</td>
</tr>
<tr>
<td>Port G bits 2–4</td>
<td>LCD control lines (RS, R/W, and clock)</td>
<td>Port B bits 5–7</td>
</tr>
<tr>
<td>Port G bits 5–6</td>
<td>Band select</td>
<td>Port A bits 0–1</td>
</tr>
<tr>
<td>Port G bits 7</td>
<td>R/W (K4)</td>
<td>N/A</td>
</tr>
<tr>
<td>Port H bits 0–1</td>
<td>Serial clock/data for VFD and PLLs</td>
<td>Port B bits 0–1</td>
</tr>
<tr>
<td>Port H bits 2</td>
<td>VFD chip enable (PH8: +/- 10.7 MHz)</td>
<td>Port B bit 2</td>
</tr>
<tr>
<td>Port H bits 3</td>
<td>Port E, bit 5 input control</td>
<td>Port A bit 3</td>
</tr>
<tr>
<td>Port H bits 4</td>
<td>MC145170 PLL chip enable</td>
<td>Port B bit 4</td>
</tr>
<tr>
<td>Port H bits 5</td>
<td>CSGP1 (K4 only)</td>
<td>N/A</td>
</tr>
<tr>
<td>Port H bits 6</td>
<td>MC145157 PLL chip enable</td>
<td>Port B bit 3</td>
</tr>
<tr>
<td>Port H bits 7</td>
<td>CSPROG (K4 only)</td>
<td>N/A</td>
</tr>
<tr>
<td>N/A</td>
<td>FM IF select (+/- 10.7 MHz)</td>
<td>Port A bit 2</td>
</tr>
<tr>
<td>N/A</td>
<td>TA = TP = 1</td>
<td>Port A bit 5</td>
</tr>
<tr>
<td>N/A</td>
<td>M/S = 1</td>
<td>Port A bit 6</td>
</tr>
<tr>
<td>N/A</td>
<td>LCD multiplex select (8/16)</td>
<td>Port A bit 7</td>
</tr>
<tr>
<td>N/A</td>
<td>Display module (LCD/VFD) select</td>
<td>Port E bit 7</td>
</tr>
</tbody>
</table>
Setup and Testing

An effective method of fault finding a PLL circuit is to initially do the tuning with a potentiometer, leaving the output of the filter disconnected from the VCO. As the radio is tuned through the frequency setup in the synthesizer, the filter output should switch from one extreme to the other. Until this test passes, it is not useful to close the loop, as it is difficult to distinguish the cause of a problem from its effects.

Check operation of the MC34064 LVI circuit. As the supply voltage is lowered, it should pull the reset pin low. This should occur between 4.70 and 4.50 volts. Adjust trimmer on the EXTAL pin of the M68HC711 for accurate timekeeping in the absence of RDS CT information. (Radio should be detuned or tuned to a station known not to provide RDS.) The trimmer on pin 2 of the PLL chip (MC145157 or MC145170) should be adjusted to provide an accurate reference frequency. This adjustment can be made simply to tuning to a strong broadcast of known frequency and adjusting for optimum reception or symmetric adjacent-channel response.

PH8 ROMed Application

The ROMed PH8s (ZC428200 and ZC428202) differ from the described E32 version of this application as follows:

1. 40 short-wave programs can be stored instead of 20. These are accessed by the use of a second memory-select line (port E, bit 7).

2. There is no display selection; both LCD and VFD signals are generated. If an LCD module is not connected, a pulldown on port A, bit 7 should be included (see Figure 4).

3. LCD multiplexing is fixed at divide-by-8.

4. Traffic announcement (retune to TA frequency) is not fully implemented in the ZC428200.
5. Time colon FLASH defeat key is not implemented; the display modules are always updated at 8 Hz.

6. TA = TP = 1 and M/S outputs are not implemented.

7. 10-kHz MW steps are not available (no 9/10 key).

8. +/-10.7-MHz IF selection (FM) is carried out on port H, bit 2 which is read after reset but before it is set up as an output. A pullup or pulldown resistor will determine the IF selection (pullup for LO high and pulldown for LO low) without affecting the pin’s subsequent function as an output (VFD chip enable).

9. The 500-ms delay at switch-on between the standby line moving and the PLLs being retuned is not implemented.

10. The sleep d.p. flashes during operation of the sleep timer.

References

1. A Radio Synthesizer Using the MC68HC05B4, Motorola document order number ANE416/D

2. CENELEC EN 50067, Specifications of the Radio Data System (RDS), formerly EBU technical document 3244

3. An RDS Decoder Using the MC68HC05E0, Motorola document order number AN460/D

4. RDS Decoding for an HC11 Controlled Radio, Motorola document order number AN495/D

5. ROMed HC11E32 and HC11PH8 Including BUFFALO Monitor and PCbug 11 Talker, Motorola document order number EB419/D
**Code Listing**

```
IMPORT  SDATA, TINTB, INITD, MOD, CLOCK, MJDAT, WAIT, CLRCON, CBCD, PROC, TFCC
EXPORT  DCON2, NEW, CLR, SHAFT
LIB     RDRAME.S11

PORTA  EQU $00 PORT A ADDRESS
PORTB  EQU $04 "    B  
PORTC  EQU $03 "    C  
PORTD  EQU $08 "    D  
PORTE  EQU $0A "    E  
PORTCD EQU $07 PORT C DATA DIRECTION REG.
PORTDD EQU $09 "    D  
TMSK2  EQU $24
PACTL  EQU $26
OPTION EQU $39
INIT   EQU $3D
RBO    EQU $1000 REGISTER BLOCK OFFSET
PPROG  EQU $3B EEPROM CONTROL REGISTER
ND     EQU 9 No. DIGITS

SECTION.S .RAM1, COMM
BMJD    RMB 3 BINARY MJD
Q       RMB 9 WORKING NUMBER 1 - RDS
TMQ     RMB 9 SCRATCH
R       RMB 9 WORKING NUMBER 2 - RDS
TMP     RMB 9 MULT. OVER. OR DIV. REMAINDER
R       RMB 9 WORKING NUMBER 3 - RDS
MJD     RMB 9 MODIFIED JULIAN DAY NUMBER
YR      RMB 9 YEAR
MNTH    RMB 2 MONTH
DOM     RMB 2 DATE
DOW     RMB 1 DAY OF WEEK

DIST    RMB 1 TRANSIENT DISPLAY, TIMEOUT, COUNTER
SLEEP   RMB 1 SLEEP TIMER MINUTES COUNTER
RSTOT   RMB 1 RDS TIMEOUT COUNTER
PSNP    RMB 1 PS DISPLAY POINTER
DAT     RMB 4 SERIAL DATA BUFFER
TMGRP   RMB 8 TEMPORARY GROUP DATA
GROUP   RMB 8 COMPLETE GROUP DATA
PTY     RMB 1 PROGRAM-TYPE CODE (CURRENT)
PTYCMP  RMB 1 PROGRAM TYPE CODE (PTY SCAN)
PI      RMB 2 PROGRAM IDENTIFICATION CODE
PION    RMB 2 PROGRAM IDENTIFICATION CODE (EON)
PIN     RMB 2 PROGRAM ITEM NUMBER
LEV     RMB 1 VALID BLOCK LEVEL
BIT     RMB 1 BIT LEVEL
ITMP1   RMB 1 TEMP BYTE FOR USE IN IRQ
SYN     RMB 2 SYNDROME

* RAM allocation, RDS & radio. *
* *

* *
```

---

**AN494**

30

MOTOROLA
CONF RMB 1 SYNDROME CONFIDENCE
TH32 RMB 1 TICS (SECONDS/32)
TH8 RMB 1 EIGHTHS OF SECONDS
SEC RMB 1 SECONDS
MIN RMB 1 MINUTES
OUR RMB 1 HOURS
AMIN RMB 1 ALARM MINUTES
AOUR RMB 1 ALARM HOURS
DISP1 RMB 1 RT DISPLAY POINTER #1
DISP2 RMB 1 RT DISPLAY POINTER #2
RQ RMB 6 WORKING BCD NUMBER 1 RADIO
RP RMB 6 * * * 2 *
RR RMB 2 * * 3 *
W1 RMB 2 W
W2 RMB 2 O
W3 RMB 2 R
W4 RMB 2 K
W5 RMB 2 I
W6 RMB 2 N
W7 RMB 2 G
KEY RMB 1 CODE OF PRESSED KEY
KOUNT RMB 1 KEYBOARD COUNTER
DIG2 RMB 1 2nd DIGIT TIMEOUT COUNTER
CARRY RMB 1 BCD CARRY
COUNT RMB 1 LOOP COUNTER
NUM1 RMB 2 1ST No. POINTER (ADD & SUBTRACT)
NUM2 RMB 2 2ND No. POINTER (ADD & SUBTRACT)
LED RMB 1 STATION NUMBER
SMEM RMB 2 CURRENT FREQUENCY
REARET RMB 1 LAST TA REASON FOR RETURN
RTDIS RMB 1 RDS DISPLAY TYPE
DI RMB 1 DECODER IDENTIFICATION
SCHAN RMB 1 SCAN CHANNEL

* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
* * * * Flags, & pages 1-2. * * * * * * * * *
* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *

STAT RMB 1
* 0: MODE 1: STATION, 0: FREQ
* 1: STEP 1: 50kHz, 0: 10kHz
* 2: CLRQ 1: CLEAR IF NO. KEYED
* 3: TIMER MS BIT TOGGLE (64 Hz)
* 4: RDS DATA CLEARING ENABLE
* 5: KEY FUNCTION PERFORMED
* 6: KEY REPEATING
* 7: NOT JUST POWERED UP

STAT2 RMB 1
* 0: VALID SYNDROME
* 1: VALID GROUP
* 2: RT DISPLAY
* 3: UPDATE DISPLAY
* 4: CLEAR DISPLAY
* 5: SPACE FLAG
* 6: NOT ON PROGRAM (AM)

STAT3 RMB 1
* 0: NOT ON PROGRAM (FM)
* 1: TEXTA/TEXTB BIT (RT)
* 2: TA FLAG
* 3: TP FLAG
* 4: SHAFT DIRECTION
* 5: SHAFT ROTATION
* 6: UPDATE DATE

STAT4 RMB 1
* 7: SHAFT INTERRUPTS
* 0: DISPLAY (OR TA SWITCH) TRANSIENT
* 1: SLEEP TIMER RUNNING
* 2: TRAFFIC ENABLED
* 3: ALARM DISPLAY
* 4: ALARM ARMED
* 5: ALARM SET-UP
* 6: ALARM HOURS (SET-UP)

STAT5 RMB 1
* 7: VALID GROUP 14B RECEIVED
* 0: BAND CHANGE TIMEOUT
* 1: RDS DISPLAYS
* 2: SLEEP DISPLAY
* 3: M/S 0: M, 1: S
Application Note

* 4: RETUNE FLAG (FREQUENCY MODE)
* 5: TA INHIBIT FLAG (NVM)
* 6: STORE MODE
* 7: WEEKDAY ONLY ALARM

STAT6 RMB 1 BAND/BANK (,,MW STEP, COLON, , A1,A0,,E6)
BCTO RMB 1 BAND CHANGE TIMEOUT
SCNT RMB 1 SHAFT DETENT COUNTER

SECTION .RAM2, COMM
EON RMB 256
SECTION .RAM3, COMM
DISP RMB 16 LCD MODULE BUFFER
DISPP RMB 16 CURRENT LCD MODULE CONTENTS
PSN RMB 8
RT RMB 69 RADITEXT
SECTION .ROM1

STRST JMP START RESET VECTOR
TMRB JMP TINTB RTI
IRQ JMP SDATA IRQ

********************************************
** Reset routine - set-up ports etc. **
** ********************************************

START LDAA #$01
STAA INIT REGISTERS AT $1000
LDAA #$10 ENABLE EEPROM WRITE (NOT CONFIG)
STAA $1035
LDAA #$30 IRQ EDGE SENSITIVE
STAA $1039
LDAA #$03 32Hz RTI (8.388MHz XTAL)
STAA $1026 PORTA, BITS 3 & 7 INPUTS
LDAA #$80 ENABLE REAL TIME INTERRUPTS
STAA $1024
LDAA #$00 DWOM = 0, PORTD PUSH-PULL
STAA $1028

LDS #$02FF INITIALISE STACK POINTER
LDY #$1000 0,1: BAND INPUTS (FM, FM, MW, SW), 2: FM IF
LDAA #$10 3: IRQ CONTROL, 4: MUTE, 5: TA=TP=1
STAA PORTA,Y 6: M/S=1, 7: 8/16 LCD MUX
LDAA #$00 0,1: SERIAL CLOCK/DATA, 5,6,7: LCD CONTROL
STAA PORTB,Y 2,3,4: LATCH SIGNALS (VFD, 5157 & 5170)

H2L LDAA #$80 0,1: SERIAL CLOCK/DATA, 5,6,7: LCD CONTROL
STAA PORTB,Y 2,3,4: LATCH SIGNALS (VFD, 5157 & 5170)
CLR PORTC,Y
LDAA #$0FF 0-7: LCD PARALLEL BUS
STAA PORTCD,Y
CLR PORTD,Y
LDAA #$3C 2-4: KEYBOARD OUTPUTS
STAA PORTDD,Y 5: STANDBY

* PORTE PORTE 0-3: KEYBOARD INPUTS, 4: SHAFT INPUT (XIRQ)
* * PORTE 5: RGB/SHAFT INPUT, 6: SW BANK, 7: LCD/VFD

********************************************
** INITIALISE LCD AND RAM. **
** ********************************************

JSR DBOUNC WAIT 15ms
LDAA #$30
JSR CLOCK INITIALISE LCD
JSR DBOUNC WAIT 15ms
LDAA #$30
JSR CLOCK INITIALISE LCD
LDX #BMJD INITIALISE PAGE 0 RAM

CLOOP CLR 0,X
INX
CPX #SCNT+1
BNE CLOOP
BSET STAT4, #04
BSET STAT, $01
LDAA #$30
JSR CLOCK INITIALISE LCD
JSR WAIT
LDAA #$30 /8 DISPLAY
BRCLR PORTA, Y, $80, M8
LDAA #$38 /16 DISPLAY
M8 JSR CLOCK LATCH IT
JSR WAIT
LDAA #$08 SWITCH DISPLAY OFF
JSR CLOCK LATCH IT
JSR WAIT
LDAA #$01 CLEAR DISPLAY
JSR CLOCK LATCH IT
JSR INITS INITD INITIALISE RDS DATA & DISPLAY
JSR CLREON AND EON DATA

* Initialise interrupt JMPs

JRT1 EQU $00EB E32 BUFFALO RAM JUMP TABLE
JIRQ EQU $00EE "    "    "    "    
JXIRQ EQU $00F1 "    "    "    "    
LDAA #$7E
STAA JRTI
STAA JIRQ
STAA JXIRQ
LD #TINTB
STD JRTI+1 RTI
LD #SDATA
STD JIRQ+1 IRQ
LD #SHAFTX
STD JXIRQ+1 XIRQ
LDAA #$00 ENABLE IRQ & XIRQ
TAP

******************************************************************************
**
** Idle loop. 
**
******************************************************************************

IDLE
LDY #$1000
BRSET STAT, #08, TBH
BRSET $0E, Y, $80, *
BSET STAT, #08
BRA NO2D
TBH
BRCLR $0E, Y, $20, *
BCLR STAT, #80
BNOTSNZ BRSET STAT4, $10, NNT2 STANDBY, ALARM ARMED?
LDAA DOW YES, TIMED OUT?
JSR CLR

NOSNZ BRCLR STAT4, $01, NOPS DISPLAY TRANSIENT?
LDAA DIST
BNE NOPS YES, TIMED OUT?
JSR CLR

NOP5 BRCLR STAT2, #08, NDU DISPLAY UPDATE REQUIRED?
JSR MOD YES, DO IT
BCLR STAT2, #08 AND CLEAR FLAG

NDU BRCLR PORTD, Y, $20, FULON STANDBY?

NOTSNZ BRSET STAT4, #10, NNT2 STANDBY, ALARM ARMED?
NTJ2 JMP NT2

NNT2 BRCLR STAT5, #80, NWA YES, WEEKDAY ALARM ONLY?
LDAA DOW YES
CMPA #4 SATURDAY OR
BHI NT2J SUNDAY?

NWA LDAA AOUR NO, COMPARE ALARM HOURS
CMPA OUR WITH TIME
Application Note

BNE  NT2J  SAME?
LDAA  AMIN  YES, COMPARE ALARM MINUTES
CMPA  MIN  WITH TIME
BNE  NT2J  SAME?
LDAA  SEC  ONLY ALLOW WAKE-UP IN FIRST SECOND
BNE  NT2  TO PREVENT SWITCH-OFF LOCKOUT

ONAG  BCLR  PORTD,Y,$20  YES, SWITCH ON,
Jsr  DEL500  WAIT 500ms,
BCLR  PORTA,Y,$10  DEMUTE
Jsr  P5170  AND TUNE (5170 & 5157)

FULON  BCLR  STAT4,$02,FLN  SLEEP TIMER RUNNING?
LDAA  SLEPT  YES
BNE  FLN  TIME TO FINISH?
BCLR  STAT4,$02  YES, CLEAR FLAG,
BSET  PORTA,Y,$10  SWITCH OFF
BSET  PORTA,Y,$10  AND MUTE

*******************************************************************************
**                           Idle loop (cont.).                              **
*******************************************************************************

FLN  BCLR  STAT4,$80,NT1  14B FLAG HIGH?
BRSET  STAT2,$80,NT2  YES, BIT AGREES?
BSET  STAT2,$80  NO, SET BIT
CLR  REARET
LDAA  #25  LOCK OUT RETURN
STAA  DIST  FOR 3 SECONDS
BSET  STAT4,$01  SET DISPLAY TRANSIENT FLAG
BSET  PORTA,Y,$10  MUTE
Jsr  DBNC  WAIT 150 ms
Jsr  RETUNE2  AND RETUNE
BCLR  STAT4,$80,NWWS  PI CODE NOT IN EON LIST?
Jsr  DEL500  WAIT 500ms
*  BCLR  PORTA,Y,$10,SOK  SIGNAL OK?
*  LDAA  #2
*  STAA  REARET
*  BRA  NT1
SOK  Jsr  DEL500  WAIT 500ms
BRSET  STAT3,$08,TPOK  TP OK?
LDAA  #5
STAA  REARET
BRA  NT1
TPOK  Jsr  P1  YES, CHECK PI CODE
CMFA  P1ON
BNE  PINOK1
LDAA  P1+1  AGAINST PI (EON)
BRA  NT2  IF OK STAY SWITCHED
PINOK1  LDAA  #3
STAA  REARET

NT1  BCLR  STAT2,$80,NT2  14B FLAG LOW, BIT AGREES?
BCLR  STAT4,$80  MAKE SURE 14B CANCELLED
BSET  PORTA,Y,$10  MUTE
Jsr  DBNC  WAIT 150 ms
NWWS  BCLR  STAT2,$80  CLEAR FLAG
LDAA  LED  SELECTED PROGRAM
Jsr  RETUNE2  AND RETURN TO ORIGINAL PROGRAM

NT2  Jsr  KBD  READ KEYBOARD
Jsr  KEYP  EXECUTE KEY
BCLR  STAT3,$20,NSRO  SHAFT ROTATION PENDING?
BCLR  STAT3,$20  YES, CLEAR FLAG
BRSET  STAT3,$10,ANTI  DIRECTION?
Jsr  PINC2  CLOCKWISE, INCREMENT
BRA  NSRO
ANTI  Jsr  PDEC2  ANTI-CLOCKWISE, DECREMENT
NSRO  BCLR  STAT3,$40,NRDSP  UPDATE DATE?
Jsr  MJDAT  YES, CONVERT FROM MJD

*******************************************************************************
**                           Idle loop (cont.).                              **
*******************************************************************************
* Retune if band or SW bank inputs changed. *

```
NRDSP
LDY #$1000
BRCLR STAT,$80,BTO
BRCLR PORTA,Y,$01,L5
BRSET STAT6,$04,CG6
BSET STAT6,$004
BSET STAT6,$008,BTO
BSET STAT3,$80
BRA CHE

L5
BRCLR STAT6,$04,CG6
BCLR PORTA,Y,$01,L5
BRSET STAT6,$004,BTO
BCLR STAT3,$80
BRA CHE

CG6
BRCLR PORTA,Y,$02,L6
BRSET STAT6,$008,CHE
BSET STAT6,$008
BRA CHE

L6
BRCLR STAT6,$008,CHE
BCLR PORTA,Y,$02,L6
BRSET STAT6,$008
BCLR STAT3,$80
BRA CHE

CHE
BRSET STAT6,$0C,BD3
BRA OK6

BD3
BRCLR PORTE,Y,$40,E6L
BSET STAT3,$80
BRA CHE

E6L
BRCLR STAT6,$001,OK6
BCLR STAT6,$001
BRA CHE

BTO
BSET STAT,$80
STAA BCTO
BSET STAT5,$01

OK6
BRCLR STAT3,$01,AOK
BDEC BCTO
BNE ARI
BCLR STAT3,$01
BSET STAT3,$001
BSET STAT6,$001
BSET E6L
RCLP

ARI
BRSET STAT3,$001,AOK
BSET PORTA,Y,$50
BRA IOK

TATP
BCLR PORTA,Y,$50
BSET STAT3,$008
BSET PORTA,Y,$50
BRA IDLJ

IOKK
BSET PORTA,Y,$50
BSET MSH
BSET PORTA,Y,$50
BRA IDLJ

MSH
BCLR PORTA,Y,$50
BSET PORTA,Y,$50
BRA IDLJ

IDLJ
JMP IDLE

RCLP
BSET PORTA,Y,$50
LDAB #120
JSR READ1
STAA LED
```

---

AN494

MOTOROLA

35
JMP RETUNE2 PROGRAM 145170/57

**********************************************************************
* Shaft rotation interrupts. *
**********************************************************************

SHAFT BRSET PORTE,Y,$20,SEM IRQ, SHAFT I/O HIGH (E5) ?
BCLR STAT3,$10 NO, CLEAR DIRECTION BIT
BRA TEM
SEM BSET STAT3,$10 YES, SET DIRECTION BIT
TEM BSET STAT3,$20 SET FLAG TO INDICATE ROTATION
RTI

SHAFTX BRSET PORTE,Y,$10,XEM XIRQ, SHAFT I/O HIGH (E4) ?
BCLR STAT3,$10 NO, CLEAR DIRECTION BIT
BRA YEM
XEM BSET STAT3,$10 YES, SET DIRECTION BIT
YEM BSET STAT3,$20 SET FLAG TO INDICATE ROTATION
RTI

**********************************************************************
* Keyboard routine. *
**********************************************************************

KBD CLR W1
LDY #$1000
LDX #$7
KEY1 LDAB W1
AODB #$04 SELECT COLUMN
STAB W1
LDAB PORTD,Y PRESERVE OTHER PORTD DATA
ANDB #$20
AODB W1
STAB PORTD,Y READ KEYBOARD
BITA #$0F ANY INPUT LINE HIGH ?
BNE L1
DEX NO, TRY NEXT COLUMN
BNE KEY1 LAST COLUMN ?
CLR KEY YES, NO KEY PRERESSED
BRA EXIT
L1 LDAB W1
LSLB
LSLB
LDAA PORTE,Y READ KEYBOARD
ANDA #$0F
ABA
CMPA KEY SAME AS LAST TIME ?
BEQ EXIT
STAA KEY NO, SAVE THIS KEY
CLR KOUNT
EXIT INC KOUNT YES, THE SAME
LDAA KOUNT
BRCLR STAT,$40,NRML REPEATING ?
LDAB PSNP YES
BEQ NOTCH CHARACTER CHANGE ?
CMPA #$8 YES, REPEAT AT 8 Hz
BRA GON2
NOTCH CMPA #$47 TIME TO DO SOMETHING ?
LDAA GOON2 TIME TO DO SOMETHING ?
BRA GON2
GOON2 CMPA #47 KEY PRESSED ?
BHI GON2
LDAA GOON2 TIME TO DO SOMETHING ?
BRA GOON3
GOON3 CMPA #$54 DEC. PROG.
BEQ GOON2

KBD CLR W1
LDY #$1000
LDX #$7
KEY1 LDAB W1
AODB #$04 SELECT COLUMN
STAB W1
LDAB PORTD,Y PRESERVE OTHER PORTD DATA
ANDB #$20
AODB W1
STAB PORTD,Y READ KEYBOARD
BITA #$0F ANY INPUT LINE HIGH ?
BNE L1
DEX NO, TRY NEXT COLUMN
BNE KEY1 LAST COLUMN ?
CLR KEY YES, NO KEY PRERESSED
BRA EXIT
L1 LDAB W1
LSLB
LSLB
LDAA PORTE,Y READ KEYBOARD
ANDA #$0F
ABA
CMPA KEY SAME AS LAST TIME ?
BEQ EXIT
STAA KEY NO, SAVE THIS KEY
CLR KOUNT
EXIT INC KOUNT YES, THE SAME
LDAA KOUNT
BRCLR STAT,$40,NRML REPEATING ?
LDAB PSNP YES
BEQ NOTCH CHARACTER CHANGE ?
CMPA #$8 YES, REPEAT AT 8 Hz
BRA GON2
NOTCH CMPA #$47 TIME TO DO SOMETHING ?
LDAA GOON2 TIME TO DO SOMETHING ?
BRA GON2
GOON2 CMPA #$54 DEC. PROG.
BEQ GOON2

KBD CLR W1
LDY #$1000
LDX #$7
KEY1 LDAB W1
AODB #$04 SELECT COLUMN
STAB W1
LDAB PORTD,Y PRESERVE OTHER PORTD DATA
ANDB #$20
AODB W1
STAB PORTD,Y READ KEYBOARD
BITA #$0F ANY INPUT LINE HIGH ?
BNE L1
DEX NO, TRY NEXT COLUMN
BNE KEY1 LAST COLUMN ?
CLR KEY YES, NO KEY PRERESSED
BRA EXIT
L1 LDAB W1
LSLB
LSLB
LDAA PORTE,Y READ KEYBOARD
ANDA #$0F
ABA
CMPA KEY SAME AS LAST TIME ?
BEQ EXIT
STAA KEY NO, SAVE THIS KEY
CLR KOUNT
EXIT INC KOUNT YES, THE SAME
LDAA KOUNT
BRCLR STAT,$40,NRML REPEATING ?
LDAB PSNP YES
BEQ NOTCH CHARACTER CHANGE ?
CMPA #$8 YES, REPEAT AT 8 Hz
BRA GON2
NOTCH CMPA #$47 TIME TO DO SOMETHING ?
LDAA GOON2 TIME TO DO SOMETHING ?
BRA GON2
GOON2 CMPA #$54 DEC. PROG.
AN494

MOTOROLA

37

CMPA #$58          INC.PROG.
BEQ GOON3
CMPA #$52          SLEEP
BNE DNT2           IF NOT A REPEAT KEY, DO NOTHING
GOON3 BSET STAT,$40 SET REPEAT FLAG
CLR KOUNT
GOON LODA KEY
BEQ RKEY           SOMETHING TO DO ?
SEC YES, SET C
RTS
RKEY BCLR STAT,$20 NO, CLEAR DONE FLAG
DNT2 BCLR STAT,$40 CLEAR REPEAT FLAG
CLR KOUNT CLEAR COUNTER
KCLC CLC
DNT RTS

********************************
** Execute key. **
********************************

KEYP BCC DNT          ANYTHING TO DO ?
KEYP2 LODA KEY         YES, GET KEY
CMPA #$54             DEC. PROG. (M)
BEQ RPT
CMPA #$58             INC. PROG. (S)
BEQ RPT
CMPA #$52             SLEEP
BEQ RPT
BRSET STAT,$20,DNT    NOT A REPEAT KEY, FLAG SET ?
RPT CLRB
RJ LDX #CTAB
ABX
LODA 0,X              FETCH KEYCODE
CMPA KEY             THIS ONE ?
BEQ PJ                YES
CMPA LAST             NO, LAST CHANCE ?
BEQ DNT               YES, ABORT
ADD #4                NO TRY THE NEXT KEY
BRA RJ
PJ BSET STAT,$20
JSR 1,X
JMP P5170

********************************
** Keyboard jump table. **
********************************

CTAB FCB $11 0
     FCB $21 1
     FCB $22 2
     FCB $24 3
     FCB $31 4
     FCB $32 5
     FCB $34 6
     FCB $41 7
     FCB $42 8
     FCB $44 9
     FCB $48 ALARM
     FCB $50 ALARM
     FCB $58 STORE/SET
     FCB $59 SAVE

Application Note
Code Listing
FCB $18 ON/OFF
JMP ONOFF
FCB $14 CLEAR/STEP
JMP CLEAR
FCB $12 MODE (PROG./FREQ.)
JMP MODE
FCB $52 SLEEP TIMER START
JMP SLEEP
FCB $54 DEC. PROG./FREQ./CHAR.
JMP PDEC
FCB $58 INC. PROG./FREQ./CHAR.
JMP PINC
FCB $61 RDS DISPLAYS
JMP RTDSP
FCB $62 TRAFFIC ENABLE (TOGGLE)
JMP TPEN
FCB $64 MW STEP 9/10kHz (TOGGLE)
JMP T910
FCB $51 COLON CONTROL
JMP TFCC
LAST FCB $68 TA TEST
JMP TEST

******************************************************************************
* Alarm key. *
******************************************************************************

ALARM BRCLR STAT4, $08, ADON ALARM DISPLAY ON ?
BRCLR STAT4, $10, ALOF YES, ALARM ON ?
BCLR STAT4, $10 YES, SWITCH OFF
BRA UDNT
ALOF BSET STAT4, $10 NO, SWITCH ON
BRA UDNT

ADON JSR CLR NO, ENABLE ALARM DISPLAY
BSET STAT4, $08 ALARM DISPLAY FLAG
UDCNT
BCLR STAT4, $20 CANCEL SET-UP
LDAA #$25 3 SECONDS TIMEOUT
STAA DIST
BSET STAT4, $01 SET DISPLAY TRANSIENT FLAG
ABOA RTS

******************************************************************************
* On/off key. *
******************************************************************************

ONOFF JSR CLR CLEAR DISPLAY TRANSIENTS
BCLR STAT4, $82 CANCELLED SLEEP TIMER & TA SWITCH FLAG
BCLR STAT4, $40 CANCEL STORE MODE
BRCR PORTD, Y, $20, ALRON ON ?
SODM BCLR PORTD, Y, $20 NO, SWITCH ON
JSR DEL500 WAIT 500ms
BCLR PORTA, Y, $10 AND DEMUTE
RTS
ALRON BSET PORTD, Y, $20 YES, SWITCH OFF
BSET PORTA, Y, $10 AND MUTE
RTS

******************************************************************************
* PS name clear. *
******************************************************************************
PSC LDX #PSN
LOAD #EFF
CPSL STAA 0, X
INX
CPX #PSN+8
BNE CPSL
RTS

AN494
* * *

** TP. **
* * *

---

** TPEN **

BRSET PORTD,Y,$20,HIGH STANDBY ?
BRSET STAT,$01,NS1 NO, NORMAL MODE ?
BRSET STATS,$20,TAEH NO, FREQ. MODE, NVM DISABLE FLAG SET ?
BSET STAT,$20 NO, SET IT
RTS

TAEH

BCLR STAT,$20 YES, CLEAR IT

NS1

BCLR STAT4,$04,TPOF NORMAL MODE, TRAFFIC ON ?
BCLR STAT4,$04 YES, DISABLE
RTS

TPOF

BSET STAT4,$04 NO, ENABLE

---

** Sleep timer. **

---

** SLEEP **

BRSET STAT4,$04,DECS ALREADY SLEEP DISPLAY ?
BSET STAT,$02,STR NO, SLEEP TIMER ALREADY RUNNING ?
INSLP

LDAA #60 NO, INITIALISE SLEEP TIMER
SLEEP

STAA SLEPT
BSET STAT4,$02 START SLEEP TIMER
STR

JSR CLTR YES, CLEAR DISPLAY TRANSIENTS
BSET STAT,$04 SLEEP DISPLAY
BRA SLPTOK NO DECREMENT IF FIRST TIME

DECS

LDAA #60 DECIMATE SLEEP TIMER
SUBA #5
STAA SLEPT
BMI INSLP

SLPTOK

LDAA #25
STAA DIST
BSET STAT4,$01 START DISPLAY TRANSIENT
BRSET PORTD,Y,$20,SODM ALREADY ON ?
BCLR PORTA,Y,$10 YES, JUST DEMUTE
RTS

---

** Number entry routine. **

---

** DIGIT **

BRSET PORTD,Y,$20,AB03 STANDBY ?
JSR CLTR NO, CLEAR DISPLAY TRANSIENTS
LSRB
LSRB

BRSET STAT,$01,SKP STATION MODE ?
BSET STAT,$40,SKP NO, STORE MODE ?
BSET STAT,$10 NO, SET RETURN FLAG (FREQUENCY MODE)
BLCR STAT,$02 AND CLEAR TA INHIBIT BIT (NVM)
STAB W3
BRC LR STAT,$04,SHIFT CLEAR Q ?
BCLR STAT,$04 YES, CLEAR FLAG
JSR CLQ AND CLEAR Q
SHIFT

BSR DR1 W1: MSD, W2: LSD
LOX W1

AGS

LDAA 1,X MOVE ALL DIGITS
STAA 1,X UP ONE PLACE
INX

CPE W2 DONE ?
LDAA W3 YES, RECOVER NEW DIGIT
STAA 0,X AND PUT IT IN LSD
RTS
**Application Note**

```
SKP       BSET          PORTA,Y,$10      MUTE
TBA       STAA          LED
JMP       RETUNE

********************************
** Save pointers & 500ms delay. **
********************************

DR1      LDX           #RQ           STORE POINTERS
STX       W1
LDAB       #5
ABX       W2
ABO3       RTS

DELS00    LDX           #255
JSR        SKDB
LDX        #255
JMP        SKDB

********************************
** Increment key (& knob). **
********************************

PINC2     BRSET         STAT4,$20,ALSU1  ALARM SET-UP ?
BRSET      STAT4,$08,TG57    NO, ALARM DISPLAY ?
BRSET      PORTD,Y,$20,DMI   NO,STANDBY ?
LDAB       PSNP
BNE        PSNO
JMP         UP
            NO, STEP UP

PINC      BRSET         STAT4,$20,ALSU1  ALARM SET-UP ?
TOG57J     BRSET         STAT4,$08,TG57    NO, ALARM DISPLAY ?
BRSET      PORTD,Y,$20,DMI   NO, STANDBY ?
BRSET      STAT,$01,NACS    NO, FREQ. MODE ?
JMP         UP
            YES, STEP UP

********************************
** Alarm inc. (hours/minutes). **
********************************

ALSU1     BRSET         STAT4,$40,IHR    YES, SET-UP HOURS ?
LDAA       AMIN
            NO, MINUTES
CMPA       #59
BHS        TOOH
INC        AMIN
BRA        TSS
TOOH       CLR        AMIN
BRA        TSS
IHR        LDAA       AOURL
CMPA       #23
BLO        HTOH
CLR        AOURL
BRA        TSS
HTOH       INC        AOURL
TSS        LDAA       #80
            10 SECOND TIMEOUT
STAA       DIST
BSET       STAT4,$01
BCLR       PORTA,Y,$10
            SET DISPLAY TRANSIENT FLAG
DMI        RTS

NACS       LDAB       PSNP
BEQ        CONTI
            NO, PS EDIT MODE ?
```
**P-S Edit inc. (ASCII) and 5/7 day toggle**

```
PSN0  LDX   #PSN-1
   ABX
   LDAA  0,X
   INCA  INCREMENT ASCII VALUE
   CMPA  #$20 SPACE
   BLS   MAK20 LESS OR EQUAL ?
   CMPA  #$2E NO, .
   BLS   MAK2E LESS OR EQUAL ?
   CMPA  #$30 NO, 0
   BLO   MAK30 LESS ?
   CMPA  #$39 NO, 9
   BLS   CNTB LESS OR EQUAL ?
   CMPA  #$41 NO, A
   BLO   MAK41 LESS ?
   CMPA  #$5A NO, Z
   BLS   CNTB LESS OR EQUAL ?
   CMPA  #$61 NO, a
   BLO   MAK61 LESS ?
   CMPA  #$7A NO, z
   BLS   CNTB LESS OR EQUAL ?

MAK20  LDAA  #$20 MAKE SPACE
   BRA   CNTB
MAK2E  LDAA  #$2E MAKE .
   BRA   CNTB
MAK30  LDAA  #$30 MAKE 0
   BRA   CNTB
MAK41  LDAA  #$41 MAKE A
   BRA   CNTB
MAK61  LDAA  #$61 MAKE a
   CNTB
   STAA  0,X
   LDAA  #80
   JMP   OUTCH

TOG57  BRCLR  STAT4,$10,DMI ALARM ARMED ?
   BRCLR  STAT5,$80,A7 YES, 7-DAY ALARM ?
   BCLR  STAT5,$80 NO, MAKE IT 7 DAY
   BRA   T5S
   A7  BSET  STAT5,$80 YES, MAKE IT 5 DAY
   BRA   T5S

CONTI  BSET   PORTA,Y,$10 MUTE
   BSET   STAT2,$08 PROG. No. INCREMENT, UPDATE DISPLAY
   LDAA  LED
   BRSET  STAT2,$80,IOK IF SWITCHED TO TA DON’T INCREMENT
   INCA  NEXT PROG.
   CMPA  #9 TOO HIGH ?
   BLS   IOK
   CLRRA YES, BACK TO ZERO
   IOK  STAA  LED
   JMP   RETUNE

IOK
```

**Decrement key (& knob).**

```
PDEC2  BRSET  STAT4,$20,ALSU2 ALARM SET-UP ?
   BRSET  STAT4,$08,T0G57 NO, ALARM DISPLAY ?
   BRSET  PORTD,Y,$20,DMD NO, STANDBY ?
   LDAB  PSNP
   BNE  PSN1 NO, PS EDIT MODE ?
   JMP   DOWN NO, STEP DOWN
```
PDEC BRSET STAT4,$20,ALSU2 ALARM SET-UP ?
BRSET STAT4,$08,TG57 NO, ALARM DISPLAY ?
BRSET PORTD,Y,$20,DMD NO, STANDBY ?
BRSET STAT,$01,NACS2 NO, FREQ. MODE ?
JMP DOWN YES, STEP DOWN
********************************
** Alarm dec. (hours/minutes).**
********************************
ALSU2 BRSET STAT4,$40,IHRD YES, SET-UP HOURS ?
TST AMIN NO, MINUTES
BEQ MZ
DEC AMIN
BRA T5SD
MZ LDAA #59
STAA AMIN
BRA T5SD
IHRD TST AOUR
BNE HZ
LDAA #24
STAA AOUR
DEC AOUR
T5SD LDAA #80 10 SECOND TIMEOUT
STAA DIST
BSET STAT4,$01 SET DISPLAY TRANSIENT FLAG
BCLR PORTA,Y,$10 DEMUTE
DMD RTS
NACS2 LDAB PSNP
BEQ CONTD PS EDIT CHARACTER CHANGE ?
***************
** P-S Edit dec. (ASCII). **
***************
PSN1 LDX #PSN-1
ABX
LDAA 0,X YES
DECA DECENT ASCII VALUE
CMPA #$20 SPACE
BLS MKE7A LESS OR EQUAL ?
CMPA #$2E NO, .
BLS MKE20 LESS OR EQUAL ?
CMPA #$30 NO, 0
BLO MKE2E LESS ?
CMPA #$39 NO, 9
BLS CNTS LESS OR EQUAL ?
CMPA #$41 NO, A
BLO MKE39 LESS ?
CMPA #$55A NO, Z
BLS CNTS LESS OR EQUAL ?
CMPA #$561 NO, a
BLO MKE5A LESS ?
CMPA #$77A NO, z
BLS CNTS LESS OR EQUAL ?
MKE20 LDAA #$20 MAKE SPACE
BRA CNTS
MKE2E LDAA #$2E MAKE .
BRA CNTS
MKE5A LDAA #$5A MAKE 2
BRA CNTS
MKE7A LDAA #$7A MAKE Z
BRA CNTS
MKE39 LDAA #$39 MAKE A
CNTS STAA 0,X
LDAA #80 OUTCH STAA DIST
BSET STAT4,$01 SET DISPLAY TRANSIENT FLAG
BCLR STAT4,$08 NOT ALARM DISPLAY MODE
RTS
CONTD
BSET   PORTA,Y,$10        MUTE
LDAA   LED              PROG. No. DECREMENT
BRSET  STAT2,$80,RETUNE  IF SWITCHED TO TA DON’T DECREMENT
PMT1   DECA              DECREMENT PROGRAM NUMBER
BPL    SK2P             TOO FAR ?
LDAA   #$9             SAVE NEW PROGRAM NUMBER
SK2P   STAA             CHANGE PROGRAM NUMBER IN NVM
LOAD   #$120           STAA LED SAVE NEW PROGRAM NUMBER
JSR    WRITE1
PULA
BRCLR  STAT4,$80,RETUNE2  TA SWITCHED ?
BCLR   STAT4,$80       YES, MANUAL RETURN FROM TA
LDAA   #$9
STAA   REARET
RTS

RETUNE2 JSR   DOIT     NEW PROGRAM
JSR    P$170
LDX    #$64           WAIT 100ms
JSR    SWOB
BCLR   PORTA,Y,$10   DEMUTE
BCLR   STAT2,$802    KILL ANY PENDING RDS GROUP
BCLR   STAT3,$501    AND INHIBIT FM FS-NAME CLEARING
BCLR   STAT,$10    RE-ENABLE RDS DATA CLEARING
RTS

FOK   LOAD   #$10
MUL    ADDB  #$5C
STAB   SMEM
ADCA   #$26
STAA   SMEM+1
JMP    NEW

TASW
CLR8
ADDAB  #$10
JSR    READ1     FIND PI
INCB
CMPA   PION      MSB OK ?
BNE    TNP
DEC8
JSR    READ1
CMPA   PION+1    LSB OK ?
BNE    TNP
SUBB   #$12     YES, FOUND IT
JSR    READ1
PSHA
ANDA   #$80     NVM INHIBIT FLAG SET ?
BQ     TASOK     NVM INHIBIT MESSAGE
LDAA   #$8
BRA    ABTA
PSHA
STAA   SMEM+1
JSR    NEWSUB2
JMP    NEW

TNP
CMNB   #$252    TRY NEXT RECORD
BLO    TPIC
PSHA
LDAA   #$7
ABTA   STAA     REARET
PULA
BCLR STAT4,$80  PI MATCH NOT FOUND, FORGET IT
RTS

******************************************************
* Program store/recall. *
******************************************************

DOIT       BRSET STAT2,$80,TASW
          LOADB #12
          MUL
          BRSET STAT5,$40,STORE
          JMP RECALL

******************************************************
* NVW write, sub-address in X. *
******************************************************

STORE       BCLR SMEM+1,$80
            BRCLR STAT5,$20,SKTA
            BSIT SMEM+1,$80
            SKTA
            LOADA SMEM+1
            JSR WRITE1
            LOADA SMEM
            JSR WRITE1
            LOADA SMEM+2
            JSR WRITE1
            LOADA SMEM+3
            JSR WRITE1
            LOADA SMEM+4
            JSR WRITE1
            LOADA SMEM+5
            JSR WRITE1
            LOADA SMEM+6
            JSR WRITE1
            LOADA SMEM+7
            JSR WRITE1
            LOADA $30
            JSR WRITE1
            LOADA $00
            JSR WRITE1
            LOADA $00
            BSIT
            FINST

PSNOK       JSR WRITE1
            LOADA PSN+1
            JSR WRITE1
            LOADA PSN+2
            JSR WRITE1
            LOADA PSN+3
            JSR WRITE1
            LOADA PSN+4
            JSR WRITE1
            LOADA PSN+5
            JSR WRITE1
            LOADA PSN+6
            JSR WRITE1
            LOADA PSN+7
            JSR WRITE1
            LOADA PI
            JSR WRITE1
            LOADA PI+1
            FINST

FINST       JSR WRITE1
            BCLR STAT5,$40 CLEAR STORE MODE
            RTS

************
*          *
***************
** NVW read, sub-address in X. **
***************

RECALL BSR NEWSUB
JMP NEW

NEWSUB JSR READ1
STAA SMEM+1
BCLR STAT5,$20
BRCLR SMEM+1,$80,NEWSUB2
BSET STAT5,$20

NEWSUB2 JSR READ1
CMPA #$FF
BNE NOTFF2
STAA SMEM+1
BCLR STAT5,$20
BRCLR SMEM+1,$80,NEWSUB2
BSET STAT5,$20

NOTFF2 STAA SMEM
JSR READ1
STAA PSN
JSR READ1
STAA PSN+1
JSR READ1
STAA PSN+2
JSR READ1
STAA PSN+3
JSR READ1
STAA PSN+4
JSR READ1
STAA PSN+5
JSR READ1
STAA PSN+6
JSR READ1
STAA PSN+7
RTS

***************
** NVW read & write one byte. **
***************

READ1 JSR GETAD
LOAA 0,X
INCB
RTS

WRITE1 LDY #$1000
BSET PPROG,Y,$16 SET EELAT, ERASE & BYTE ERASE BITS
BSR WBYTE ERASE BYTE
JSR DBOUNC WAIT 15 ms
BSET PPROG,Y,$02 SET EEPGM BIT TO START PROGRAMMING
DCSB

WBYTE JSR GETAD
STAA 0,X LATCH DATA
BSET PPROG,Y,$01 SET EEPGM BIT TO START PROGRAMMING
JSR DBOUNC WAIT 15 ms
 CLR PPROG,Y STOP
INCB
RTS

GETAD PSHA
PSHB
JSR BAND GET BAND
LDX #$B618 EEPROM START ADDRESS
TBA
CMFA #1 FM ?
BLS FMB
LDAB #122 NO, AM
ABX
CMFA #2 MW ?
BQF FMB

AN494
MOTOROLA
Application Note

ABX
BRCLR PORTE,Y,$40,SWB2 SECOND BANK ?
ABX YES

SWB2
* BRCLR PORTE,Y,$80,FMB SECOND PAIR OF BANKS ?
* ABX YES
* ABX
FMB PULB
PULA
ABX
RTS

********************************
** RDS displays. **
********************************

RTDSP BRSET PORTD,Y,$20,SRT STANDBY ?
BRSET STAT5,$02,NORT ALREADY RDS DISPLAY ?
BRCLR STAT2,$04,NORT ALREADY RT DISPLAY ?

NOTRT BSET STAT5,$02 SET RDS DISPLAY FLAG
LDAA RTDIS YES, MOVE ON
INCA
CMPA #26
BBQ NORT
STAA RTDIS
LDAA #100
STAA DIST
BSET STAT4,$01 RE-START TRANSIENT TIMEOUT
RTS

NORT JSR CLTR
BSET STAT2,$04 SET RT DISPLAY FLAG
LDAA #9
STAA DISP1
LDAA #1
STAA DISP2
RTS

************************************************************************
* Increment and decrement routines. *
************************************************************************

UP BSR LDXR
IF INC SMEM NO, INCREMENT LSB
BNE TT1 IF NOT LEAVE MSD
INC SMEM+1 YES, INCREMENT MSB
TT1 DECB
BNE IF ALL DONE ?
BRA NEWJ

DOWN BSR LDXR
DF TST SMEM NO, IS LSB ZERO ?
BNE TT2 IF NOT LEAVE MSD
DEC SMEM+1 DECREMENT MSB
TT2 DEC SMEM DECREMENT LSB
DECB
BNE DF ALL DONE ?
BRA NEWJ

NEWJ JSR NEW
JSR P5170
BCLR PORTA,Y,$10 DEMUTE
RTS

LDXR BCLR STAT6,$08,LDXR2 AM ?
BSET STAT2,$80 YES, CLEAR PS NAME
BRA NFMB

LDXR2 BSET STAT2,$01 NO, FM, ENABLE PS NAME CLEARING
NFMB JSR BAND GET BAND
TBA
LDAB #1 SINGLE STEP (1,5,10 kHz FOR MW,SW,FM)
BCLR STAT1,$02,SRT LARGE STEPS SELECTED ?
CMPA #1 YES, BAND 3 (SW) ?
BEQ    SRT
LDAB   #5
CMPA   #2
BNE    SRT
LDAB   #9
BRCLR  STA6,$40,SRT
INCB   YES
SRT    RTS

****************************************************************************
*                              TA test.                              *
****************************************************************************
TEST   BRSET   PORTD,Y,$20,AOB   STANDBY ?
LD      #$C5B1
STD     PION
BRSET   STAT4,$04,NABT   TA SWITCHING ENABLED ?
LDAA    #1
STAA    REARET
AOB    RTS
NABT   BSET   STAT4,$80   YES, DO IT
RTS    

*****************************************************************************
* Store key.                                                              *
*****************************************************************************
SAVE   BRCLR   STAT4,$08,NAME   ALARM DISPLAY ?
BRCLR   STAT4,$10,NTB2   YES, ALARM ARMED ?
BRSET   STAT4,$20,AISM   YES, ALREADY SET-UP MODE ?
BSET    STAT4,$60   NO, ENTER SET-UP MODE, HOURS
A5SD    LDAA    #$8
BRA     SUT
AISM    BRSET   STAT4,$40,MSM   YES, SET-UP HOURS ?
BCLR    STAT4,$20   NO, CANCEL SET-UP
BRA     A5SD
MSM    BCLR   STAT4,$40   YES, MAKE IT MINUTES
BRA     A5SD
NAME    BRSET   PORTD,Y,$20,NTB2   STANDBY ?
BRSET   STAT5,$01,NFM   NO, FREQUENCY MODE ?
BRSET   STAT5,$40,ASM   YES, STORE MODE ?
BSET    STAT5,$40   NO, ENTER STORE MODE
RTS    
ASM    LDAA    LED
JMP     DOIT   SAVE
NFM    LDAA    PSNP   NOT FREQUENCY MODE
BNE     SKPCLR  SET
JSR     CLRTR  UP
SKPCLR   INC    PSNP   PS-NAME CHANGE MODE
LDAA    PSNP
CMPA    #$8
BLS     NTB3
CLR     PSNP
NTB3    LDAA    #$9
SDT     STAA    DIST
BSET    STAT4,$01   SET DISPLAY TRANSIENT FLAG
RTS    

*****************************************************************************
*        * PROG, the displayed number is added to * *
*        * the IF offset, converted to binary and * *
*        * stored in SMEM & SMEM+1. * *
*        * NEW takes binary working frequency * *
*        * in SMEM & SMEM+1 converts it to BCD and * *
*        * subtracts the IF offset. * *
*****************************************************************************

AN494

MOTOROLA
Application Note

PROG BRSET STAT,$01,NEW STATION MODE ?
JSR IFO P < IF OFFSET
JSR ADB Q < FREQ + IF
JSR BAND
BNE ONE BAND 3 (SW) ?
JSR ADD YES, DIVIDE BY 5, Q < 2 X (FREQ + IF)
LDX #5
LPP LDAA RQ-1,X MOVE ALL DIGITS
STAA RQ,X IN Q DOWN ONE
DEX PLACE TO DEVIDE
BNE LPP BY 10 (Q < Q/5)
ONE JSR BCON CONVERT Q TO BINARY
NEW JSR DCON CONVERT TO BCD IN Q
JSR BAND
BNE STIF BAND 3 (SW) ?
STX NUM1 YES
LDX #RP
JSR ADD P < 2Q
LDX #RP
STX NUM1
LDX #RQ
JSR ADD Q < 3Q
LDX #RQ
JSR ADD Q < 5Q
STIF JSR IFO P < IF OFFSET
BRSET STAT,$04 JMP SUB Q < (RATIO X STEP) -IF

*********************************************************
*         * The IF offset is selected according to *
*         * the required band and placed in "RP."   *
*********************************************************
IPO BSR BAND FIND BAND
BRSET PORTA,Y,$04,NOTN NEGATIVE FM IF ?
CMPB #1 YES
BHI NOTN BUT IS IT FM ?
LDAB #4 YES, FIFTH IS FROM TABLE
NOTN LDAA #6
MUL TIMES 6
LDX #1FS
ABX
LOD #RP
LP6 LDAAA 0,X TRANSFER
STAA 0,Y INTO RP
INX
INY
CPY #RP+6
BLO LP6 DONE ?
LDO $1000 RE-INITIALISE Y
LDX #RP
SET-UP POINTERS
STX NUM2
LOD #RQ
STX NUM1
RTS
IFS FCB 0,0,1,0,7,0 10.70 MHz FM OSC HIGH
FCB 0,0,1,0,7,0 10.70 MHz FM OSC HIGH
FCB 0,0,0,4,5,5 455 kHz SW/MW
FCB 0,1,0,7,0,0 10.70 MHz SW (EXT/5 FOR 5157)
FCB 9,9,8,9,3,0 -10.70 MHz FM OSC LOW
BAND LDABB PORTA,Y GET BAND
ANDB #$03
LDX #RQ
STX NUM2
CMPB #3 BAND 3 (SW, /5) ?
**Mode change & clear routines.**

---

### Mode

**BRSET PORTD,Y,0,CLP**

STANDBY ?

**JSR CLTR**

**JSR PROG**

SEND DISPLAYED FREQUENCY

**SKIP**

**BRCLR STAT,01,SK**

FREQUENCY MODE ?

**BCLR STAT,01**

NO, SET TO FREQUENCY MODE

RTS

**SK**

**BCLR STAT,01**

FREQ. MODE, CLEAR STORE MODE

**BRCLR STAT,02,NNTR**

NEW FREQUENCY ENTERED ?

**BSET PORTA,Y,20**

YES, MUTE

**JSR DBNC**

WAIT 15ms

**JSR P5170**

**LDX #64**

**JSR SKDB**

WAIT 100ms

**BCLR PORTA,Y,1**

DE-MUTE

**BCLR STAT2,02**

AND KILL ANY PENDING RDS GROUP

**SKSM**

**BCLR STAT,01**

CLEAR RETUNE FLAG

RTS

**NNTR**

**BSET STAT,01**

NO, RETURN TO STATION MODE

**BCLR STAT,04**

CANCEL STORE MODE

RTS

**CLEAR**

**BRSET PORTD,Y,0,CLP**

STANDBY ?

**BSET STAT,01,SM**

NO, STATION MODE ?

**BSET STAT,01**

FREQUENCY CHANGED

**CLAL BSR CLQ**

NO, CLEAR Q

**SM LDAA PSNP**

**BEQ SPCC**

**JSR PSC**

**SPCC JSR CLTR**

CLEAR DISPLAY TRANSIENTS

**BSET STAT,2**

9 (MW), 50 (FM) kHz STEPS

RTS

**KHZ BCLR STAT,2**

1 (MW), 10 (FM) kHz STEPS

**CLP RTS**

**CLQ LDX #RQ**

CLEAR RQ

**L2 LDAA SMEM**

CLEAR 6 BYTES

**LDAA COUNT**

STARTING AT X

**CR CLR 0,X**

**INX**

**DEC COUNT**

**BNE CR**

DONE ?

**RTS**

**CLUD BRCLR STAT4,001**

CLEAR DISPLAY TRANSIENT FLAG

**CLUD2 BRCLR STAT2,004**

CANCEL RT DISPLAY

**CLUD RTDIS**

**CLUD BRCLR STAT,028**

NOT ALARM (DISPLAY OR SET-UP)

**CLUD BRCLR STAT,006**

NOT RT OR SLEEP DISPLAY

**CLUD PSNP**

NOT PS-EDIT

**RTS**

---

**BCON CLR SMEM**

CLEAR WORKING

**CLR SMEM+1**

FREQUENCY LOCATIONS

**LDX #0**

**L2 LDAA SMEM**

LS BYTE

**LSLA 2xLSB**

**STAA W1**

SAVE 2xLSB

**ROL SMEM+1**

2xMS BYTE

**LOA SMEM+1**

**STAA W2**

SAVE 2xMSB

---

AN494

MOTOROLA

49
LDAA W1 2xLSB
LSLA SMEM+1 4xLSB
LSLA W1 8xLSB
ROL SMEM+1 8xMSB
ADDA W1 10xLSB
STAA SMEM
LOADA SMEM+1
ADCA W2
STAA SMEM+1
ADCA W2 10xMSB
STAA SMEM+1
INX FETCH
LDAA RQ,X NEXT
ADDA SMEM DIGIT
STAA SMEM AND
LDAA #0 (CLRA CLEARS THE C BIT)
ADCA SMEM+1 ADD IT TO WORKING
STAA SMEM+1 FREQUENCY
CPX #5 DONE ?
BNE L2
RTS

*************************************************************************
* Clear NVM - not used. *
*************************************************************************

CLRNVW CLR COUNT
CLREP LDAA #$FF
LDAB COUNT
JSR WRITE1
INC COUNT
BNE CLREP
CLRA
LDAB #120 CLEAR MAX. PROG. No.
JMP WRITE1

*************************************************************************
* Addition and subtraction of BCD numbers. *
*************************************************************************

SUB STX W5 ANSWER POINTER
COM2 LDX NUM2 9S COMPLEMENT
CMP LDAA #$06 SECOND NUMBER
LOOP3 LDAA #$09
SUBA S,X SUBTRACT FROM 9
STAA S,X AND PUT IT BACK
DEX
DECB
BNE LOOP3
CLR CARRY SET CARRY TO ONE
INC CARRY BEFORE ADDING
BRA AD ADD FIRST NUMBER

ADD CLR CARRY
STX W5 ANSWER POINTER
AD LDAB #$06
LDX NUM1 1st No. POINTER
STX W3
LDX NUM2 2nd No. POINTER
STX W4
LOOP LDX W3
LODAA S,X
DEX
STX W3
LDX W4
ADD A S,X ADD
DEX
STX W4
ADD A CARRY SET ON ADDITION OVERFLOW
CLR CARRY OR POS. RESULT SUBTRACTION
BSR ADJ DECIMAL ADJUST
LOX  W5
STAA 5,X   SAVE ANSWER
DEX
STX  W5
DECX
BNE  LOOP  DONE ?
RTS

AJ  SUBA  #10  YES, SUBTRACT 10
INC  CARRY  AND RECORD CARRY
ADJ  CMPA  #10
BHS  AJ  10 OR MORE ?
RTS

******************************************
*       *
*  Current binary divide ratio in SMEM & *
*  SMEM+1 is converted to decimal in RQ.  *
*       *
******************************************

DCON  LDAA  SMEM+1  TRANSFER CURRENT
STAA  W2  FREQUENCY DIVIDE
LDAA  SMEM  RATIO INTO
STAA  W1  WORKING AREA

DCON2  LDX  #RR  CLEAR
STX  NUM1
JSR  CLRAS  RR
INC  RR+5  RR <- 1
JSR  CLQ  CLEAR RQ
LDAA  #14  14 BITS TO CONVERT
STAA  W6

LOOP2  LSR  W2  MOVE OUT
ROR  W1  FIRST (LS) BIT
BCC  NXT  ZERO
LDX  #RQ  ONE, ADD
STX  NUM2  CURRENT VALUE
BSR  ADD  OF RR
NXT  LDX  #RR  ADD RR
STX  NUM2  TO
BSR  ADD  ITSELF
DEC  W6  ALL
BNE  LOOP2  DONE ?
RTS

******************************************************************************
*       *
*  Delay (X x 1.5mS).  *
*       *
******************************************************************************

DBNC  LDX  #100  150mS
BRA  SKDB

DBOUNC  LDX  #10  APPROX 15mS WITH A 8.388 MHz XTAL
SKDB  STX  W6  X x 1.5mS
DLP  LDX  #$FF  PAUSE
DLOOP  BRN  256X12
BRN  "  CYCLES
DEX
BNE  DLOOP
DEC  W6+1
BNE  DLP
ABO
RTS

******************************************************************************
*       *
* Serial output routine to the MC145170.  *
*       *
******************************************************************************

P5170  BCLR  PORTB,Y,$01  CLOCK LOW
BCLR  PORTB,Y,$10  LE LOW
LDAA  #0  CLEAR
BSR  SQU81  CONTROL REGISTER
BSET  PORTB,Y,$10  LATCH IT

AN494
MOTOROLA
BCLR PORTB,Y,$10 LE LOW
LDAA SMEM+1
ANDA #$7F
BSR SQU8I SEND MSBYTE
LDAA SMEM AND LSBYTE OF
BSR SQU8I NEW FREQUENCY
BSET PORTB,Y,$10 LATCH IT
BCLR PORTB,Y,$10 LE LOW
LDAA #$03 SEND
BSR SQU7I REFERENCE
LDAA #$20 DIVIDE RATIO
BSR SQU8I 800 = 8MHz/10kHz
BSET PORTB,Y,$10 LATCH IT

******************************************
*                              *
*  Serial output routine to the MC145157. *
* *        *
******************************************
P5157
LDAA SMEM TRANSFER SMEM AND
LSLA W4 MEM+1 TO TEMPORARY
STAA SMEM+1 LOCATIONS AND MOVE
LDAA SMEM+1 UP ONE BIT TO INCLUDE
RLA THE 5157 CONTROL BIT.
BSR SQU7 SEND MSBYTE (7 BITS)
LDAA W4 AND LSBYTE OF
BSR SQU8 NEW FREQUENCY
BSET PORTB,Y,$08 LATCH
BCLR PORTB,Y,$08 IT
LDAA #$4E SEND 15 BIT (14+1)
BSR SQU7 REFERENCE
LDAA #$21 DIVIDE RATIO
BSET PORTB,Y,$08 LATCH IT
BCLR PORTB,Y,$08 ALL LOW (5157/70 SWITCHED OFF)
RTS

******************************************************************************
*                              *
*  Subroutines for the MC145157/170. *
*                              *
******************************************************************************
SQU8I LDAB #8 SEND 8 BITS
BRA S1I
SQU7I LSLA MOVE OUT MS BIT
LDAB #7 AND SEND OTHER ?
S1I LSLA MOVE 1 BIT INTO "C"
BCC S2I ZERO ?
BSET PORTB,Y,$02 NO
S2I BSET PORTB,Y,$01 CLOCK
BCLR PORTB,Y,$01 IT
BCLR PORTB,Y,$02
DEC S1I ANY MORE ?
RTS

SQU8 LDAB #8 SEND 8 BITS
BRA S1
SQU7 LSLA MOVE OUT MS BIT
LDAB #7 AND SEND OTHER ?
S1 LSLA MOVE 1 BIT INTO "C"
BCC S2 ZERO ?
BSET PORTB,Y,$02 NO
S2 BCLR PORTB,Y,$01 CLOCK
BCLR PORTB,Y,$01 IT
BCLR PORTB,Y,$02
DEC S1 ANY MORE ?
RTS
************************************************
**        Toggle 9/10 kHz step (MW).          **
************************************************

T910
BRSET STAT6,$40,CBH
BSET STAT6,$40
RTS

CBH
BCLR STAT6,$40
RTS

*****************************************************************************
* LINK batch files (RLE.BAT & RDE.LD) and PCBUG11 Vectors.                *
* ILD11 RADE.O FNCE.O RDSE.O -MKUF E32.MAP -G RDE -O RDE.OUT               *
* INEX RDE.OUT -O RDE.0                                                  *
* TYPE E32.MAP                                                         *
* section .RAM1 BSS origin 0x0000                                         *
* section .RAM2 BSS origin 0x0100                                         *
* section .RAM3 BSS origin 0x0200 $E32                                   *
* section .ROM1 origin 0xD000 $9000                                      *
* section .ROM2 origin 0xE000 $9c00                                      *
* section .ROM3 origin 0xF000 $A000                                      *
* section .VECT origin 0xBFC1                                            *
* section .VECT2 origin 0xFFD6 ($FFD6)                                   *
*****************************************************************************

* SECTION .VECT
* JMP START SCI
* JMP START SPI
* JMP START PULSE ACCUMULATOR EDGE
* JMP START " " " OVER
* JMP START TIMER OVER
* JMP START " IC4/OC5
* JMP START " OC4
* JMP START " OC3
* JMP START " OC2
* JMP START " OC1
* JMP START " IC3
* JMP START " IC2
* JMP START " IC1
* JMP TINTB RTI
* JMP SHAFTX IRA NOT USED, XIRQ USED BY PCbug11
* JMP START SWI
* JMP START ILLEGAL OP CODE
* JMP START COP
* JMP START CLOCK MONITOR
* JMP START RESET

*****************************************************************************
* * * MC68HC11E32 Vectors.                   * *
* * *                                      *
*****************************************************************************

SECTION .VECT2
ORG $FFD6

FDB START SCI
FDB START SPI
FDB START PULSE ACCUMULATOR EDGE
FDB START " " " OVER
FDB START TIMER OVER
FDB START " IC4/OC5
FDB START " OC4
FDB START " OC3
FDB START " OC2
FDB START " OC1
FDB START " IC3
FDB START " IC2
Symbol table

.RAM1 1 00000000 | CONTD 4 000005df | INSLP 4 000003fe | NNTR 4 00000968 | RECALL 4 00000702
.RAM2 2 00000000 | CONTI 4 00000529 | IOK 4 00000229 | NOTC 4 000000c6 | RETURN 4 00000500
.RAM3 3 00000000 | COUNT 1 0000009a | IOOK 4 0000053c | NOPS 4 000000c6 | RETURN2 4 00000603
.ROM1 4 00000000 | CPSL 4 000003cc | IRQ 4 00000006 | NORT 4 000007c0 | RJ 4 0000030a
.VECT2 5 00000000 | CR 4 0000099a | ITMP1 1 00000069 | NOTCH 4 000002bc | RKEY 4 000002ea

Section synopsis

174) .RAM1
256) .RAM2
109) .RAM3
2889) .ROM1
42) .VECT2

Application Note

AN494
54 MOTOROLA
AN494

MOTOROLA

55
Application Note

Motorola reserves the right to make changes without further notice to any products herein. Motorola makes no warranty, representation or guarantee regarding the suitability of its products for any particular purpose, nor does Motorola 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 Motorola 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. Motorola does not convey any license under its patent rights nor the rights of others. Motorola 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 Motorola product could create a situation where personal injury or death may occur. Should Buyer purchase or use Motorola products for any such unintended or unauthorized application, Buyer shall indemnify and hold Motorola 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 Motorola was negligent regarding the design or manufacture of the part. Motorola and are registered trademarks of Motorola, Inc. Motorola, Inc. is an Equal Opportunity/Affirmative Action Employer.

How to reach us:

USA/EUROPE/Locations Not Listed: Motorola Literature Distribution; P.O. Box 5405, Denver, Colorado 80217. 1-303-675-2140 or 1-800-441-2447

JAPAN: Motorola Japan Ltd.; SPS, Technical Information Center, 3-20-1, Minami-Azabu, Minato-ku, Tokyo 106-8573 Japan. 81-3-3440-3569

ASIA/PACIFIC: Motorola Semiconductors H.K. Ltd.; Silicon Harbour Centre, 2 Dai King Street, Tai Po Industrial Estate, Tai Po, N.T., Hong Kong. 852-26668334

Technical Information Center: 1-800-521-6274

HOME PAGE: http://www.motorola.com/semiconductors/