```
.TITLE DUT_HH.ASM
; This program is for testing the CT25 Hammerhead telephone 548-C Board
.MACLIST OFF
.PL 71
.PW 136
.LIST OFF

.INCLUDE    CGTMACRO.H      ;Header include files
.INCLUDE    CALLS.H
.INCLUDE    DUTDEF.H
.INCLUDE    DUTGLOBAL.H
.INCLUDE    GLBDATA.H
.INCLUDE    SYSDEF.H
.INCLUDE    SYSREG.H
.INCLUDE    DUT0229.H
.INCLUDE    DUT_MAC.H

.LIST ON
.SYMBOLS

;*************************************************************************
; Revisions table
; 3/16/96 ---- Made the following changes to the SAMSUNG_RESET subroutine
;              1. Disabled DUT output port @ U39
;              2. Added go on-hook and go off-hook lines
;              3. Increased delay to 750 milli-seconds after turning power off
;         ---- Made the following changes to the DTMF GAIN TEST
;              1. Changed the limit values for checking the 948 Hz gain
; 3/19/96 ---- Added code to set the ISET DAC output of U14 to +5Vdc
;         ---- Changed the fixture switch closed condition from N/O to N/C
; 6/06/96 ---- Added repeat test loop if an error occurs in the DTMF Gain test
;
;Internal error code 1: LCD_SCREEN_1 subroutine was passed an invalid line
;                       number to print
;Internal error code 2: DSP_ERROR
;                          ERR_3 routine went through message table without
;                          locating the appropriate message for the error
;                          code in the ERR_MSG buffer.

.CODE
;
;                    12345678901234567890       ;20 Column LCD Display.
APP_NAME:   .BYTE    ' CIDCO PCB-C TESTER ',$00  ;Name of application
APP_REV:    .BYTE    ' Rev 1.02 - 29FEB96 ',$00  ;Revision and last modified
                                                 ;date
APP_MEXE:   .DEFW    INITIAL_START               ;Where the starting point is
ID:         .BYTE    $01
APP_EXESZ   .DEFW    INITIAL_END                 ;Size of this application
;                                                ;program

*************************************************************************
* Name     : INITIAL_START                                              *
* Function : Initialize the DUT                                         *
* Entry    : None                                                       *
* Return   : None                                                       *
*************************************************************************

INITIAL_START:
            ACALL   acMANUAL_PB
            ACALL   acACTIVE_DUT_LIB    ;Must call to activate the DUT
                                        ;Library
; ADDED ON 02/26/96
            ACALL   acACPL_RLY_OFF      ;Make sure AC coupling relay is off

            LDAB    #TGT_PC0
            LDAA    #$00                ;PC07=EN_LINE_CUR_RLY initially = L
            STAA    LTGT_PC0
            JSR     _WR_DUT

            ACALL   acLCD_CLEAR_DISPLAY ;Clear LCD
            JSR     _INITDUT_EEXE       ;Resets all analog O/P Ports
                                        ;to 0.0 Volts
                                        ;All ananlog O/P to target are
                                        ;disabled. Only four channels.
                                        ;All relays are in the off pos.
                                        ;All LED's off.
                                        ;All P.B. Matrix row/col off.
                                        ;Microwire initialized
                                        ;RMS PGA set to gain of 1
```

```
            JSR     _ENA_DUT_LIP0           ;Enable input from LIP0
            JSR     _ENA_DUT_LIP1           ;Enable input from LIP1

            LDAA    #$01                    ;Initialize I/O port 1
            JSR     _WR_DUTIO1              ;Bit 0 = 1, Bit 1->7 = 0
            JSR     _ENA_DUTIO1
;           STAA    IO_PORT1_BANK_REG       ;Save copy to bank register

            LDAA    #11111101B
            JSR     _CLR_DUT_LOP_b          ;clear bits 2-7
            LDAA    #00000010
            JSR     _SET_DUT_LOP_b          ;set bits 0-1
            CLR     CAPRLY_FLAG             ;Clear CAP relay flag
            CLR     PSW1_FLAG               ;Clear the init PSW1 flag
            JSR     PWR_OFF                 ;
            JSR     _SET_DUTIO2_IP          ;Set DUTIO2 port to input
            JSR     _ENA_DUTIO2             ;Enable the IO port #2
            LDAA    #0                      ;Set DUT current sense to off
            LDAB    #255                    ;
            JSR     _SET_DUT_DAC            ;
            CLR     START_PB_FLAG           ;Clear START pushbutton flag

;*****************************************************************************
;  MAIN PROGRAM LOOP

MAIN:
            LDX     #START_MSG0
            JSR     LCD_SCREEN_4            ;Display the power up message
            JSR     CLR_BUFF                ;Clear the error buffer
            ACALL   acHV_RELAY_ON           ;Apply +/-200Vdc power
            LDAA    START_PB_FLAG           ;If first time through test or fixture
            BEQ     MAIN_PB_BYPASS          ;lid has been opened, bypass START
MAIN_PB:                                    ;pushbutton scan
            JSR     _DUT_START_PB           ;Wait for start button
            BNE     MAIN_PB                 ;Loop until receive start button press
MAIN_PB_BYPASS
            LDAA    #1                      ;Set START pusbutton flag
            STAA    START_PB_FLAG
            LDAA    #PASS_LED^FAIL_LED
            JSR     _CLR_DUTIO1_b
            LDAA    #TIP_LED
            JSR     _SET_DUTIO1_b

            JSR     CK_DUT                  ;Check PCB installation

            LDX     #TEST1_MSG0             ;Get pointer to message
            JSR     LCD_SCREEN_4            ;Display testing message
            LDY     #$0005
            JSR     WAIT

; Start of test routines

            JSR     PWR_TST                 ;1 Go to DC voltage (AC & Batt)check

            JSR     DC_RES_TST              ;2 Go to off hook DC res. check

            JSR     LIU_TST                 ;3 Go to LIU check

;           JSR     DIAL_TONE_TST           ;4 Go to dial tone detect test

            JSR     HOLD_REL_TST            ;5 Go to hold release test

;            JSR     AUD_AMP_TST            ;6 Go to audio amplifier test

            JSR     RXD_GAIN_TST            ;7 Go to handset receive gain check

            JSR     TXD_GAIN_TST            ;8 Go to handset transmit gain check

;           JSR     RXD_SP_TST              ;9 Go to speakerphone receive gain test

;           JSR     TXD_SP_TST              ;10 Go to speakerphone transmit gain test

;           JSR     RXD_SWITCH_TST          ;11 Go to transmit-to-receive switching test

;           JSR     TXD_SWITCH_TST          ;12 Go to receive-to-transmit switching test

            JSR     DTMF_GAIN_TST           ;13 Go to DTMF gain check

            JSR     FSK_GAIN_TST            ;14 Go to FSK gain check
```

```
            JSR     CAL_2130_TST        ;15 Go to calibrate 1st & 2nd 2130 filter stages

            JSR     CAL_2750_TST        ;16 Go to calibrate 2750 filter stage

            JSR     CAS_DET_TST         ;17 Go to the CAS detect check

            JSR     RING_TST            ;18 Go to ring det & piezo drive check

main_ret:
            JSR     PWR_OFF             ;Turn off primary power to DUT
            JSR     CK_PASS             ;Check for Pass or Fail & turn on LED
            JSR     DSP_ERR             ;Display error codes/messages if any

; ADDED ON 02/26/96
            ACALL   acHV_RELAY_OFF      ;Switch out +/-200Vdc power
;                                       ;and apply +/- 20Vdc
MAIN_END:   JMP     MAIN                ;Loop back to start of Main program
            EJECT
;*******************************************************************************
; The following subroutines are the individual tests for the CT-25CWi 'C' Board
;
;-------------------------------------------------------------------------------
PWR_TST:
;  Check all DC voltages (AC adapter & battery) and control signals.
;  I/P: None
;  O/P: None
;
            LDAB    #PWR_TST_NUM        ;Set to the correct test number
            STAB    TEST_NUM            ;for the err_rpt routine.
            LDAA    #1                  ;reinitialize for each test (err_rpt)
            STAA    ERR_CODE            ;Set up error code

            LDAB    #$02                ;Display line number 3
            LDX     #TEST_NUM_2
            JSR     LCD_SCREEN_1

            LDAA    #$01                ;Set to a nonzero flag OR ELSE MFU
            STAA    RMS_DC_FLAG         ;Set up for DC measurement
            CLRA
            STAA    DC_ERR_FLAG         ;Clear DC power error flag register
            LDAA    #EPS_BIT            ;Check -EPS signal, should be active low
            JSR     _TST_DUT_LIPORT     ;with AC adapter voltage applied
            BEQ     ?pwr0               ;Report error if not active

            JSR     ERR_RPT             ;Report error
; Check if VADPT power is within limits
?pwr0
            INC     ERR_CODE            ;Set up error code
            LDAA    #1                  ;Load the analog channel number
            STAA    CHAN_NUM            ;
            JSR     GET_AVG
            CPX     #VADAPT_VAL
            BHI     ?pwr20              ;Nominal voltage > 7.8Vdc
            JSR     ERR_RPT             ;Report error
; Check if VAMP power is within limits
?pwr20
            INC     ERR_CODE            ;Set up error code
            LDAA    #4                  ;Load the analog channel number
            STAA    CHAN_NUM            ;
            JSR     GET_AVG
            CPX     #VADAPT_VAL
            BHI     ?pwr1               ;Nominal voltage > 7.8Vdc
            JSR     ERR_RPT             ;Report error
?pwr1
; Check if VDD power is within limits
            INC     ERR_CODE            ;Set up error code
            LDAA    #2                  ;Load the analog channel number
            STAA    CHAN_NUM            ;
            JSR     GET_AVG             ;Get the average reading
            CPX     #$0433              ;Check if above maximum = 5.25Vdc
            BHI     ?pwr2               ;Nominal voltage = 5.00Vdc
            CPX     #$0366              ;Check if below minimum = 4.50Vdc
            BHI     ?pwr3               ;
?pwr2
            JSR     ERR_RPT             ;Report error
            LDAA    DC_ERR_FLAG         ;Set DC power error flag
            ORAA    #1                  ;Show VDD error
```

```
            STAA    ERR_FLAG
?pwr3
; Check if VCD power is within limits
            INC     ERR_CODE            ;Set up error code
            LDAA    #3                  ;Load the analog channel number
            STAA    CHAN_NUM            ;
            JSR     GET_AVG             ;Get reading for VCD
            CPX     #$0433              ;Check if above maximum = 5.25Vdc
            BHI     ?pwr4               ;Nominal voltage = 5.00Vdc
            CPX     #$0399              ;Check if below minimum = 4.50Vdc
            BHI     ?pwr5               ;
?pwr4
            JSR     ERR_RPT             ;Report error
            LDAA    DC_ERR_FLAG         ;Set DC power error flag
            ORAA    #2                  ;Show VCD error
            STAA    ERR_FLAG
?pwr5
; Check if +VCCA power is within limits
            INC     ERR_CODE            ;Set up error code
            LDAA    #6                  ;Load the analog channel number
            STAA    CHAN_NUM            ;
            JSR     GET_AVG             ;Get the average reading
            CPX     #$05A0              ;Check if above maximum > 7.5Vdc
            BHI     ?pwr6               ;

            JSR     ERR_RPT             ;Report error
            LDAA    DC_ERR_FLAG         ;Set DC power error flag
            ORAA    #4                  ;Show +VCAA error
            STAA    ERR_FLAG
?pwr6
; Check if +VCCAX power is within limits
            INC     ERR_CODE            ;Set up error code
            LDAA    #7                  ;Load the analog channel number
            STAA    CHAN_NUM            ;
            JSR     GET_AVG             ;Get the average reading
            CPX     #$05D7              ;Check if above maximum > 7.0Vdc
            BHI     ?pwr7               ;
            CPX     #$04CC              ;Check if in range
            BHI     ?pwr8               ;
?pwr7
            JSR     ERR_RPT             ;Report error
            LDAA    DC_ERR_FLAG         ;Set DC power error flag
            ORAA    #8                  ;Show +VCCAX error
            STAA    ERR_FLAG
?pwr8
; Check if VDDX power is within limits
            INC     ERR_CODE            ;Set up error code
            LDAA    #5                  ;Load the analog channel number
            STAA    CHAN_NUM            ;
            JSR     GET_AVG             ;Get the average reading
            CPX     #$040A              ;Check if above maximum = 5.05Vdc
            BHI     ?pwr9               ;Nominal = 4.8Vdc
            CPX     #$0366              ;Check if below minimum = 4.25Vdc
            BHI     ?pwr10              ;
?pwr9
            JSR     ERR_RPT             ;Report error
            LDAA    DC_ERR_FLAG         ;Set DC power error flag
            ORAA    #$10                ;Show VDDX error
            STAA    ERR_FLAG
?pwr10
; Check if VMID power is within limits
            INC     ERR_CODE            ;Set up error code
            LDAA    #8                  ;Load the analog channel number
            STAA    CHAN_NUM            ;
            JSR     GET_AVG             ;Get the average reading
            CPX     #$0370              ;Check if above maximum = 4.3Vdc
            BHI     ?pwr11              ;Nominal = % of VDD
            CPX     #$0200              ;Check if below minimum = 3.60Vdc
            BHI     ?pwr12              ;
?pwr11
            JSR     ERR_RPT             ;Report error
            LDAA    DC_ERR_FLAG         ;Set DC power error flag
            ORAA    #$20                ;Show VMID error
            STAA    ERR_FLAG
?pwr12
; Check if VMID2 power is within limits
            INC     ERR_CODE            ;Set up error code
            LDAA    #10                 ;Load the analog channel number
            STAA    CHAN_NUM            ;
```

```
            JSR     GET_AVG             ;Get the average reading
            CPX     #$0266              ;Check if above maximum = 2.625V
            BHI     ?pwr13              ;Nominal = 50% of VDD
            CPX     #$199               ;Check if below minimum = 2.25V
            BHI     ?pwr14              ;
?pwr13
            JSR     ERR_RPT             ;Report error
            LDAA    DC_ERR_FLAG         ;Set DC power error flag
            ORAA    #$40                ;Show VMID2 error
            STAA    ERR_FLAG
?pwr14
; Switch to +9.0 Vdc battery power only
            INC     ERR_CODE            ;Increment the error code
            LDAA    #FSK_EN             ;Set -FSK.EN signal to active low
            JSR     _CLR_DUT_LOP_b      ;
            JSR     _ENA_DUT_LOP        ;Enable the output port
            JSR     _ENA_DUT_PWRELAY    ;Switch the relay to battery power
            LDX     #100                ;Delay for power to settle
            ACALL   acDELAY_MSEC          ;
; Check if VCD is within limits
            LDAA    #3                  ;Load the analog channel number
            STAA    CHAN_NUM            ;
            STX     SAVX1
            JSR     GET_AVG             ;Get the average reading
            CPX     #$0433              ;Check if above maximum = 5.25Vdc
            BHI     ?pwr15              ;Nominal voltage = 5.00Vdc
            CPX     #$0366              ;Check if below minimum = 4.50Vdc
            BHI     ?pwr16              ;
?pwr15
            JSR     ERR_RPT             ;Report error
            LDAA    DC_ERR_FLAG         ;Set DC power error flag
            ORAA    #2                  ;Show VCD error
            STAA    ERR_FLAG
?pwr16
; Switch to POTS mode (off-hook & telephone line power only)
            LDAA    #FSK_EN              ;Set -FSK.EN signal to inactive high
            JSR     _SET_DUT_LOP_b       ;
            JSR     _DIS_DUT_PWR         ;Disable DUT power
            JSR     _ENA_HKSW_RELAY      ;Go off-hook

; Check if VCD is within limits during POTs mode
            INC     ERR_CODE            ;Increment the error code
            LDAA    #3                  ;Load the analog channel number
            STAA    CHAN_NUM            ;
            JSR     GET_AVG             ;Get the average reading
            CPX     #$49A               ;Check if above maximum = 2.9Vdc
            BHI     ?pwr18              ;Nominal voltage = 2.4Vdc
            CPX     #$0199              ;Check if below minimum = 2.0Vdc
            BHI     ?pwr19              ;
?pwr18
            JSR     ERR_RPT             ;Report error
            LDAA    DC_ERR_FLAG         ;Set DC power error flag
            ORAA    #2                  ;Show VCD error
            STAA    ERR_FLAG
?pwr19
            JSR     _DIS_DUT_LOP        ;Tri-state the output port
            LDAA    #EN_9VDC            ;Enable AC adapter power to the DUT
            JSR     _ENA_DUT_PWR        ;
            JSR     _DIS_DUT_PWRELAY    ;Power relay to AC adapter side
            LDX     #250                ;Delay for power to settle
            ACALL   acDELAY_MSEC
            CLR     RMS_DC_FLAG         ;Clear flag
            RTS
            EJECT
;-------------------------------------------------------------------------------

DC_RES_TST:
;  Check off-hook DC resistance across Tip & Ring
;  I/P: None
;  O/P: None
;
            LDAB    #DC_RES_TST_NUM     ;Set to the correct test number
            STAB    TEST_NUM            ;for the err_rpt routine.
            LDAA    #1                  ;reinitialize for each test (err_rpt)
            STAA    ERR_CODE            ;Set up error code

            LDAB    #$02                ;Display line number 3
            LDX     #TEST_NUM_3
```

```
            JSR     LCD_SCREEN_1

            LDX     #$0428              ;Set line voltage to +48.0Vdc
            ACALL   acSET_DC_LINEVOLTAGE

            LDX     #10                 ;Delay 10mSec for line voltage to set
            ACALL   acDELAY_MSEC

            JSR     _ENA_HKSW_RELAY     ;Go off-hook
            LDX     #50                 ;Delay for off-hook settle
            ACALL   acDELAY_MSEC
            INC     RMS_DC_FLAG         ;Set up for DC measurement

            LDAA    #0                  ;Load the analog channel number
            STAA    CHAN_NUM            ;
            JSR     GET_AVG             ;Get the average reading
            STX     SAVX1
            CPX     #$04F5              ;Check if above maximum
            BHI     ?dc_res0            ;Range: 5.0Vdc =< Vt-r =< 6.2Vdc
            CPX     #$03E1              ;Check if in range
            BHI     ?dc_res1            ;
?dc_res0
            JSR     ERR_RPT             ;Report error
            PULY                        ;Purge the stack
            JMP     main_ret            ;Go back to main program
?dc_res1
            INC     ERR_CODE            ;Increment the error code
            LDAA    #DUTIO04            ;Do a line reversal
            JSR     _SET_DUTIO1_b       ;
            LDX     #100                ;Delay for line reversal settle
            ACALL   acDELAY_MSEC
            JSR     GET_AVG             ;Get the average reading
            STX     SAVX2               ;Negative readings are 2's compliment
            CPX     #$0CCE              ;Check if below minimum

            BHI     ?dc_res2            ;Range: -5.0Vdc =< Vt-r =< -6.2Vdc
            CPX     #$0B35              ;Check if in range
            BHI     ?dc_res3            ;
?dc_res2
            JSR     ERR_RPT             ;Report error
?dc_res3
            LDAA    #DUTIO04            ;Do a line reversal
            JSR     _CLR_DUTIO1_b       ;
dc_res4
            LDX     #100                ;Delay between tests
            ACALL   acDELAY_MSEC
            RTS
            EJECT
;-------------------------------------------------------------------------
LIU_TST:
;  Check the Line-in-use detect circuit thresholds
;  I/P: None
;  O/P: None
;
            LDAB    #LIU_TST_NUM        ;Set to the correct test number
            STAB    TEST_NUM            ;for the err_rpt routine.
            LDAA    #1                  ;reinitialize for each test (err_rpt)
            STAA    ERR_CODE            ;Set up error code

            LDAB    #$02                ;Display line number 3
            LDX     #TEST_NUM_6
            JSR     LCD_SCREEN_1

            JSR     _DIS_HKSW_RELAY     ;Go on-hook

            LDAA    #DUT_LOP_1          ;Enable -FSK_EN
            JSR     _CLR_DUT_LOP_b      ;Turn on voltage to the LIU circuit
            JSR     _ENA_DUT_LOP        ;Enable the logical output port

            LDX     #50                 ;Delay 50mSec for settle
            ACALL   acDELAY_MSEC

            LDAA    #LIU_BIT            ;Mask the LIU bit
            JSR     _TST_DUT_LIPORT     ;Read DUT input port #1
            BNE     ?liu0               ;LIU should not be active
            JSR     ERR_RPT             ;Report LIU active error
 ?liu0
            INC     ERR_CODE            ;Set up error code
 ;           LDX     #$06C7              ;Set line voltage to 17.5V
```

```
            LDX     #$0700              ;Set line voltage to 17.5V
            ACALL   acSET_DC_LINEVOLTAGE ;
            LDX     #50                 ;Delay 50mSec for line voltage to set
            ACALL   acDELAY_MSEC

; Check the Line-in-use, should be active below 18.0 Vdc

            LDAA    #LIU_BIT            ;Mask the LIU bit
            JSR     _TST_DUT_LIPORT     ;Read DUT input port #1
            BEQ     ?liu1               ;LIU should be active
            JSR     ERR_RPT             ;Report LIU not active error
?liu1
            LDAA    #DUT_LOP_1          ;Disable -FSK_EN
            JSR     _SET_DUT_LOP_b      ;Turn off voltage to the LIU circuit
;            LDX     #$0428              ;Set line voltage to 48.0V
            LDX     #$0487              ;Set line voltage to 48.0V
            ACALL   acSET_DC_LINEVOLTAGE ;
            LDX     #50                 ;Delay 50mSec for line voltage to set
            ACALL   acDELAY_MSEC
            RTS
            EJECT
;-------------------------------------------------------------------------
HOLD_REL_TST:
;  Check the hold release circuit functionalty
;  I/P: None
;  O/P: None
;
            LDAB    #HREL_TST_NUM       ;Set to the correct test number
            STAB    TEST_NUM            ;for the err_rpt routine.
            LDAA    #1                  ;reinitialize for each test (err_rpt)
            STAA    ERR_CODE            ;Set up error code

            LDAB    #$02                ;Display line number 3
            LDX     #TEST_NUM_6
            JSR     LCD_SCREEN_1

            JSR     _DIS_HKSW_RELAY     ;Go on-hook

            LDAA    #HREL_BIT           ;Mask the HOLD.REL bit
            JSR     _TST_DUT_LIPORT     ;Read DUT input port #1
            BNE     ?hrel0              ;LIU should not be active
            JSR     ERR_RPT             ;Report HOLD.REL active error
            BRA     hrel3               ;Exit with error
?hrel0
            INC     ERR_CODE            ;Set up error code
;            LDX     #$06C7              ;Set line voltage to 6.5V
            LDX     #$0700              ;Set line voltage to 6.5V
            ACALL   acSET_DC_LINEVOLTAGE ;
            LDX     #50                 ;Delay 50mSec for line voltage to set
            ACALL   acDELAY_MSEC
            LDX     #$FFFF              ;Set up timeout counter
?hrel1
; Check hold release, should be active around 6.5 Vdc

            LDAA    #HREL_BIT           ;Mask the HOLD.REL bit
            JSR     _TST_DUT_LIPORT     ;Read DUT input port #1
            BEQ     ?hrel2              ;Exit if HOLD.REL is active low
            DEX                         ;Else decrement timeout counter
            BNE     ?hrel1              ;Error if timeout counter = 0
            BRA     hrel3
?hrel2
            INC     ERR_CODE            ;Set up error code
            LDX     #10                 ;Delay 10mSec
            ACALL   acDELAY_MSEC

            LDAA    #HREL_BIT           ;Mask the HOLD.REL bit
            JSR     _TST_DUT_LIPORT     ;Read DUT input port #1
            BEQ     ?hrel4              ;HOLD.REL should still be active low
hrel3
            JSR     ERR_RPT             ;Report HOLD.REL not active error
?hrel4
;            LDX     #$0428              ;Set line voltage to 48.0V
            LDX     #$0487              ;Set line voltage to 48.0V
            ACALL   acSET_DC_LINEVOLTAGE ;
            LDX     #50                 ;Delay 50mSec for line voltage to set
            ACALL   acDELAY_MSEC
            RTS
            EJECT
;-------------------------------------------------------------------------
```

```
DIAL_TONE_TST:
;  Check the functionality of the dial tone detection circuit
;  I/P: None
;  O/P: None
;
            LDAB    #DIAL_TONE_TST_NUM  ;Set to the correct test number
            STAB    TEST_NUM            ;for the err_rpt routine.
            LDAA    #1                  ;reinitialize for each test (err_rpt)
            STAA    ERR_CODE            ;Set up error code

            LDAB    #$02                ;Display line number 3
            LDX     #TEST_NUM_5
            JSR     LCD_SCREEN_1

            INC     ERR_CODE            ;Set up error code
            LDX     #$00E2              ;Set up FSK attenuator #'0 for 30mVp-p
            ACALL   acSET_FSK0_ATT      ;
            LDX     #$00E2              ;Set up FSK attenuator #'1 for 30mVp-p
            ACALL   acSET_FSK1_ATT      ;
            LDX     #$03FF              ;Set Mark frequency to 1200 Hz
            ACALL   acSET_PSWFSK_ZERO   ;
            LDX     #$0755              ;Set Space frequency to 2200 Hz
            ACALL   acSET_PSWFSK_ONE    ;
            ACALL   acFSK_POWER_UP      ;Power up the FSK

            LDX     #50                 ;Delay for FSK to settle
            ACALL   acDELAY_MSEC        ;

            LDAA    #DTONE_BIT          ;Mask the DIAL.TONE bit
            JSR     _TST_DUT_LIPORT     ;Read DUT input port #1
            BNE     ?dtone0             ;DIAL.TONE should not be active
            JSR     ERR_RPT             ;Report DIAL.TONE active error
            BRA     dtone2
?dtone0
            INC     ERR_CODE            ;Set up error code
            LDAA    #TONE_OUT           ;Turn on the FSK signal
            ACALL   acADD_AMUX          ;Output 350/440Hz stutter dial signal
            LDX     #50                 ;Delay for FSK to settle
            ACALL   acDELAY_MSEC        ;
            LDX     #$FFFF              ;Set up timeout counter
?dtone1
            LDAA    #DTONE_BIT          ;Mask the DIAL.TONE bit
            JSR     _TST_DUT_LIPORT     ;Read DUT input port #1
            BEQ     ?dtone3             ;DIAL.TONE should be active
            DEX                         ;Else decrement timeout counter
            BNE     ?dtone1             ;Error if timeout counter = 0
            BRA     hrel3
dtone2
            JSR     ERR_RPT             ;Report DIAL.TONE error
?dtone3
            LDAA    #TONE_OUT           ;Turn off FSK signal
            ACALL   acRMV_AMUX          ;
            ACALL   acFSK_POWER_DN      ;Power down the FSK
            LDX     #50                 ;Delay 50mSec between tests
            ACALL   acDELAY_MSEC
            RTS
            EJECT
;-------------------------------------------------------------------------------
AUDIO_AMP_TST:
;  Check the gain of the audio amplifier and all confidence tone inputs
;  I/P: None
;  O/P: None
;
            LDAB    #AUDIO_AMP_TST_NUM  ;Set to the correct test number
            STAB    TEST_NUM            ;for the err_rpt routine.
            LDAA    #1                  ;reinitialize for each test (err_rpt)
            STAA    ERR_CODE            ;Set up error code

            LDAB    #$02                ;Display line number 3
            LDX     #TEST_NUM_5
            JSR     LCD_SCREEN_1


            CLR     RMS_DC_FLAG         ;Set up for RMS measurement
            LDX     #$0400              ;Set up FSK attenuator #'0
            ACALL   acSET_FSK0_ATT      ;
            LDX     #$0400              ;
            ACALL   acSET_FSK1_ATT      ;Set up FSK attenuator #'1
            LDX     #$01AB              ;Set frequency to 500Hz
```

```
            ACALL   acSET_PSWFSK_ONE    ;
            LDX     #$01AB
            ACALL   acSET_PSWFSK_ZERO
            ACALL   acFSK_POWER_UP      ;Power up the FSK signal generator

            LDAA    #TONE_OUT           ;Output the FSK signal to the DUT
            ACALL   acADD_AMUX

            LDX     #4096               ;No attenuation of FSK signal
            JSR     _SET_DUT_FSKSIG
            JSR     _ENA_DUT_FSK2       ;Enable FSK_SIG2 analog port

            LDAA    #DUTIO1             ;Enable ring sound path
            JSR     _SET_DUTIO1_b       ;SELA=1,SELB=0
            LDAA    #DUTIO2^DUTIO3      ;Enable analog multiplexer #2
            JSR     _CLR_DUTIO1_b       ;-EN_MUX2=0

            LDAA    #DUTIO12^DUTIO13^DUTIO14
            JSR     _CLR_DUTIO1_b       ;Set ring volume to maximum value

; Check audio amplifier gain with frequency set to 1200 Hz

            LDAB    #$EF                ;Set PGA to gain of 10
            STAB    GAIN_VALUE          ;
            LDAA    #2                  ;Load the analog channel number
            STAA    CHAN_NUM            ;for reading audio amp gain value
            LDX     #100                ;Delay before reading
            ACALL   acDELAY_MSEC

            JSR     GET_AVG             ;Get audio amp gain reading at 500 Hz
            STX     SAVX1
            CPX     #$0150              ;Check if above maximum = 127mVrms
            BHI     ?amp0               ;Nominal voltage = 106mVrms
            CPX     #$00A0              ;Check if below minimum = 85mVrms
            BHI     ?amp1               ;
?amp0
            JSR     ERR_RPT             ;Report error
?amp1
; Check audio amplifier gain with frequency set to 1200 Hz

            INC     ERR_CODE            ;Set up error code
            LDX     #$03FF              ;Set frequency to 1200 Hz
            ACALL   acSET_PSWFSK_ZERO   ;
            LDX     #$03FF
            ACALL   acSET_PSWFSK_ONE
            LDX     #50                 ;Delay before reading
            ACALL   acDELAY_MSEC

            JSR     GET_AVG             ;Get audio amp gain reading at 1200 Hz
            STX     SAVX2
            CPX     #$0200              ;Check if above maximum = 136mVrms
            BHI     ?amp2               ;Nominal voltage = 113mVrms
            CPX     #$00A0              ;Check if below minimum = 90mVrms
            BHI     ?amp3               ;
?amp2
            JSR     ERR_RPT             ;Report error
?amp3
; Check audio amplifier gain with frequency at 2000 Hz

            INC     ERR_CODE            ;Set up error code
            LDX     #$06AA              ;Set frequency to 2000 Hz
            ACALL   acSET_PSWFSK_ZERO   ;
            LDX     #$06AA
            ACALL   acSET_PSWFSK_ONE
            LDX     #50                 ;Delay before reading
            ACALL   acDELAY_MSEC

            JSR     GET_AVG             ;Get audio amp gain reading at 2000 Hz
            STX     SAVX3
            CPX     #$0200              ;Check if above maximum = 138mVrms
            BHI     ?amp4               ;Nominal voltage = 115mVrms
            CPX     #$00A0              ;Check if below minimum = 92mVrms
            BHI     ?amp5               ;
?amp4
            JSR     ERR_RPT             ;Report error
?amp5
            INC     ERR_CODE            ;Set up error code
```

```
            JSR     _DIS_HKSW_RELAY     ;Go on-hook
            LDAA    #TONE_OUT           ;Turn off FSK signal
            ACALL   acRMV_AMUX          ;
            ACALL   acFSK_POWER_DN      ;Power down the FSK
            LDX     #50                 ;Delay 50mSec between tests
            ACALL   acDELAY_MSEC
            RTS
            EJECT
;-------------------------------------------------------------------------
RXD_GAIN_TST:
;  Check handset receive gain of the speech network
;  I/P: None
;  O/P: None
;
            LDAB    #RXD_GAIN_TST_NUM   ;Set to the correct test number
            STAB    TEST_NUM            ;for the err_rpt routine.
            LDAA    #1                  ;reinitialize for each test (err_rpt)
            STAA    ERR_CODE            ;Set up error code

            LDAB    #$02                ;Display line number 3
            LDX     #TEST_NUM_4
            JSR     LCD_SCREEN_1

            JSR     _ENA_HKSW_RELAY     ;Go off-hook

;            LDX     #$0400              ;
;            STX     FSK_ATT1
;            STX     FSK_ATT2
;            LDX     #$01AB              ;
;            STX     FSK_FRQ1
;            STX     FSK_FRQ2            ;
;            JSR     INIT_FSK            ;

            CLR     RMS_DC_FLAG         ;Set up for RMS measurement
            LDX     #$0400              ;Set up FSK attenuator #'0
            ACALL   acSET_FSK0_ATT      ;With load, RxD gain should be ~100mVrms
            LDX     #$0400              ;
            ACALL   acSET_FSK1_ATT      ;Set up FSK attenuator #'1
            LDX     #$01AB              ;
            ACALL   acSET_PSWFSK_ONE    ;
            LDX     #$01AB
            ACALL   acSET_PSWFSK_ZERO
            ACALL   acFSK_POWER_UP

            LDAA    #TONE_OUT
            ACALL   acADD_AMUX

            LDAB    #$EF                ;Set PGA to gain of 10
            STAB    GAIN_VALUE          ;
            LDAA    #1                  ;Load the analog channel number
            STAA    CHAN_NUM            ;
            LDX     #500                ;Delay before reading
            ACALL   acDELAY_MSEC

            JSR     GET_AVG             ;Get RxD gain reading at 500 Hz
            STX     SAVX1
            CPX     #$0150              ;Check if above maximum = 127mVrms
            BHI     ?rxd2               ;Nominal voltage = 106mVrms
            CPX     #$00A0              ;Check if below minimum = 85mVrms
            BHI     ?rxd3               ;
?rxd2
            JSR     ERR_RPT             ;Report error
?rxd3
; Check Receive gain with frequency set to 1200 Hz
            INC     ERR_CODE            ;Set up error code
            LDX     #$03FF              ;Set frequency to 1200 Hz
            ACALL   acSET_PSWFSK_ZERO   ;
            LDX     #$03FF
            ACALL   acSET_PSWFSK_ONE
            LDX     #50                 ;Delay before reading
            ACALL   acDELAY_MSEC

            JSR     GET_AVG             ;Get RxD gain reading at 1200 Hz
            STX     SAVX2
            CPX     #$0200              ;Check if above maximum = 136mVrms
            BHI     ?rxd4               ;Nominal voltage = 113mVrms
            CPX     #$00A0              ;Check if below minimum = 90mVrms
            BHI     ?rxd5               ;
?rxd4
```

```
            JSR     ERR_RPT             ;Report error
?rxd5
; Check Receive gain with frequency at 2000 Hz
            INC     ERR_CODE            ;Set up error code
            LDX     #$06AA              ;Set frequency to 2000 Hz
            ACALL   acSET_PSWFSK_ZERO   ;
            LDX     #$06AA
            ACALL   acSET_PSWFSK_ONE
            LDX     #50                 ;Delay before reading
            ACALL   acDELAY_MSEC

            JSR     GET_AVG             ;Get RxD gain reading at 2000 Hz
            STX     SAVX3
            CPX     #$0200              ;Check if above maximum = 138mVrms
            BHI     ?rxd6               ;Nominal voltage = 115mVrms
            CPX     #$00A0              ;Check if below minimum = 92mVrms
            BHI     ?rxd7               ;
?rxd6
            JSR     ERR_RPT             ;Report error
?rxd7
            JSR     _DIS_HKSW_RELAY     ;Go on-hook
            LDAA    #TONE_OUT           ;Turn off FSK signal
            ACALL   acRMV_AMUX          ;
            ACALL   acFSK_POWER_DN      ;Power down the FSK
            LDX     #50                 ;Delay 50mSec between tests
            ACALL   acDELAY_MSEC
            RTS
            EJECT
;-------------------------------------------------------------------------
TXD_GAIN_TST:
;  Check handset transmit gain of the speech network
;  I/P: None
;  O/P: None
;
            LDAB    #TXD_GAIN_TST_NUM   ;Set to the correct test number
            STAB    TEST_NUM            ;for the err_rpt routine.
            LDAA    #1                  ;reinitialize for each test (err_rpt)
            STAA    ERR_CODE            ;Set up error code

            LDAB    #$02                ;Display line number 3
            LDX     #TEST_NUM_5
            JSR     LCD_SCREEN_1

            JSR     _ENA_HKSW_RELAY     ;Go off-hook
            CLR     RMS_DC_FLAG         ;Set up for RMS measurement
            LDX     #$01F4              ;Set up FSK attenuator #'0 for 1.0Vp-p
            ACALL   acSET_FSK0_ATT      ;
            LDX     #$01F4              ;Set up FSK attenuator #'1 for 1.0Vp-p
            ACALL   acSET_FSK1_ATT      ;
            LDX     #$01AB              ;Set frequency = 500 Hz
            ACALL   acSET_PSWFSK_ONE    ;
            LDX     #$01AB
            ACALL   acSET_PSWFSK_ZERO
            ACALL   acFSK_POWER_UP      ;Power up the FSK generator
            LDAA    #TONE_OUT           ;Switch FSK output to the line
            ACALL   acADD_AMUX

            LDAA    #$EF                ;Set PGA to gain of 10
            STAA    GAIN_VALUE          ;
            LDAA    #0                  ;Load the analog channel number
            STAA    CHAN_NUM            ;
            LDX     #50                 ;Delay before reading
            ACALL   acDELAY_MSEC

; Attenuate signal to 5.1 mVrms here
            JSR     _ENA_DUT_FSK1       ;Enable FSK_SIG1 output port
            LDX     #$0053              ;VALUE TO BE SENT TO THE DAC
                                        ;TO CREATE 5.1V RMS @ U12 DAC8043GP.
                                        ;THE MSN IS IGNORED, ONLY USE BITS 0-11
                                        ; 11 IS THE MSb 0 IS THE LSb
;           JSR     SET_DUT_FSKSIG      ;Work around subroutine to set the DAC
            JSR     _SET_DUT_FSKSIG
            LDX     #50                 ;Delay before reading
            ACALL   acDELAY_MSEC
?txd1
; Check Transmit gain with frequency at 500 Hz
            JSR     GET_AVG             ;Get DTMF gain reading at 500 Hz
            STX     SAVX1               ;Save contents
            CPX     #$0150              ;Check if above maximum = 109mVrms
```

```
            BHI     ?txd2               ;Nominal voltage = 91mVrms
            CPX     #$0095              ;Check if below minimum = 73mVrms
            BHI     ?txd3               ;Min lowered to 50mV for testing
?txd2
            JSR     ERR_RPT             ;Report error
?txd3
; Check Transmit gain with frequency at 1200 Hz
            INC     ERR_CODE            ;Set up error code
            LDX     #$03AA              ;Set frequency to 1200 Hz
            ACALL   acSET_PSWFSK_ZERO   ;
            LDX     #$03AA
            ACALL   acSET_PSWFSK_ONE
            LDX     #50                 ;Delay before reading
            ACALL   acDELAY_MSEC

            JSR     GET_AVG             ;Get DTMF gain reading at 1200 Hz
            STX     SAVX2               ;Save contents
            CPX     #$01A0              ;Check if above maximum = 120mVrms
            BHI     ?txd4               ;Nominal voltage = 100mVrms
            CPX     #$0095              ;Check if below minimum = 80mVrms
            BHI     ?txd5               ;
?txd4
            JSR     ERR_RPT             ;Report error
?txd5
; Check Transmit gain with frequency at 2000 Hz
            INC     ERR_CODE            ;Set up error code
            LDX     #$06AA              ;Set frequency to 2000 Hz
            ACALL   acSET_PSWFSK_ZERO   ;
            LDX     #$06AA
            ACALL   acSET_PSWFSK_ONE
            LDX     #50                 ;Delay before reading
            ACALL   acDELAY_MSEC

            JSR     GET_AVG             ;Get DTMF gain reading at 2000 Hz
            STX     SAVX3               ;Save contents
            CPX     #$0200              ;Check if above maximum = 154mVrms
            BHI     ?txd6               ;Nominal voltage = 128mVrms
            CPX     #$0100              ;Check if below minimum = 102mVrms
            BHI     ?txd7               ;
?txd6
            JSR     ERR_RPT             ;Report error
?txd7
            LDAA    #TONE_OUT           ;Turn off FSK signal
            ACALL   acRMV_AMUX          ;
            ACALL   acFSK_POWER_DN      ;Power down the FSK
            JSR     _DIS_DUT_FSK1       ;Disable FSK_SIG1 output port
            JSR     _DIS_HKSW_RELAY     ;Go on-hook
            LDX     #50                 ;Delay 50mSec between tests
            ACALL   acDELAY_MSEC
            RTS
            EJECT
;-------------------------------------------------------------------------
RXD_GAIN_SP_TST:
;  Check receive gain of the speakerphone network
;  I/P: None
;  O/P: None
;
            LDAB    #RXD_SP_TST_NUM     ;Set to the correct test number
            STAB    TEST_NUM            ;for the err_rpt routine.
            LDAA    #1                  ;reinitialize for each test (err_rpt)
            STAA    ERR_CODE            ;Set up error code

            LDAB    #$02                ;Display line number 3
            LDX     #TEST_NUM_4
            JSR     LCD_SCREEN_1

            JSR     _DIS_HKSW_RELAY     ;Go on-hook

            CLR     RMS_DC_FLAG         ;Set up for RMS measurement
            SFSK_1  #$400,#$1AB         ;Set FSK
;            LDX     #$0400              ;Set up FSK attenuator #'0
;            ACALL   acSET_FSK0_ATT      ;With load, RxD gain should be ~100mVrms
;            LDX     #$0400              ;
;            ACALL   acSET_FSK1_ATT      ;Set up FSK attenuator #'1
;            LDX     #$01AB              ;
;            ACALL   acSET_PSWFSK_ONE    ;
;            LDX     #$01AB
            ACALL   acSET_PSWFSK_ZERO
            ACALL   acFSK_POWER_UP
```

```
            LDAA    #TONE_OUT
            ACALL   acADD_AMUX

            LDAB    #$EF                ;Set PGA to gain of 10
            STAB    GAIN_VALUE          ;
            LDAA    #1                  ;Load the analog channel number
            STAA    CHAN_NUM            ;
            LDX     #500                ;Delay before reading
            ACALL   acDELAY_MSEC

            JSR     GET_AVG             ;Get RxD gain reading at 500 Hz
            STX     SAVX1
            CPX     #$0150              ;Check if above maximum = 127mVrms
            BHI     ?rxd2               ;Nominal voltage = 106mVrms
            CPX     #$00A0              ;Check if below minimum = 85mVrms
            BHI     ?rxd3               ;
?rxd2
            JSR     ERR_RPT             ;Report error
?rxd3
; Check Receive gain with frequency set to 1200 Hz
            INC     ERR_CODE            ;Set up error code
            LDX     #$03FF              ;Set frequency to 1200 Hz
            ACALL   acSET_PSWFSK_ZERO   ;
            LDX     #$03FF
            ACALL   acSET_PSWFSK_ONE
            LDX     #50                 ;Delay before reading
            ACALL   acDELAY_MSEC

            JSR     GET_AVG             ;Get RxD gain reading at 1200 Hz
            STX     SAVX2
            CPX     #$0200              ;Check if above maximum = 136mVrms
            BHI     ?rxd4               ;Nominal voltage = 113mVrms
            CPX     #$00A0              ;Check if below minimum = 90mVrms
            BHI     ?rxd5               ;
?rxd4
            JSR     ERR_RPT             ;Report error
?rxd5
; Check Receive gain with frequency at 2000 Hz
            INC     ERR_CODE            ;Set up error code
            LDX     #$06AA              ;Set frequency to 2000 Hz
            ACALL   acSET_PSWFSK_ZERO   ;
            LDX     #$06AA
            ACALL   acSET_PSWFSK_ONE
            LDX     #50                 ;Delay before reading
            ACALL   acDELAY_MSEC

            JSR     GET_AVG             ;Get RxD gain reading at 2000 Hz
            STX     SAVX3
            CPX     #$0200              ;Check if above maximum = 138mVrms
            BHI     ?rxd6               ;Nominal voltage = 115mVrms
            CPX     #$00A0              ;Check if below minimum = 92mVrms
            BHI     ?rxd7               ;
?rxd6
            JSR     ERR_RPT             ;Report error
?rxd7
            JSR     _DIS_HKSW_RELAY     ;Go on-hook
            LDAA    #TONE_OUT           ;Turn off FSK signal
            ACALL   acRMV_AMUX          ;
            ACALL   acFSK_POWER_DN      ;Power down the FSK
            LDX     #50                 ;Delay 50mSec between tests
            ACALL   acDELAY_MSEC
            RTS
            EJECT
;-------------------------------------------------------------------------------
TXD_GAIN_SP_TST:
;  Check transmit gain of the speakerphone network
;  I/P: None
;  O/P: None
;
            LDAB    #TXD_SP_TST_NUM     ;Set to the correct test number
            STAB    TEST_NUM            ;for the err_rpt routine.
            LDAA    #1                  ;reinitialize for each test (err_rpt)
            STAA    ERR_CODE            ;Set up error code

            LDAB    #$02                ;Display line number 3
            LDX     #TEST_NUM_5
            JSR     LCD_SCREEN_1
```

```
            JSR     _ENA_HKSW_RELAY     ;Go off-hook
            CLR     RMS_DC_FLAG         ;Set up for RMS measurement
            LDX     #$01F4              ;Set up FSK attenuator #'0 for 1.0Vp-p
            ACALL   acSET_FSK0_ATT      ;
            LDX     #$01F4              ;Set up FSK attenuator #'1 for 1.0Vp-p
            ACALL   acSET_FSK1_ATT      ;
            LDX     #$01AB              ;Set frequency = 500 Hz
            ACALL   acSET_PSWFSK_ONE    ;
            LDX     #$01AB
            ACALL   acSET_PSWFSK_ZERO
            ACALL   acFSK_POWER_UP      ;Power up the FSK generator
            LDAA    #TONE_OUT           ;Switch FSK output to the line
            ACALL   acADD_AMUX

            LDAA    #$EF                ;Set PGA to gain of 10
            STAA    GAIN_VALUE          ;
            LDAA    #0                  ;Load the analog channel number
            STAA    CHAN_NUM            ;
            LDX     #50                 ;Delay before reading
            ACALL   acDELAY_MSEC

; Attenuate signal to 5.1 mVrms here
            JSR     _ENA_DUT_FSK1       ;Enable FSK_SIG1 output port
            LDX     #$0053              ;VALUE TO BE SENT TO THE DAC
                                        ;TO CREATE 5.1V RMS @ U12 DAC8043GP.
                                        ;THE MSN IS IGNORED, ONLY USE BITS 0-11
                                        ; 11 IS THE MSb 0 IS THE LSb
;           JSR     SET_DUT_FSKSIG      ;Work around subroutine to set the DAC
            JSR     _SET_DUT_FSKSIG
            LDX     #50                 ;Delay before reading
            ACALL   acDELAY_MSEC
?txd1
; Check Transmit gain with frequency at 500 Hz
            JSR     GET_AVG             ;Get DTMF gain reading at 500 Hz
            STX     SAVX1               ;Save contents
            CPX     #$0150              ;Check if above maximum = 109mVrms
            BHI     ?txd2               ;Nominal voltage = 91mVrms
            CPX     #$0095              ;Check if below minimum = 73mVrms
            BHI     ?txd3               ;Min lowered to 50mV for testing
?txd2
            JSR     ERR_RPT             ;Report error
?txd3
; Check Transmit gain with frequency at 1200 Hz
            INC     ERR_CODE            ;Set up error code
            LDX     #$03AA              ;Set frequency to 1200 Hz
            ACALL   acSET_PSWFSK_ZERO   ;
            LDX     #$03AA
            ACALL   acSET_PSWFSK_ONE
            LDX     #50                 ;Delay before reading
            ACALL   acDELAY_MSEC

            JSR     GET_AVG             ;Get DTMF gain reading at 1200 Hz
            STX     SAVX2               ;Save contents
            CPX     #$01A0              ;Check if above maximum = 120mVrms
            BHI     ?txd4               ;Nominal voltage = 100mVrms
            CPX     #$0095              ;Check if below minimum = 80mVrms
            BHI     ?txd5               ;
?txd4
            JSR     ERR_RPT             ;Report error
?txd5
; Check Transmit gain with frequency at 2000 Hz
            INC     ERR_CODE            ;Set up error code
            LDX     #$06AA              ;Set frequency to 2000 Hz
            ACALL   acSET_PSWFSK_ZERO   ;
            LDX     #$06AA
            ACALL   acSET_PSWFSK_ONE
            LDX     #50                 ;Delay before reading
            ACALL   acDELAY_MSEC

            JSR     GET_AVG             ;Get DTMF gain reading at 2000 Hz
            STX     SAVX3               ;Save contents
            CPX     #$0200              ;Check if above maximum = 154mVrms
            BHI     ?txd6               ;Nominal voltage = 128mVrms
            CPX     #$0100              ;Check if below minimum = 102mVrms
            BHI     ?txd7               ;
?txd6
            JSR     ERR_RPT             ;Report error
?txd7
            LDAA    #TONE_OUT           ;Turn off FSK signal
```

```
            ACALL   acRMV_AMUX          ;
            ACALL   acFSK_POWER_DN      ;Power down the FSK
            JSR     _DIS_DUT_FSK1       ;Disable FSK_SIG1 output port
            JSR     _DIS_HKSW_RELAY     ;Go on-hook
            LDX     #50                 ;Delay 50mSec between tests
            ACALL   acDELAY_MSEC
            RTS
            EJECT
;-------------------------------------------------------------------------
RXD_SWITCH_TST:
;  Check the transmit-to-receive switching time of the speakerphone network
;  I/P: None
;  O/P: None
;
            LDAB    #TXD_SWITCH_TST_NUM ;Set to the correct test number
            STAB    TEST_NUM            ;for the err_rpt routine.
            LDAA    #1                  ;reinitialize for each test (err_rpt)
            STAA    ERR_CODE            ;Set up error code

            LDAB    #$02                ;Display line number 3
            LDX     #TEST_NUM_5
            JSR     LCD_SCREEN_1

            JSR     _ENA_HKSW_RELAY     ;Go off-hook
            CLR     RMS_DC_FLAG         ;Set up for RMS measurement
            LDX     #$03FF              ;Set up FSK attenuator #'0 for 1.0Vp-p
            ACALL   acSET_FSK0_ATT      ;
            LDX     #$03FF              ;Set up FSK attenuator #'1 for 1.0Vp-p
            ACALL   acSET_FSK1_ATT      ;
            LDX     #$01AB              ;Set frequency = 1200 Hz
            ACALL   acSET_PSWFSK_ONE    ;
            LDX     #$01AB
            ACALL   acSET_PSWFSK_ZERO
            ACALL   acFSK_POWER_UP      ;Power up the FSK generator
            LDAA    #TONE_OUT           ;Switch FSK output to the line
            ACALL   acADD_AMUX

            LDAA    #TONE_OUT           ;Turn off FSK signal
            ACALL   acRMV_AMUX          ;
            ACALL   acFSK_POWER_DN      ;Power down the FSK
            RTS
            EJECT
;-------------------------------------------------------------------------
TXD_SWITCH_TST:
;  Check the receive-to-transmit switching time of the speakerphone network
;  I/P: None
;  O/P: None
;
            LDAB    #TXD_SWITCH_TST_NUM ;Set to the correct test number
            STAB    TEST_NUM            ;for the err_rpt routine.
            LDAA    #1                  ;reinitialize for each test (err_rpt)
            STAA    ERR_CODE            ;Set up error code

            LDAB    #$02                ;Display line number 3
            LDX     #TEST_NUM_5
            JSR     LCD_SCREEN_1

            JSR     _ENA_HKSW_RELAY     ;Go off-hook
            CLR     RMS_DC_FLAG         ;Set up for RMS measurement
            LDX     #$03FF              ;Set up FSK attenuator #'0 for 1.0Vp-p
            ACALL   acSET_FSK0_ATT      ;
            LDX     #$03FF              ;Set up FSK attenuator #'1 for 1.0Vp-p
            ACALL   acSET_FSK1_ATT      ;
            LDX     #$01AB              ;Set frequency = 1200 Hz
            ACALL   acSET_PSWFSK_ONE    ;
            LDX     #$01AB
            ACALL   acSET_PSWFSK_ZERO
            ACALL   acFSK_POWER_UP      ;Power up the FSK generator
            LDAA    #TONE_OUT           ;Switch FSK output to the line
            ACALL   acADD_AMUX

            LDAA    #TONE_OUT           ;Turn off FSK signal
            ACALL   acRMV_AMUX          ;
            ACALL   acFSK_POWER_DN      ;Power down the FSK
            RTS
            EJECT
;-------------------------------------------------------------------------
DTMF_GAIN_TST:
;  Check the DTMF gain of the speech network
```

```
            JSR     LED_ADJ             ;Turn on the appropriate LED
            JSR     _DUT_START_PB       ;START pushbutton?
            BNE     ?dtmf8              ;If yes, exit; else, keep looping
; This section outputs a 941 Hz low band 'D tone from the STEALTH
?dtmf2
            LDX     #$0322              ;Set frequency = 941 Hz
            ACALL   acSET_PSWFSK_ONE    ;
            LDX     #$0322
            ACALL   acSET_PSWFSK_ZERO

            LDX     #50                 ;Delay 5mSec for line voltage to set
            ACALL   acDELAY_MSEC
; Check the gain difference between the 'D tone low and high bands
DTMF_D2_GOOD:
            CLR     RMS_DC_FLAG         ;Set up for RMS measurement
            JSR     GET_AVG             ;Get DTMF gain reading at 1200 Hz
            STX     SAVX2
            CPX     #$037B              ;Check if above maximum
            BHI     ?dtmf3              ;
            CPX     #$0366              ;Check if in range
            BHI     ?dtmf5              ;
; Low band tone should track with high band tone within 2db else error
?dtmf3
            LDAB    #$01                ;Display line number 2
            LDX     #GAIN_ERR_MSG
            JSR     LCD_SCREEN_1        ;
            LDAB    #$02                ;Display line number 3
            LDX     #FUNCT_PB_MSG
            JSR     LCD_SCREEN_1
            LDAB    #$03                ;Display line number 4
            LDX     #START_PB_MSG
            JSR     LCD_SCREEN_1

            LDAA    #0                  ;Turn off all LEDs and try again
            JSR     _DUT_LED_OFF
?dtmf4
            JSR     _DUT_START_PB       ;Make sure pushbutton has been released
            BEQ     ?dtmf4
dtmf8
            JSR     _DUT_START_PB       ;START pushbutton?
            BEQ     dtmf9               ;If yes, exit; else, keep looping
            ACALL   acGET_PB            ;Push FUNCTION key on CGT to repeat test
            BEQ     dtmf8               ;Else keep looking for a key
            CMPA    #F1_PB              ;If FUNCTION key, then repeat test
            BNE     dtmf8               ;Else, keep looping
            JMP     DTMF_GAIN_TST       ;Repeat the test
dtmf9
            JSR     ERR_RPT             ;Else, report DTMF trim pot gain error
dtmf5
; While MUTE.DIAL is active check the signal across the earpiece.
            LDX     #50                 ;Delay 50mSec for line voltage to set
            ACALL   acDELAY_MSEC
            INC     ERR_CODE            ;Set up error code
            LDAA    #ANA_RMS2           ;No signal should be present across
            STAA    CHAN_NUM            ;the earpeice with MUTE.DIAL active
            LDAA    #DUT_RMS_GAIN_1     ;Set PGA to gain of 1
            STAA    GAIN_VALUE          ;
            JSR     GET_AVG             ;
            BMI     dtmf6               ;Make sure not reading a negative value
            CPX     #100                ;Reading should be close to zero & positive
            BLS     dtmf6               ;
            JSR     ERR_RPT             ;Report earpiece mute error
dtmf6
;           LDAA    #1                  ;Set bit 0
;           JSR     _CLR_DUTIO2_b       ;Clear CASDET signal bit 0
?dtmf7
;           LDAA    #10000000B          ;Set bit 7 = 0, CASDET inactive low
;           JSR     _CLR_DUT_LOP_b      ;
            JSR     _DIS_DUTIO2         ;Disable DUTIO2 port
            LDAA    #00000001B          ;Set bit 0 = 0
            JSR     _CLR_DUTIO2_b       ;Force CASDET inactive low
            CLRA                        ;Turn off the adjustment LED
            JSR     _DUT_LED_OFF        ;
            JSR     _DIS_HKSW_RELAY     ;Go on-hook
            LDAA    #TONE_OUT           ;Turn off FSK signal
            ACALL   acRMV_AMUX          ;
            ACALL   acFSK_POWER_DN      ;Power down the FSK
            JSR     _DIS_DUT_FSK1       ;Disable the DTMF (FSK_SIG1) analog port
            LDX     #50                 ;Delay 50mSec between tests
```

```
            ACALL   acDELAY_MSEC
            RTS
            EJECT
;-------------------------------------------------------------------------
FSK_GAIN_TST:
;  Check the FSK circuit gain
;  Calibrate the XR2211 duty cyle trim pot
;  I/P: None
;  O/P: None
;
            LDAB    #FSK_GAIN_TST_NUM   ;Set to the correct test number
            STAB    TEST_NUM            ;for the err_rpt routine.
            LDAA    #1                  ;reinitialize for each test (err_rpt)
            STAA    ERR_CODE            ;Set up error code


            LDAB    #$02                ;Display line number 3
            LDX     #TEST_NUM_9
            JSR     LCD_SCREEN_1

            LDAA    #DUT_LOP_1          ;Enable -FSK_EN
            JSR     _CLR_DUT_LOP_b      ;Turn on voltage to the FSK circuit
            JSR     _ENA_DUT_LOP        ;Enable the logical output port
            JSR     _DIS_HKSW_RELAY     ;Go on-hook
            LDX     #50                 ;Delay 50mSec for things to settle
            ACALL   acDELAY_MSEC

            CLR     RMS_DC_FLAG         ;Set up for RMS measurement
; Check the status of the -CD signal, should be inactive high if the FSK
; signal is not present.
            CLRA                        ;Set up for lower input port
            LDAA    #CD_BIT             ;Mask off the -CD bit
            JSR     _TST_DUT_LIPORT     ;Read the port
            BNE     ?fsk0               ;Should be inactive high
            JSR     ERR_RPT             ;If not report an error
?fsk0
            INC     ERR_CODE            ;Set up error code
            LDAA    #FSK_ADJ_LED        ;Turn on the FSK duty cycle adj LED
            JSR     _DUT_LED_ON         ;
            LDX     #$00E2              ;Set up FSK attenuator #'0 for 30mVp-p
            ACALL   acSET_FSK0_ATT      ;
            LDX     #$00E2              ;Set up FSK attenuator #'1 for 30mVp-p
            ACALL   acSET_FSK1_ATT      ;
            LDX     #$03FF              ;Set Mark frequency to 1200 Hz
            ACALL   acSET_PSWFSK_ZERO   ;
            LDX     #$0755              ;Set Space frequency to 2200 Hz
            ACALL   acSET_PSWFSK_ONE    ;
            ACALL   acFSK_POWER_UP      ;Power up the FSK
            LDAA    #TONE_OUT           ;Turn on the FSK signal
            ACALL   acADD_AMUX          ;

            LDX     #100                ;Delay for FSK to settle
            ACALL   acDELAY_MSEC        ;

; Check the gain of the FSK signal prior to the XR2211
            LDAA    #$EF                ;Set PGA to gain of 10
            STAA    GAIN_VALUE          ;
            LDAA    #6                  ;Load the analog channel number
            STAA    CHAN_NUM            ;
            JSR     GET_AVG             ;Get the average reading
            STX     SAVX1
            CPX     #$0050              ;Check if above maximum > 39mVrms
            BHI     ?fsk1               ;Check if signal present & > min limit
            LDAA    FSK_REG             ;
            ORAA    #2                  ;Set FSK gain error flag
            STAA    FSK_REG
            JSR     ERR_RPT             ;Report FSK gain error
?fsk1
; Check the status of the -CD signal, should be active low if the FSK signal
; is present with sufficient gain.
            INC     ERR_CODE            ;Set up error code
            CLRA                        ;Set up for lower input port
            LDAA    #CD_BIT             ;Mask off the -CD bit
            JSR     _TST_DUT_LIPORT     ;Read the port
            BEQ     ?fsk2               ;Should be active low
            JSR     ERR_RPT             ;If not report a-CD not active error
?fsk2
; Check the gain of the TONE.OUT signal prior to the filter stages
            INC     ERR_CODE            ;Set up error code
```

```
            LDAA    #7                  ;Channel number for TONE.OUT gain
            STAA    CHAN_NUM            ;
            JSR     GET_AVG             ;Get the average reading
            STX     SAVX1
            CPX     #$0050              ;Check if above maximum > 39mVrms
            BHI     ?fsk3               ;Check if signal present & > min limit
            LDAA    FSK_REG             ;
            ORAA    #1                  ;Set TONE.OUT gain error flag
            STAA    FSK_REG
            JSR     ERR_RPT             ;Report TONE.OUT gain error
?fsk3
; Adjust the FSK duty cycle trim pot for 50%, +/-2% and push the start
; button when finished
;           INC     ERR_CODE            ;Set up error code
;           STX     MAX                 ;Save maximum setting
;           STY     MIN                 ;Save minimum setting
;           JSR     LED_ADJ             ;Turn on the appropriate LED

            LDAB    #$01                ;Display Test 9 in progress message
            LDX     #TIP09_MSG          ;on line number 2
            JSR     LCD_SCREEN_1
            LDAB    #$02                ;Display FSK adjustment message
            LDX     #ADJ_FSK_MSG        ;on line number 3
            JSR     LCD_SCREEN_1
            LDAB    #$03                ;Display line number 4
            LDX     #START_PB_MSG       ;
            JSR     LCD_SCREEN_1
?fsk4
            JSR     _DUT_START_PB       ;START pushbutton?
            BNE     ?fsk4               ;If yes, exit; else, keep looping

;           LDAA    ERR_FLAG            ;Check for adjustment error
;           BEQ     ?fsk5               ;Exit if no error
;           JSR     ERR_RPT             ;Else, report FSK adjustment error
?fsk5
            LDAA    #TONE_OUT           ;Turn off FSK signal
            ACALL   acRMV_AMUX          ;
            ACALL   acFSK_POWER_DN      ;Power down the FSK
            LDAA    #DUT_LOP_1          ;Disable -FSK_EN
            JSR     _SET_DUT_LOP_b      ;Turn off voltage to the FSK circuit
            JSR     _ENA_DUT_LOP        ;Enable the logical output port

            LDAA    #INH_MUX            ;Inhibit analog mux channels
            JSR     _SET_DUTIO1_b       ;
            CLRA                        ;Turn off the adjustment LED
            JSR     _DUT_LED_OFF        ;
            LDX     #50                 ;Delay 50mSec between tests
            ACALL   acDELAY_MSEC
            RTS
            EJECT
;-------------------------------------------------------------------------

CAL_2130_TST:
;  Calibrate the 1st and 2nd 2130 filter stages
;  I/P: None
;  O/P: None
;
            LDAB    #CAL_2130F1_TST_NUM ;Set to the correct test number
            STAB    TEST_NUM            ;for the err_rpt routine.
            LDAA    #1                  ;reinitialize for each test (err_rpt)
            STAA    ERR_CODE            ;Set up error code

            LDAB    #$02                ;Display line number 3
            LDX     #TEST_NUM_10
            JSR     LCD_SCREEN_1

            JSR     _ENA_HKSW_RELAY     ;Go off-hook
            LDX     #100                ;Delay 100mSec for settling
            ACALL   acDELAY_MSEC

            JSR     _DIS_HKSW_RELAY     ;Go on-hook
            LDX     #100                ;Delay 100mSec for settling
            ACALL   acDELAY_MSEC

            LDAA    #INH_MUX^SELA^SELB  ;Select channel 0
            JSR     _CLR_DUTIO1_b       ;INH, SELA, SELB = 0
            LDAA    #F2130_ADJ_LED1     ;Turn on the 1st stage adj LED
            JSR     _DUT_LED_ON         ;
```

```
            LDX     #$0071              ;Set up FSK attenuator #'0 for 61.5mVp-p
            ACALL   acSET_FSK0_ATT      ;
            LDX     #$0071              ;Set up FSK attenuator #'1 for 61.5mVp-p
            ACALL   acSET_FSK1_ATT      ;
            LDX     #$0719              ;Set first frequency to 2130 Hz
            ACALL   acSET_PSWFSK_ZERO   ;
            LDX     #$0719              ;Set second frequency to 2130 Hz
            ACALL   acSET_PSWFSK_ONE    ;
            ACALL   acFSK_POWER_UP      ;Power up the FSK generator
            LDAA    #TONE_OUT           ;Switch in the FSK signal
            ACALL   acADD_AMUX          ;

            LDAB    #$01                ;Display Test 10 in progress message
            LDX     #TIP10_MSG          ;on line number 2
            JSR     LCD_SCREEN_1
            LDAB    #$02                ;Display line number 3
            LDX     #ADJ_2130F1_MSG
            JSR     LCD_SCREEN_1
            LDAB    #$03                ;Display line number 4
            LDX     #START_PB_MSG
            JSR     LCD_SCREEN_1
?pb_rel1
            JSR     _DUT_START_PB       ;Make sure pushbutton has been released
            BEQ     ?pb_rel1
?cal0
; Adjust the 2130 first stage filter stage trim Pot & when finished,
; push the START pushbutton on the fixture to continue the test.
            JSR     _DUT_START_PB       ;START pushbutton?
            BNE     ?cal0               ;If yes, exit; else, keep looping

;  Calibrate the 2nd 2130 filter stage

            LDAA    #SELA               ;Select channel 1 of the mux
            JSR     _SET_DUTIO1_b       ;SELA = 1
            LDAA    #SELB               ;SELB = 0
            JSR     _CLR_DUTIO1_b       ;INH  = 0

            LDAA    #F2130_ADJ_LED1     ;Turn off the 1st stage adj LED
            JSR     _DUT_LED_OFF        ;
            LDAA    #F2130_ADJ_LED2     ;Turn on the 2nd stage adj LED
            JSR     _DUT_LED_ON         ;

            LDX     #$0719              ;Set first frequency to 2130 Hz
            ACALL   acSET_PSWFSK_ZERO   ;
            LDX     #$0719              ;Set second frequency to 2130 Hz
            ACALL   acSET_PSWFSK_ONE    ;

            LDAB    #$01                ;Display line number 2
            LDX     #TIP10b_MSG
            JSR     LCD_SCREEN_1
            LDAB    #$02                ;Display line number 3
            LDX     #ADJ_2130F2_MSG
            JSR     LCD_SCREEN_1
            LDAB    #$03                ;Display line number 4
            LDX     #START_PB_MSG
            JSR     LCD_SCREEN_1
?pb_rel2
            JSR     _DUT_START_PB       ;Make sure pushbutton has been released
            BEQ     ?pb_rel2
?cal1
; Adjust the 2130 second stage filter stage trim Pot & when finished,
; push the START pushbutton on the fixture to continue the test.
            JSR     _DUT_START_PB       ;START pushbutton?
            BNE     ?cal1               ;If yes, exit; else, keep looping

            CLRA                        ;Turn off the adjustment LED
            JSR     _DUT_LED_OFF        ;
            LDAA    #DUT_RMS_GAIN_10    ;Set PGA to gain of 10
            STAA    GAIN_VALUE          ;
            LDAA    #ANA_RMS5           ;Set up analog channel #5
            STAA    CHAN_NUM            ;
            CLR     RMS_DC_FLAG         ;Set up for RMS measurement

; Sweep the 2130 filter and check the band pass limits
; Read the peak frequency value at 2130 Hz
            JSR     GET_AVG             ;Get nominal reading
            STX     FILT_NOM            ;and save it
            CPX     #$00C8              ;If low gain exit with error
            BHI     ?gain_ok            ;
```

```
            JMP     cal_2130_err        ;
?gain_ok
            LDX     #$0711              ;Set first frequency to 2130 Hz
            ACALL   acSET_PSWFSK_ZERO   ;
            LDX     #$0711              ;Set second frequency to 2130 Hz
            ACALL   acSET_PSWFSK_ONE    ;
            LDAA    #DUTIO01            ;Select channel 1 of the mux
            JSR     _SET_DUTIO1_b       ;SELA = 1
            LDX     #50                 ;Delay before reading
            ACALL   acDELAY_MSEC

; Read the low frequency value at 2120 Hz
            JSR     GET_AVG             ;Get average reading
            STX     FILT_LOW            ;Save low frequency value

; Read the high frequency value at 2140 Hz
            LDX     #$0722              ;Set first frequency to 2140 Hz
            ACALL   acSET_PSWFSK_ZERO   ;
            LDX     #$0722              ;Set second frequency to 2140 Hz
            ACALL   acSET_PSWFSK_ONE    ;
            LDX     #50                 ;Delay before reading
            ACALL   acDELAY_MSEC

            JSR     GET_AVG             ;Get average reading
            STX     FILT_HIGH           ;Save high frequency value

; The nominal value of the filter adjustment should be the peak reading
; and be greater than both the low value and the high value.
            LDX     FILT_NOM
            CPX     FILT_LOW            ;Compare nominal to low
            BHI     ?swp0               ;Should be higher
            BRA     cal_2130_err        ;If lower, report adjustment error
?swp0
            CPX     FILT_HIGH           ;Compare nominal to high
            BHI     swp5                ;Should be higher
cal_2130_err
            LDAB    #$01                ;Display line number 2
            LDX     #GAIN_ERR_MSG
            JSR     LCD_SCREEN_1
            LDAB    #$02                ;Display line number 3
            LDX     #FUNCT_PB_MSG
            JSR     LCD_SCREEN_1
            LDAB    #$03                ;Display line number 4
            LDX     #START_PB_MSG
            JSR     LCD_SCREEN_1
?pb_rel3
            JSR     _DUT_START_PB       ;Make sure pushbutton has been released
            BEQ     ?pb_rel3
swp2
            JSR     _DUT_START_PB       ;START pushbutton?
            BEQ     swp4                ;If yes, exit; else, keep looping
            ACALL   acGET_PB            ;Push FUNCTION key on CGT to repeat test
            BEQ     swp2                ;Else keep looking for a key
            CMPA    #F1_PB              ;If FUNCTION key, then repeat test
            BNE     swp2                ;Else, keep looping
            JMP     CAL_2130_TST        ;Repeat the test
swp4
            JSR     ERR_RPT             ;Report 2130 filter verify error
swp5
            LDAA    #TONE_OUT           ;Turn off FSK signal
            ACALL   acRMV_AMUX          ;
            ACALL   acFSK_POWER_DN      ;Power down the FSK
            LDAA    #DUTIO00            ;Disable pogo_if analog mux
            JSR     _CLR_DUTIO1_b       ;
            LDX     #50                 ;Delay 50mSec between tests
            ACALL   acDELAY_MSEC
            RTS
;-------------------------------------------------------------------------

CAL_2750_TST:
;  Calibrate the 2750 filter stage
;  I/P: None
;  O/P: None
;
            LDAB    #CAL_2750_TST_NUM   ;Set to the correct test number
            STAB    TEST_NUM            ;for the err_rpt routine.
            LDAA    #1                  ;reinitialize for each test (err_rpt)
            STAA    ERR_CODE            ;Set up error code
```

```
            LDX     #TEST_NUM_11
            LDAB    #$02                ;Display line number 2
            JSR     LCD_SCREEN_1

            JSR     _ENA_HKSW_RELAY     ;Go off-hook
            LDX     #100                ;Delay 100mSec for settling
            ACALL   acDELAY_MSEC

            JSR     _DIS_HKSW_RELAY     ;Go on-hook
            LDX     #100                ;Delay 100mSec for settling
            ACALL   acDELAY_MSEC

;           JSR     _SET_DUTIO2_OP      ;Enable the I/O DUT port to output
            LDAA    #INH_MUX            ;Enable pogo_if analog mux
            JSR     _CLR_DUTIO1_b       ;
            LDAA    #SELA               ;Select channel 2 of the mux
            JSR     _CLR_DUTIO1_b       ;SELA = 0
            LDAA    #SELB               ;SELB = 1
            JSR     _SET_DUTIO1_b       ;INH  = 0
            LDAA    #F2750_ADJ_LED      ;Turn on the 1st stage adj LED
            JSR     _DUT_LED_ON         ;

            LDX     #$0071              ;Set up FSK attenuator #'0 for 61.5mVp-p
            ACALL   acSET_FSK0_ATT      ;
            LDX     #$0071              ;Set up FSK attenuator #'1 for 61.5mVp-p
            ACALL   acSET_FSK1_ATT      ;
            LDX     #$092A              ;Set first frequency to 2750 Hz
            ACALL   acSET_PSWFSK_ZERO   ;
            LDX     #$092A              ;Set second frequency to 2750 Hz
            ACALL   acSET_PSWFSK_ONE    ;
            ACALL   acFSK_POWER_UP      ;Power up the FSK generator
            LDAA    #TONE_OUT           ;Switch in the FSK signal
            ACALL   acADD_AMUX          ;

            LDAB    #$01                ;Display Test 11 in progress message
            LDX     #TIP11_MSG          ;on line number 2
            JSR     LCD_SCREEN_1
            LDAB    #$02                ;Display line number 3
            LDX     #ADJ_2750_MSG
            JSR     LCD_SCREEN_1
            LDAB    #$03                ;Display line number 4
            LDX     #START_PB_MSG
            JSR     LCD_SCREEN_1

?pb_rel1
            JSR     _DUT_START_PB       ;Make sure pushbutton has been released
            BEQ     ?pb_rel1
?cal0
; Adjust the 2750 first stage filter stage trim Pot & when finished,
; push the START push button on the fixture to continue the test.
            JSR     _DUT_START_PB       ;START pushbutton?
            BNE     ?cal0               ;If yes, exit; else, keep looping

; Sweep the 2750 filter and check the band pass limits
            LDAA    #DUT_RMS_GAIN_10    ;Set PGA to gain of 10
            STAA    GAIN_VALUE          ;
            LDAA    #ANA_RMS5           ;Set up analog channel #5
            STAA    CHAN_NUM            ;
            CLR     RMS_DC_FLAG         ;Set up for RMS measurement

; Read the peak frequency value at 2750 Hz
            JSR     GET_AVG             ;Get peak reading
            STX     FILT_NOM            ;Save peak value
            CPX     #$00C8              ;If low gain exit with error
            BHI     ?gain_ok            ;
            JMP     cal_2750_err        ;
?gain_ok
            LDX     #$0922              ;Set first frequency to 2740 Hz
            ACALL   acSET_PSWFSK_ZERO   ;
            LDX     #$0922              ;Set second frequency to 2740 Hz
            ACALL   acSET_PSWFSK_ONE    ;

            LDX     #50                 ;Delay 5mSec for line voltage to set
            ACALL   acDELAY_MSEC
; Read the low frequency value at 2740 Hz
            JSR     GET_AVG             ;Get average reading
            STX     FILT_LOW            ;Save low frequency value

; Read the high frequency value at 2760 Hz
```

```
            LDX     #$0933              ;Set first frequency to 2760 Hz
            ACALL   acSET_PSWFSK_ZERO   ;
            LDX     #$0933              ;Set second frequency to 2760 Hz
            ACALL   acSET_PSWFSK_ONE    ;

            LDX     #50                 ;Delay 5mSec for line voltage to set
            ACALL   acDELAY_MSEC
            JSR     GET_AVG             ;Get average reading
            STX     FILT_HIGH           ;Save high frequency value

; The nominal value of the filter adjustment should be the peak reading
; and be greater than both the low value and the high value.
            LDX     FILT_NOM            ;Get nominal peak value
            CPX     FILT_LOW            ;Compare nominal to low
            BHI     ?swp0               ;Should be higher
            BRA     cal_2750_err        ;If lower, report adjustment error
?swp0
            CPX     FILT_HIGH           ;Compare nominal to high
            BHI     swp_5               ;Should be higher
cal_2750_err
            LDAB    #$01                ;Display line number 2
            LDX     #GAIN_ERR_MSG
            JSR     LCD_SCREEN_1
            LDAB    #$02                ;Display line number 3
            LDX     #FUNCT_PB_MSG
            JSR     LCD_SCREEN_1
            LDAB    #$03                ;Display line number 4
            LDX     #START_PB_MSG
            JSR     LCD_SCREEN_1
?pb_rel2
            JSR     _DUT_START_PB       ;Make sure pushbutton has been released
            BEQ     ?pb_rel2
swp_2
            JSR     _DUT_START_PB       ;START pushbutton?
            BEQ     swp_4               ;If yes, exit; else, keep looping
            ACALL   acGET_PB            ;Push FUNCTION key on CGT to repeat test
            BEQ     swp_2               ;Else keep looking for a key
            CMPA    #F1_PB              ;If FUNCTION key, then repeat test
            BNE     swp_2               ;Else, keep looping
            JMP     CAL_2750_TST        ;Repeat the test
swp_4
            JSR     ERR_RPT             ;Report 2130 filter adjustment error
swp_5
            LDAA    #TONE_OUT           ;Turn off FSK signal
            ACALL   acRMV_AMUX          ;
            ACALL   acFSK_POWER_DN      ;Power down the FSK
            LDAA    #INH_MUX            ;Disable pogo_if analog mux
            JSR     _SET_DUTIO1_b       ;
            LDX     #50                 ;Delay 50mSec between tests
            ACALL   acDELAY_MSEC
            RTS
            EJECT
;-------------------------------------------------------------------------

CAS_DET_TST:
;  Check the CAS tone detection circuit
;  I/P: None
;  O/P: None
;
            LDAB    #CAS_DET_TST_NUM    ;Set to the correct test number
            STAB    TEST_NUM            ;for the err_rpt routine.
            LDAA    #1                  ;reinitialize for each test (err_rpt)
            STAA    ERR_CODE            ;Set up error code

            LDAB    #$01                ;Display line number 2
            LDX     #BLANK_MSG
            JSR     LCD_SCREEN_1
            LDAB    #$02                ;Display line number 3
            LDX     #TEST_NUM_12
            JSR     LCD_SCREEN_1

            LDX     #$00E2              ;Set up FSK attenuator #'0 for 61.5mVp-p
            ACALL   acSET_FSK0_ATT      ;-23dbm signal
            LDX     #$00E2              ;Set up FSK attenuator #'1 for 61.5mVp-p
            ACALL   acSET_FSK1_ATT      ;
            LDX     #$0719              ;Set CAS low band frequency to 2130 Hz
            ACALL   acSET_PSWFSK_ZERO   ;
            LDX     #$0719              ;Set CAS low band frequency to 2130 Hz
            ACALL   acSET_PSWFSK_ONE    ;
```

```
            LDX     #$00E2              ;Set up for -23dbm amplitude
            ACALL   acSET_PSW1_ATT      ;
            LDX     #$092A              ;Frequency number 2 set to 2750 Hz
            ACALL   acSET_PSW1_FRQ      ;

            ACALL   acFSK_POWER_UP      ;Power up the 2130 Hz signal generator
            ACALL   acPSW1_POWER_UP     ;Power up the 2750 Hz signal generator

            JSR     _ENA_HKSW_RELAY     ;Go off-hook

;           JSR     TP_RESET            ;Turn on voltage to the LIU circuit
;            JSR     SAMSUNG_STROBE
;           LDAA    #$D9
;           JSR     SAMSUNG_READ

            LDAA    #DUT_LOP_1          ;Enable -FSK_EN
            JSR     _CLR_DUT_LOP_b      ;Turn on voltage to the LIU circuit
            JSR     _ENA_DUT_LOP        ;Enable the logical output port

;           JSR     _SET_DUTIO2_IP      ;Set up Input/Output port as input
;           JSR     _ENA_DUTIO2         ;Enable input buffer

            LDX     #50                 ;Delay 50mSec for line voltage to set
            ACALL   acDELAY_MSEC

;            JSR     _RD_DUTIO2          ;Read DUT input port #1
;           BITA    #$01                ;Mask the CASDET bit
;            BEQ     ?cas0               ;CASDET should not be active
;            JSR     ERR_RPT             ;Report CASDET active error
?cas0
            INC     ERR_CODE            ;Set up error code
            LDAA    #CASDET_BIT         ;Mask the -CASDET bit
            JSR     _TST_DUT_LIPORT     ;Read DUT input port #1
            BNE     ?cas1               ;-CASDET should not be active
            JSR     ERR_RPT             ;Report -CASDET active error
            JMP     cas4
?cas1
            INC     ERR_CODE            ;Set up error code
            LDAA    #TONE_OUT           ;Turn on FSK signal
            ACALL   acADD_AMUX          ;

            LDX     #80                 ;Generate 80mSec of CAS tone
            ACALL   acDELAY_MSEC        ;

            LDAA    #TONE_OUT           ;Turn off FSK signal
            ACALL   acRMV_AMUX          ;
            ACALL   acFSK_POWER_DN      ;Power down the 2130 Hz signal generator
            ACALL   acPSW1_POWER_DN     ;Power down the 2750 Hz signal generator
;            LDX     #$0000              ;Set CAS low band frequency to 0 Hz
;            ACALL   acSET_PSWFSK_ZERO   ;
;            LDX     #$0000              ;
;            ACALL   acSET_PSWFSK_ONE    ;
;           LDX     #$0000              ;Set CAS tone high band to 0 Hz
;           ACALL   acSET_PSW1_FRQ      ;

;            LDX     #50000              ;Load timeout counter
            LDX     #50
            ACALL   acDELAY_MSEC
?cas2
;            JSR     _RD_DUTIO2          ;Read DUT input port #1
;           BITA    #$01                ;Mask the CASDET bit
;            BNE     ?cas3               ;CASDET should be active
;           DEX                         ;Decrement timeout counter
;           BNE     ?cas2               ;Loop until timeout or active CASDET

;            JSR     ERR_RPT             ;Report CASDET not active error
?cas3
            INC     ERR_CODE            ;Set up error code
            LDAA    #CASDET_BIT         ;Mask the -CASDET bit
            JSR     _TST_DUT_LIPORT     ;Read DUT input port #1
            BEQ     cas4                ;-CASDET should be active
            JSR     ERR_RPT             ;Report -CASDET not active error
cas4
;            JSR     _DIS_DUT_LOP        ;Disable the logical output port
;           JSR     _DIS_DUTIO2         ;Disable input buffer
            JSR     _DIS_HKSW_RELAY     ;Go on-hook
            RTS
            EJECT
```

```
;-------------------------------------------------------------------------
RING_TST:
;  Check the ring detect circuit and piezo driver output signal
;  I/P: None
;  O/P: None
;
            LDAB    #RING_TST_NUM       ;Set to the correct test number
            STAB    TEST_NUM            ;for the err_rpt routine.
            LDAA    #1                  ;reinitialize for each test (err_rpt)
            STAA    ERR_CODE            ;Set up error code

            LDAB    #$02                ;Display line number 3
            LDX     #TEST_NUM_13
            JSR     LCD_SCREEN_1

            LDX     #$0428              ;Set line voltage to 0.0Vdc
            ACALL   acSET_DC_LINEVOLTAGE

            JSR     _DIS_HKSW_RELAY     ;Go on-hook
            LDX     #20                 ;Delay 20 mSec for relay settle
            ACALL   acDELAY_MSEC
            ACALL   acRING_RELAY_ON     ;Activate ring relay
            ACALL   acPSW3_POWER_UP     ;Turn on Ring generator
            LDX     #$0011              ;Set ring frequency to 20 Hz
            ACALL   acSET_PSW3_FRQ
            LDX     #$0800              ;Set ring amplitude to 40 Vrms
            ACALL   acSET_PSW3_ATT      ;
; Check the -RING digital signal, should be inactive high with no ring present
            LDAA    #RING_BIT           ;Mask the -RING bit
            JSR     _TST_DUT_LIPORT     ;Read DUT input port #1
            BNE     ?ring0              ;Ring should not be active
            JSR     ERR_RPT             ;Report Ring active error
?ring0
            INC     ERR_CODE            ;Set up error code
            LDAA    #RING_OUT           ;
            ACALL   acADD_AMUX          ;Switch in the ring voltage
            LDX     #100                ;Delay 100mSec for ring settle
            ACALL   acDELAY_MSEC        ;
            LDAA    #RING_BIT           ;Mask the -RING bit
            JSR     _TST_DUT_LIPORT     ;Read DUT input port #1
            BEQ     ?ring1              ;Ring should be active
            JSR     ERR_RPT             ;Report Ring not active error
?ring1
            INC     CAPRLY_FLAG         ;Enable cap relay
            INC     RMS_DC_FLAG         ;Set up for RMS measurement
            LDAA    #DUT_RMS_GAIN_1     ;Set PGA to gain of 1
            STAA    GAIN_VALUE          ;
            LDAA    #3                  ;Load the analog channel number
            STAA    CHAN_NUM            ;
            JSR     _RMS_MEASURMENT     ;Get the average reading
            CPX     #$0100              ;
            BHI     ?ring2              ;Check if signal present & > min limit
            JSR     ERR_RPT             ;Report piezo driver error
?ring2
            LDAA    #RING_OUT           ;Remove Ring voltage
            ACALL   acRMV_AMUX          ;
            ACALL   acPSW3_POWER_DN     ;Turn off the Ring generator
            ACALL   acRING_RELAY_OFF    ;Switch off the Ring relay
            CLR     CAPRLY_FLAG         ;Clear cap relay flag
            JSR     _ENA_HKSW_RELAY     ;Go off-hook
            LDX     #50                 ;Delay 50mSec between tests
            ACALL   acDELAY_MSEC
            RTS
            EJECT
;-------------------------------------------------------------------------
DIAL_TONE_TST:
;  Check the dial tone detection circuit
;  I/P: None
;  O/P: None
;
            LDAB    #DIAL_TONE_TST_NUM  ;Set to the correct test number
            STAB    TEST_NUM            ;for the err_rpt routine.
            CLR     ERR_CODE            ;reinitialize for each test (err_rpt)

            LDAB    #$02                ;Display line number 3
            LDX     #TEST_NUM_14
            JSR     LCD_SCREEN_1

            LDX     #$0161              ;Set up FSK attenuator #'0 for 1.0Vp-p
```

```
            ACALL   acSET_FSK0_ATT      ;
            LDX     #$0161              ;Set up FSK attenuator #'1 for 1.0Vp-p
            ACALL   acSET_FSK1_ATT      ;
            LDX     #$0119              ;Set first frequency to 330 Hz
            ACALL   acSET_PSWFSK_ZERO   ;
            LDX     #$0177              ;Set second frequency to 440 Hz
            ACALL   acSET_PSWFSK_ONE    ;
            ACALL   acFSK_POWER_UP      ;Power up the FSK

            BITA    #DIALTONE_BIT       ;Mask the DIAL.TONE bit
            JSR     _TST_DUT_LIPORT     ;Read DUT input port #1
            BNE     ?dt0                ;DIAL.TONE should not be active
            JSR     ERR_RPT             ;Report DIAL.TONE active error
?dt0
            INC     ERR_CODE            ;Set up error code
            LDAA    #TONE_OUT           ;Turn on FSK signal
            ACALL   acADD_AMUX          ;
            LDX     #50                 ;Delay before reading
            ACALL   acDELAY_MSEC

            LDAA    #DIALTONE_BIT       ;Mask the DIAL.TONE bit
            JSR     _TST_DUT_LIPORT     ;Read DUT input port #1
            BEQ     ?dt1                ;DIAL.TONE should not be active
            JSR     ERR_RPT             ;Report DIAL.TONE active error
?dt1
            LDAA    #TONE_OUT           ;Turn off FSK signal
            ACALL   acRMV_AMUX          ;
            ACALL   acFSK_POWER_DN      ;Power down the FSK
            RTS
            EJECT
;*****************************************************************************
;* Name     : WAIT                                                           *
;* Function : SPEND TIME LOOPING AROUND ~ 100mS * Y                          *
;* Enter    : Y = number of 100mS loops to be executed                       *
;* Return   : None                                                           *
;*            [ ( 6 * 40953 +7) * Y + 35 ]                                   *
;*            ----------------------                                         *
;*                  2.4576 MHz                                               *
;*****************************************************************************
;
WAIT:       PSHY                        ;5 cycles
            PSHX                        ;4 cycles
            LDX     #$9FF9              ;3 cycles
?WT_LP1     DEX                         ;3 cycles
            BNE     ?WT_LP1             ;3 cycles
            DEY                         ;4 cycles
            BNE     ?WT_LP1             ;3 cycles
            PULX                        ;5 cycles
            PULY                        ;6 cycles
            RTS                         ;5 cycles

;*****************************************************************************
;* Name     : CLR_BUFF                                                       *
;* Function : Clear error buffer routine                                     *
;* ENTRY    : none                                                           *
;* Return   : ERR_BUFF -> ERR_BUFF_END (ERR_BUFF+52) = 0                     *
;*****************************************************************************
;
CLR_BUFF:
            CLRA
            LDX     #ERR_BUFF           ;Point to error buffer
            STX     ERR_PTR             ;Save error register pointer
?CLR_0:
            STAA    0,X                 ;Clear error buffer
            INX                         ;Point to next buffer location
            CPX     #ERR_BUFF_END
            BNE     ?CLR_0
            CLR     ERR_CNT             ;Clear the error counter
            RTS
;*****************************************************************************
;* Name     : LCD_SCREEN_1                                                   *
;* Function : Display one line of data                                       *
;* Entry    : X Point to start of 4 line message                             *
;*          : B line number of message (0,1,2,3) no error checking           *
;* Return   : None                                                           *
;*****************************************************************************
;
LCD_SCREEN_1:
            CMPB    #$03
```

```
            BLS     LCD_SCREEN_1_CONT
            LDAB    #$31
            STAB    ERR_INTERNALA       ;Internal error so print & abort
            JSR     INTERN_ERROR
            LDAB    #$20
            STAA    ERR_INTERNALA
            JMP     LCD_SCREEN_1_EXIT
LCD_SCREEN_1_CONT
            PSHY                        ;Save contents of Y
            LDY     #LCD_LPTR
            LSLB
            ABY                         ;Save index to memory where to store
                                        ;pointer of data to be printed to LCD
            PSHY
            PSHX                        ;Save data pointer to message
LCD_SCREEN_1_L1:
            ACALL   acDISP_BF           ;1st check to see if data is
                                        ;still waiting to be displayed
            BNE     LCD_SCREEN_1_L1     ;if ne then loop to wait until
                                        ;previous data is sent out to
                                        ;the LCD
            PULX                        ;Restore previous data pointer
                                        ;to message to be displayed
            PULY
            STX     $0,Y                ;Write to the 1st LCD line
            PULY
LCD_SCREEN_1_EXIT:
            RTS


;*************************************************************************
;* Name     : LCD_SCREEN_4                                               *
;* Function : Display the intial screen                                  *
;* Entry    : X Point to start of 4 line message                         *
;* Return   : None                                                       *
;*************************************************************************
;
LCD_SCREEN_4:
            PSHX                        ;Save data pointer to message
            PSHY
            PSHB
LCD_SCREEN_4_L1:

            LDY     #LCD_LPTR
            LDAB    #$15
            BSR     LCD_SCREEN_4_PRNT
            ABX
            INY
            INY
            BSR     LCD_SCREEN_4_PRNT
            ABX
            INY
            INY
            BSR     LCD_SCREEN_4_PRNT
            ABX
            INY
            INY
            BSR     LCD_SCREEN_4_PRNT
            PULB
            PULY
            PULX
            RTS

LCD_SCREEN_4_PRNT:

            PSHX
            PSHY
            PSHB

LCD_SCREEN_4_PRNT_LP:

            ACALL   acDISP_BF           ;1st check to see if data is
                                        ;still waiting to be displayed
            BNE     LCD_SCREEN_4_PRNT_LP  ;if ne then loop to wait until
                                        ;previous data is sent out to
            PULB                        ;the LCD
            PULY                        ;Restore previous data pointer
            PULX                        ;to message to be displayed
            STX     0,Y                 ;Write to the 1st LCD line
```

```
          RTS

;*****************************************************************************
;* Name     : CK_DUT                                                         *
;* Function : Check DUT continuity routine                                   *
;* Entry    : None                                                           *
;* Return   : None                                                           *
;*****************************************************************************

CK_DUT:
          JSR     _ENA_DUT_LIP1
?CK_LID
          JSR     _DUT_FIXPB          ;Check that the lid is closed
          BNE     ?CK_DUT_L1
          LDX     #CHK1_MSG0          ;Display that the fixture top
          JSR     LCD_SCREEN_4        ;is not properly closed.
?CK_LID1
          JSR     _DUT_START_PB       ;Wait for START button to be pushed.
          BNE     ?CK_LID1
          BRA     ?CK_LID
?CK_DUT_L1
          LDAA    #DUT_LIN5           ;This line is the continuity bit
          JSR     _TST_DUT_LIPORT     ;on the DUT
;         BRA     ?DUT_RET
          BEQ     ?DUT_RET
          LDX     #CHK_MSG0           ;Display CHECK INSTALLATION message
          JSR     LCD_SCREEN_4
?CK_DUT1
          JSR     _DUT_START_PB       ;Wait for START button to be pushed.
          BNE     ?CK_DUT1
          BRA     ?CK_LID             ;Go back and check continuity again
?DUT_RET
          LDX     #START_MSG0
          JSR     LCD_SCREEN_4        ;Display hit START to continue message
          RTS

;*****************************************************************************
;* Name     : EN_LIPORT                                                      *
;* Function : Enable the logical input ports                                 *
;* Name     : None                                                           *
;* Return   : None                                                           *
;*****************************************************************************
;
EN_LIPORT:
          JSR     _ENA_DUT_LIP0
          JSR     _ENA_DUT_LIP1
          RTS


;*****************************************************************************
;* Name     : PWR_OFF                                                        *
;* Function : Turn off DUT power                                             *
;* Name     : None                                                           *
;* Return   : None                                                           *
;*****************************************************************************
;
PWR_OFF:

***WATCHOUT
          LDAB    #TGT_PC0            ;Turn off EN_SIF so that no
          LDAA    #$00                ;power leaks to Samsung during reset
***WATCHOUT
          STAA    LTGT_PC0
          JSR     _WR_DUT
          JSR     _DIS_DUT_PWR
          LDY     #$05                ; Delay ~ 500mS to allow power to
          JSR     WAIT                ; stabilize.

;          LDAA    #TIP_LED            ;Test In Progress LED
;          JSR     _SET_DUTIO1_b       ;Turn off LED (5V-off 0V-on)
          JSR     _DIS_DUT_LOP
          JSR     _DIS_HKSW_RELAY
          LDX     #DC_OUT
          ACALL   acRMV_AMUX
          RTS

;*****************************************************************************
;* Name     : CK_PASS                                                        *
;* Function : Check if test passes or fails                                  *
```

```
;* Entry    : ERR_CNT flag = 0, test passes                                    *
;*                   flag = 1. test fails                                      *
;* Return   : Pass or Fail LED is turned on                                    *
;* Saved    : X                                                                *
;*****************************************************************************
;
CK_PASS:

          LDAA    ERR_CNT             ;Check for error
          BNE     ?ERR_TRUE           ;If error, turn on Fail LED


?ERR_FALSE

          LDAA    #PASS_LED
          JSR     _SET_DUTIO1_b
          LDAA    #FAIL_LED
          JSR     _CLR_DUTIO1_b
          BRA     ?CK_PASS_EXIT

?ERR_TRUE
          LDAA    #PASS_LED
          JSR     _CLR_DUTIO1_b
          LDAA    #FAIL_LED
          JSR     _SET_DUTIO1_b


?CK_PASS_EXIT
          LDAA    #TIP_LED
          JSR     _CLR_DUTIO1_b

          RTS

;*****************************************************************************
;*       Routine:  DSP_ERR                                                   *
;*       Function: Display error code and accompanying error message         *
;*       Entry:    Total number of errors in ERR.CNT                         *
;*       Return:   None                                                      *
;*****************************************************************************
;
DSP_ERR:
          LDAB    ERR_CNT             ;Check for any error
          BNE     DSP_ERR_0
          JMP     DSP_ERR_EXIT        ;If no error, exit
DSP_ERR_0:
          LDX     #ERR_BUFF           ;Save error buffer pointer
          STX     SAVX1

          LSLB                        ;Error codes are 2 bytes so mult. by 2
          SUBB    #$02
          ABX                         ;Add last error location to pointer
          STX     MAX_BUF             ;Save error buffer pointer max limit

          LDX     #ERR_MSG            ;Point to start of error message table
          STX     MSG_BUF             ;Save error message table pointer

          LDX     #CODE_MSG0          ;Display active switch functions
          JSR     LCD_SCREEN_4        ;Display functions on 4th line

          CLRB                        ;Display the number of errors
          LDAA    ERR_CNT             ;Get the error count
          CMPA    #10                 ;Check if > 10
          BGE     ERR_CNT_GT_10
          ORAA    #$30                ;Else, display single BCD number
          BRA     ERR_CNT_LT_10

ERR_CNT_GT_10:

          SUBA    #10
          BCS     ERR_MISCALC
          INCB
          BRA     ERR_CNT_GT_10

ERR_MISCALC:

          ADDA    #10
          PSHA
          ORAB    #$30                ;Adjust to ASCII
          STAB    CODE_MSG1B
```

```
            LDX     #CODE_MSG0
            PULA
            ORAA    #$30                ;Adjust to ASCII
            STAA    CODE_MSG1B+1
            JSR     LCD_SCREEN_4
            BRA     ERR_1

ERR_CNT_LT_10:

            STAA    CODE_MSG1B
            JSR     LCD_SCREEN_4

ERR_1:

            LDX     MSG_BUF             ;Restore error message pointer

ERR_SRCH_CODE_NUM:

            LDD     $0,X                ;Get 1st word of error message
            LDX     SAVX1
            CPD     $0,X
            BNE     ERR_INC_MSG_BUF     ;Branch if not
            LDX     MSG_BUF             ;Restore error message pointer
            INX                         ;and bump pointer to start of message
            INX
            BRA     ERR_FOUND_CODE_NUM  ;and go display it

ERR_INC_MSG_BUF:

            LDD     MSG_BUF             ;Add offset of 22 to pointer
            ADDD     #$0017
            XGDX                        ;X now contains pointer to err_msg data
            STX     MSG_BUF             ;Save pointer
            CPX     #ERR_END            ;Check if end of message table
            BLO     ERR_SRCH_CODE_NUM   ;Else, check next byte in table

DSP_ERROR_INTERN_ERR:

            LDAB    #$32                ;Internal error past end of message table code 2
            STAB    ERR_INTERNALA
            JSR     INTERN_ERROR        ;display error message
            LDAB    #$20                ;reset ram to initial value
            STAB    ERR_INTERNALA
            JMP     DSP_ERR_EXIT        ;break out of routine ouch!

ERR_FOUND_CODE_NUM:

            LDAB    #$02                ;Point to 3rd line of LCD
;            ABX                        ;Point to data & point to line #
            JSR     LCD_SCREEN_1        ;display error message

            LDX     SAVX1               ;Restore current error buffer pointer
            LDAB    $0,X                ;Get test number where error occured
            JSR     XLAT_BIN_TO_ASCII   ;from the err_buff and translate to ascii
            STD     CODE_MSG1A          ;& store in message line
            INX
            LDAB    $0,X                ;Get test sub function where error occured
            JSR     XLAT_BIN_TO_ASCII   ;from the err_buff and translate to ascii
            STD     CODE_MSG1A+2        ;& store in message line
            LDX     #CODE_MSG1
            LDAB    #$01                ;Point to 3rd line of LCD
            JSR     LCD_SCREEN_1        ;display test# & dubfunction# where
                                        ;error occured

;*************** Check for a key closure *********************************
DSP_ERR_LK_KEY:

            ACALL   acGET_PB            ;Clear previous PB in register

DSP_ERR_LK_KEY_LP:

            ACALL   acGET_PB            ;Get key closure
            BEQ     DSP_ERR_LK_KEY_LP

            CMPA    #$02
            BEQ     DSP_ERR_INC_ERR_BUF ;Increment error code

            CMPA    #$03
            BEQ     DSP_ERR_DEC_ERR_BUF ;Decrement error code
```

```
            CMPA    #$01                ;Esc key pressed?
            BNE     DSP_ERR_LK_KEY      ;If not, then loop around again
            BRA     DSP_ERR_EXIT        ;Otherwise exit

;**************  Increment the error message & error code ******************

DSP_ERR_INC_ERR_BUF:

            LDX     SAVX1               ;Get error buffer pointer
            INX                         ;Increment buffer pointer
            INX
            CPX     MAX_BUF             ;Check for last error
            BLS     ?DSP_ERR_INC_ERR_NORM ;If last error, branch

?DSP_ERR_WRAP_BEGIN

            LDX     #ERR_BUFF           ;Else, point to beggining of buffer

?DSP_ERR_INC_ERR_NORM

            STX     SAVX1               ;and save pointer
            LDX     #ERR_MSG            ;Point to start of error message table
            STX     MSG_BUF             ;Save error message table pointer
            JMP     ERR_SRCH_CODE_NUM   ;Jump back to error code display

;**************  Decrement the error message & error code ******************
DSP_ERR_DEC_ERR_BUF:

            LDX     SAVX1               ;Get error buffer pointer
            DEX                         ;Decrement buffer pointer
            DEX
            CPX     #ERR_BUFF           ;Check for 1st error location
            BHS     ?DSP_ERR_DEC_ERR_NORM ;If not first error, branch

?DSP_ERR_DEC_WRAP_END

            LDX     MAX_BUF             ;Else, point to last error in buffer

?DSP_ERR_DEC_ERR_NORM

            STX     SAVX1               ;and save pointer
            LDX     #ERR_MSG            ;Point to start of error message table
            STX     MSG_BUF             ;Save error message table pointer
            JMP     ERR_SRCH_CODE_NUM   ;Jump back to error code display

DSP_ERR_EXIT:

            LDD     #$2020              ;Restore message lines that were changed
            STD     CODE_MSG1A          ;to error number test etc. (in string)
            STD     CODE_MSG1A+2        ;back to spaces.
            STD     CODE_MSG1B
            RTS


;*****************************************************************************
;*       Routine : INTERN_ERROR                                              *
;*       Function: Display INTERNAL error code                               *
;*       Entry   : B                                                         *
;*       Return  : None                                                      *
;*****************************************************************************

INTERN_ERROR:
            LDX     #ERR_INTERNAL
            LDAB    #$0                 ;1st line
            JSR     LCD_SCREEN_1
            LDY     #$05                ;Wait ~500mS
            JSR     WAIT
            LDAB    #$01                ;2nd line
            JSR     LCD_SCREEN_1
            JSR     WAIT                ;Wait ~500mS
            LDAB    #$02                ;3rd line
            JSR     LCD_SCREEN_1
            JSR     WAIT                ;Wait ~500mS
            LDAB    #$03                ;4th line
            JSR     LCD_SCREEN_1
            JSR     WAIT                ;Wait ~500mS
            RTS
```

```
;*************************************************************************
;*       Routine : XLAT_BIN_TO_ASCII                                     *
;*       Function: translate binary to ascii character                   *
;*       Entry   : B binary number to be converted                       *
;*       Return  : A MS nyble converted to ascii                         *
;*               : B LS nyble converted to ascii                         *
;*************************************************************************

XLAT_BIN_TO_ASCII:

XLAT_MSN:

           LSLD
           LSLD                        ;Shift upper nibble to lower nibble
           LSLD
           LSLD
           ANDA    #$0F
           CMPA    #$A                 ;Check for Alpha character
           BGE     XLAT_MSN_GR_10
           ORAA    #$30                ;Translate to ASCII numeric
           BRA     XLAT_LSN

XLAT_MSN_GR_10:

           SUBA    #9                  ;Adjust for Alpha
           ORAA    #$40                ;and adjust again for ASCII

XLAT_LSN:
           LSRB
           LSRB
           LSRB
           LSRB
           ANDB    #$0F
           CMPB    #$A
           BGE     XLAT_LSN_GR_10
           ORAB    #$30                ;Translate to ASCII numeric
           BRA     XLAT_BIN_TO_ASCII_EXIT

XLAT_LSN_GR_10:

           SUBB    #9                  ;Adjust for Alpha
           ORAB    #$40                ;and adjust again for ASCII

XLAT_BIN_TO_ASCII_EXIT:
           RTS

;*************************************************************************
;*       Routine : REFRESH_LED                                           *
;*       Function: UPDATE VISUAL LED'S                                   *
;*       Entry   : NONE                                                  *
;*       Return  : NONE                                                  *
;*       Saved   : X                                                     *
;*************************************************************************

REFRESH_LED:
           PSHX
           JSR     _ENA_DUTIO1
           LDAA    LED_BANK_1
           ANDA    #$FF
           JSR     _WR_DUTIO1
           PULX
           RTS

;*************************************************************************
;*       Routine : ERR_RPT                                               *
;*       Function: Insert error code into error buffer                   *
;*       Entry   : A:TEST NUMBER                                         *
;*                 B:FUNCTION NUMBER                                     *
;*       Return  : none                                                  *
;*************************************************************************

ERR_RPT:

           PSHA                        ;save current data
           PSHB                        ;get starting location of error buffer
           LDX     #ERR_BUFF           ;generate an index into the error buffer
           LDAB    ERR_CNT             ;by multiplying the error count by 2
           LSLB                        ;and adding that to the start of the
           ABX                         ;error message buffer
```

```
            LDAB    ERR_CNT             ;get error count again
            INCB                        ;and update the count by one
            STAB    ERR_CNT
            LDAA    TEST_NUM
            LDAB    ERR_CODE
            STD     $0,X
            PULB
            PULA
            RTS

;-------------------------------------------------------------------------

GET_AVG:
;  Take average of 16 RMS or DC readings
;  I/P: None
;  O/P: X = Average Value (quotient)
;
            LDAA    #16                 ;Load average counter
            STAA    COUNT               ;
            LDD     #0                  ;Clear out sum area
            STD     SUM
            LDAA    RMS_DC_FLAG         ;Get RMS or DC measurement flag
            BNE     avg1
; Take sum of 16 RMS readings
avg0
            LDAA    CAPRLY_FLAG         ;
            BEQ     ?cap_off            ;CAP relay on or off?
            SEC                         ;C=1 for CAP relay on
            BRA     ?cap_on
?cap_off
            CLC                         ;C=0 for Cap relay off
?cap_on
            LDAA    CHAN_NUM            ;Select channel to read
            LDAB    GAIN_VALUE          ;Set PGA to gain of 5
            JSR     _RMS_MEASURMENT     ;Get RMS gain reading
            ADDD    SUM                 ;Sum value
            STD     SUM                 ;Save new sum
            DEC     COUNT               ;Decrement average counter
            BNE     avg0                ;Loop until 16 reads have been taken
            BRA     avg2
avg1
; Take sum of 16 DC readings
            LDAA    CHAN_NUM            ;Select channel to read
            LDAB    #10                 ;Load settle time = 20uSec
            JSR     _CNVRT_ANA_ICH      ;Get DC reading
            ADDD    SUM                 ;Sum value
            STD     SUM                 ;Save new sum
            DEC     COUNT               ;Decrement average counter
            BNE     avg1                ;Loop until 16 reads have been taken
; Take the average of the sum
avg2
            LDX     #16                 ;Load denominator value
            LDD     SUM                 ;Load numerator value
            IDIV                        ;Do 16-bit divide
            RTS
            EJECT
;-------------------------------------------------------------------------

LED_ADJ:
;  Turn on LOW, OK or HIGH front panel LED in response to A-to-D reading
;  as a result of adjusting the DTMF gain trim pot.
;  I/P: Maximum A/D reading in location MAX
;       Minimum A/D reading in location MIN
;       A-to-D value in the X reister
;
            CPX     MAX                 ;Is the window above the max limit?
            BHI     ?ledhi              ;If yes, turn on the HIGH LED
            CPX     MIN                 ;Is the window below the min limit?
            BHI     ?ledok              ;If yes, turn on the LOW LED
                                        ;Else, turn on the OK LED
            LDAA    #6                  ;Turn on LOW LED
            JSR     _DUT_LED_ON
            LDAA    #7                  ;Turn off OK LED
            JSR     _DUT_LED_OFF
            LDAA    #8                  ;Turn off HIGH LED
            JSR     _DUT_LED_OFF
            CLRA                        ;Clear adjustment error flag
            STAA    ERR_FLAG
            BRA     ?led2               ;Return
```

```
?ledhi
            LDAA    #6                  ;Turn off LOW LED
            JSR     _DUT_LED_OFF
            LDAA    #7                  ;Turn off OK LED
            JSR     _DUT_LED_OFF
            LDAA    #8                  ;Turn on HIGH LED
            JSR     _DUT_LED_ON
            LDAA    ERR_FLAG            ;Set adjustment error flag
            ANDA    #$01
            STAA    ERR_FLAG
            BRA     ?led2               ;Return
?ledok
            LDAA    #6                  ;Turn off LOW LED
            JSR     _DUT_LED_OFF
            LDAA    #8                  ;Turn off HIGH LED
            JSR     _DUT_LED_OFF
            LDAA    #7                  ;Turn on OK LED
            JSR     _DUT_LED_ON
            LDAA    ERR_FLAG            ;Set adjustment error flag
            ANDA    #$01
            STAA    ERR_FLAG
?led2
            RTS
            EJECT
;-------------------------------------------------------------------------
DCRES_RLY:
; Enable or disable DC resistance measurement relay
; I/P: ACCA = 0, relay off
;           = 1, relay on
; O/P: None
;
            BEQ     ?rly0               ;Relay on or off
            LDAA    LTGT_PC0            ;On, so set bit high
            ORAA    #$80                ;
            BRA     ?rly1               ;Go turn on relay
?rly0
            LDAA    LTGT_PC0            ;Off, so set bit low
            ANDA    #$7F                ;
?rly1
            STAA    LTGT_PC0            ;Save port contents
            LDAB    #TGT_PC0            ;Get Target PC1 address
            JSR     _WR_DUT             ;Write to target port PC1
            RTS

;-------------------------------------------------------------------------
EN_SIF:
; Enable or disable the microwire serial interface buffers
; I/P: ACCA = 0, SIF off
;           = 1, SIF on
; O/P: None
;
            BEQ     ?sif0               ;SIF enabled or disabled
            LDAA    LTGT_PC0            ;On, so set bit high
            ORAA    #$40                ;
            BRA     ?sif1               ;Go enable SIF
?sif0
            LDAA    LTGT_PC0            ;Off, so set bit low
            ANDA    #$BF                ;
?sif1
            STAA    LTGT_PC0            ;Save port contents
            LDAB    #TGT_PC0            ;Get Target PC0 address
            JSR     _WR_DUT             ;Write to target port PC1
            RTS
;-------------------------------------------------------------------------
SMOS_RDY:
; Sends SMOS ready status to the STEALTH
; I/P: None
; O/P: None
;
            LDAA    #DUT_LOP_0          ;Toggle WAKE.UP bit from low to high
            JSR     _CLR_DUT_LOP_b      ;this indicates SMOS is ready
            LDX     #1                  ;1mSec pulse
            ACALL   acDELAY_MSEC
            LDAA    #DUT_LOP_0
            JSR     _SET_DUT_LOP_b
            RTS

;-------------------------------------------------------------------------
SET_DUT_FSKSIG:
```

```
; Wait for Free SPI time.
;       Is SPI service busy?
;       I/P: none
;       O/P: Z=0 if SPI service is NOT FREE.
;            Z=1 if SPI if FREE.
; Check availability of SPI
; Preserve Acc. A reg.
; Checking SPSR is Not a valid test. Most of SPI services communicate via SPI
; on multiple byte protocal.
?Wait:
               TST     DUT25TST_SPI_TXDNUM      ;
               BNE     ?Wait           ; Br. if SPI is busy.

               LDAA    #$01
               STAA    DUT25TST_SPI_TXDNUM      ; Make SPI busy.

               XGDX                     ;TRANSFER DATA FROM THE X REGISTER TO D
               PSHB

               STAA    SPDR            ; transmit.

?Wt:           LDAA    SPSR
               ANDA    #SPIF           ; Is SPI buffer ready?
               BEQ     ?Wt             ; Br. if Not, in process of data transmission.
               LDAA    SPDR            ; the received data.

               PULA                    ;GET READY TO SEND THE SECOND BYTE TO THE DAC

               STAA    SPDR

?Wt2:          LDAA    SPSR
               ANDA    #SPIF           ; Is SPI buffer ready?
               BEQ     ?Wt2            ; Br. if Not, in process of data transmission.
               LDAA    SPDR            ; the received data.

               LDAA    LHSE_PA0
               ANDA    #\_LD_ATT       ; Enable
               LDAB    #HSE_PA0
               JSR     _WR_DUT
               LDAA    LHSE_PA0
               ORAA    #_LD_ATT        ; Disable
               LDAB    #HSE_PA0
               JSR     _WR_DUT
               CLR     DUT25TST_SPI_TXDNUM      ; SPI Free.

KRUGER2:
               LDAB    #S2A
               PSHA
;              JSR     SET_LDB         ; Set local data bus.
;SET_LDB:
;       Set local data bus, LDB0 -- LDB3.
;
;       I/P: Acc. B the 4 bit data/addr.  Acc. 4 -- Acc. 7
;       O/P: none.
;
; Preserves IX, IY

               LDAA    LHSE_PB0
               ANDA    #$0F
               STAA    LHSE_PB0
               ANDB    #$F0            ; Make sure no other bits, only GPIO0-3
               ORAB    LHSE_PB0
               STAB    LHSE_PB0
               TBA
               LDAB    #HSE_PB0
               JSR     _WR_DUT

               LDAA    LHSE_PA0
               ANDA    #\_FSK_WR       ; Wr strobe
               LDAB    #HSE_PA0
               JSR     _WR_DUT
               LDAA    LHSE_PA0
               LDAB    #HSE_PA0
               JSR     _WR_DUT          ; Negate Wr. st.
               PULA
               RTS

;-------------------------------------------------------------------------
INIT_FSK:
```

```
;  Initialize the FSK tone generators
;  I/P: None
;  O/P: None
;
            LDX     FSK_ATT1            ;Set up FSK attenuator #'0 for 61.5mVp-p
            ACALL   acSET_FSK0_ATT      ;-23dbm signal
            LDX     FSK_ATT2            ;Set up FSK attenuator #'1 for 61.5mVp-p
            ACALL   acSET_FSK1_ATT      ;
            LDX     FSK_FRQ1            ;Set CAS low band frequency to 2130 Hz
            ACALL   acSET_PSWFSK_ZERO   ;
            LDX     FSK_FRQ2            ;Set CAS low band frequency to 2130 Hz
            ACALL   acSET_PSWFSK_ONE    ;
            LDAA    PSW1_FLAG           ;If flag set, initialize PSW1
            BEQ     PWRUP_FSK           ;PSW1 = the second FSK tone generator
INIT_PSW1
            LDX     PSW1_ATT            ;Set up for -23dbm amplitude
            ACALL   acSET_PSW1_ATT      ;
            LDX     PSW1_FRQ            ;Frequency number 2 set to 2750 Hz
            ACALL   acSET_PSW1_FRQ      ;
            ACALL   acPSW1_POWER_UP     ;Power up the 2750 Hz signal generator
PWRUP_FSK
            ACALL   acFSK_POWER_UP      ;Power up the 2130 Hz signal generator
            RTS
            EJECT
;*******************************************************************************************
TP_RESET:
; Reset the microwire interface chip
; Input : None
; Output: None
;
            LDAA    #00001000B          ;TP reset line
            JSR     _CLR_DUTIO1_b       ;Bring reset line low  (reset)
            LDX     #$32                ;~32mS LOW
            ACALL   acDELAY_MSEC
            LDAA    #00001000B
            JSR     _SET_DUTIO1_b       ;Bring reset line high (out of reset)
            LDX     #$01F4
            ACALL   acDELAY_MSEC        ;~.5S before accessing part
            JSR     _uW_SLAVE
            JSR     _DUT_uW_BYTE
            RTS

;-------------------------------------------------------------------------
SAMSUNG_RESET:
; On return:
; B: error code status
;    1=error  Z=0
;    0= ok    Z=1
            JSR     _DIS_DUT_LOP        ;Turn WAKE.UP off so no power leaks to shark
            LDAA    #DUT_LOP_0
            JSR     _CLR_DUT_LOP_b
            LDAA    #DUT_LOP_1          ;Clear -FSK_LED switch
            JSR     _CLR_DUT_LOP_b
;***WATCHOUT
            LDAB    #TGT_PC0            ;Turn off EN_MID so that no
            LDAA    #$00                ;power leaks to Samsung during reset
;***WATCHOUT
            STAA    LTGT_PC0
            JSR     _WR_DUT
            JSR     _DIS_DUT_PWR
            JSR     _DIS_HKSW_RELAY     ;Go on-hook
            LDX     #750                ;line voltage is present
            ACALL   acDELAY_MSEC        ;Delay for voltage to decay to 0V

            JSR     _ENA_HKSW_RELAY     ;Go off-hook
            LDX     #100                ;Off-hook settle time
            ACALL   acDELAY_MSEC
            LDAA    #EN_9VDC            ;
            JSR     _ENA_DUT_PWR        ;
            JSR     _DIS_DUT_PWRELAY    ;Power relay to AC adapter side
            LDAA    #DUT_LOP_1          ;Set -FSK_LED switch inactive
            JSR     _SET_DUT_LOP_b      ;after power is reapplied
            LDY     #10000              ;Load timeout counter

SAMS_RES_HI:
            LDAA    #RESET_BIT          ;Wait for _RESET to go high
            PSHY                        ;
            JSR     _TST_DUT_LIPORT
            PULY
```

```
            DEX
            BNE     SAMS_uW_READY_CHK
            DEC     mMATRIX_MAX_ATT
            BEQ     SAMS_COMM_FAIL
            JSR     TP_RESET
            BRA     SAMS_uW_RCV_DATA_START
SAMS_COMM_FAIL:
            PULA
            LDAB    #$01                ;Failed to receive the data from Samsung
            RTS                         ;QUIT cannot continue if Samsung does not respond
SAMS_uW_READ_OK:
            PULX
            PULA
            JSR     _uWX_RDBYTE
            CLRB
            RTS

;-------------------------------------------------------------------------
SAMSUNG_STROBE:
          LDAA      #DUT_LOP_0          ;Toggle WAKE.UP bit before sending command
          JSR       _SET_DUT_LOP_b
          LDX       #1
          ACALL     acDELAY_MSEC
          LDAA      #DUT_LOP_0
          JSR       _CLR_DUT_LOP_b
          LDX       #1
          ACALL     acDELAY_MSEC
          LDX       #50
          ACALL     acDELAY_MSEC
          RTS

.DATA

;-------------------------------------------------------------------------
; RAM data variable declarations
;
CAPRLY_FLAG     .DS     1           ;C=0, cap relay off; C=1, cap relay on
CHAN_NUM        .DS     1           ;Analog channel mux number
COUNT           .DS     1           ;General purpose counter register
DC_ERR_FLAG     .DS     1           ;DC voltage error flag
ERR_CNT         .DS     1           ;Error counter
ERR_CODE        .DS     1           ;Error code register
ERR_FLAG        .DS     1           ;Error flag register
FSK_REG         .DS     1           ;FSK / Tone out gain error register
FSK_ATT1        .DS     1           ;
FSK_ATT2        .DS     1           ;
FSK_FRQ1        .DS     1           ;
FSK_FRQ2        .DS     1           ;
GAIN_VALUE      .DS     1           ;Gain value for PGA
KEY_NUM         .DS     1           ;Key number register for key matrix test
LED_BANK_1      .DS     1           ;PASS,FAIL,TIP LED'S
IO_PORT1_BANK_REG EQU   1           ;Bank register
PSW1_ATT        .DS     1           ;
PSW1_FRQ        .DS     1           ;
PSW1_FLAG       .DS     1           ;
REV_FLAG        .DS     1           ;Line reversal flag
RMS_DC_FLAG     .DS     1           ;1=DC measurement, 0=RMS measurement
START_PB_FLAG   .DS     1           ;Set/clear START pushbutton flag
TEST_NUM        .DS     1           ;Test number storage
ERR_PTR         .DS     2           ;Error buffer pointer
FILT_LOW        .DS     2           ;2130 filter low freqency value
FILT_HIGH       .DS     2           ;2130 filter high frequency value
FILT_NOM        .DS     2           ;2130 filter nominal peak value
MAX             .DS     2           ;Maximum A-to-D value
MIN             .DS     2           ;Minimum A-to-D value
MAX_BUF         .DS     2           ;Last error buffer address pointer
MSG_BUF         .DS     2           ;Error message pointer buffer
SAVX1           .DS     2           ;Index storage register #1
SAVX2           .DS     2           ;Index storage register #2
SAVX3           .DS     2           ;Index storage register #3
SUM             .DS     2           ;Sum area for for average routine
mMATRIX_MAX_ATT .DS     1
ERR_BUFF        .DS     $34         ;Error code storage buffer
ERR_BUFF_END:

;-------------------------------------------------------------------------
; LCD display message string table
;
;                       12345678901234567890
```

```
TEST_NUM_1:     .BYTE   ' Test #1            ',0
TEST_NUM_2:     .BYTE   ' Test #2            ',0
TEST_NUM_3:     .BYTE   ' Test #3            ',0
TEST_NUM_4:     .BYTE   ' Test #4            ',0
TEST_NUM_5:     .BYTE   ' Test #5            ',0
TEST_NUM_6:     .BYTE   ' Test #6            ',0
TEST_NUM_7:     .BYTE   ' Test #7            ',0
TEST_NUM_8:     .BYTE   ' Test #8            ',0
TEST_NUM_9:     .BYTE   ' Test #9            ',0
TEST_NUM_10:    .BYTE   ' Test #10           ',0
TEST_NUM_11:    .BYTE   ' Test #11           ',0
TEST_NUM_12:    .BYTE   ' Test #12           ',0
TEST_NUM_13:    .BYTE   ' Test #13           ',0
TEST_NUM_14:    .BYTE   ' Test #14           ',0
START_PB_MSG:   .BYTE   'Push START to advanc',0
FUNCT_PB_MSG:   .BYTE   'Push F1 to repeat   ',0
TIP08_MSG:      .BYTE   'Test #8: DTMF Gain  ',0
TIP09_MSG:      .BYTE   'Test #9: FSK Gain   ',0
TIP10_MSG:      .BYTE   'Test #10: 2130 1st  ',0
TIP10b_MSG:     .BYTE   'Test #10: 2130 2nd  ',0
TIP11_MSG:      .BYTE   'Test #11: 2750 Filtr',0
TIP12_MSG:      .BYTE   'Test #12 in progress',0
ADJ_DTMF_MSG:   .BYTE   'Adjust R152 on DUT  ',0
ADJ_FSK_MSG:    .BYTE   'Adjust R158 on DUT  ',0
ADJ_2130F1_MSG: .BYTE   'Adjust R156 on DUT  ',0
ADJ_2130F2_MSG: .BYTE   'Adjust R155 on DUT  ',0
ADJ_2750_MSG:   .BYTE   'Adjust R157 on DUT  ',0
GAIN_ERR_MSG:   .BYTE   'Gain adjust error...',0
SAMSUNG_MSG:    .BYTE   'Resetting SAMSUNG...',0
BLANK_MSG:      .BYTE   '                    ',0
KEY_#_MSG:      .BYTE   'Checking key "#     ',0
KEY_*_MSG:      .BYTE   'Checking key "*     ',0
KEY_0_MSG:      .BYTE   'Checking key "0     ',0
KEY_9_MSG:      .BYTE   'Checking key "9     ',0
KEY_8_MSG:      .BYTE   'Checking key "8     ',0
KEY_7_MSG:      .BYTE   'Checking key "7     ',0
KEY_6_MSG:      .BYTE   'Checking key "6     ',0
KEY_5_MSG:      .BYTE   'Checking key "5     ',0
KEY_4_MSG:      .BYTE   'Checking key "4     ',0
KEY_3_MSG:      .BYTE   'Checking key "3     ',0
KEY_2_MSG:      .BYTE   'Checking key "2     ',0
KEY_1_MSG:      .BYTE   'Checking key "1     ',0

START_MSG0:     .DB     ' CIDCO PCB-C TESTER ',$00
START_MSG1:     .DB     'Put PCB in fixture &',$00
START_MSG2:     .DB     'push button labeled ',$00
START_MSG3:     .DB     'START to start test.',$00

CHK_MSG0:       .DB     'CHECK PCB PLACEMENT ',$00
CHK_MSG1:       .DB     'IN FIXTURE and      ',$00
CHK_MSG2:       .DB     'Push START key to   ',$00
CHK_MSG3:       .DB     'start test.         ',$00

CHK1_MSG0:      .DB     ' Error: Handle open ',$00
CHK1_MSG1:      .DB     ' Close Handle &     ',$00
CHK1_MSG2:      .DB     ' Push START key to  ',$00
CHK1_MSG3:      .DB     ' continue test.     ',$00

CODE_MSG0:      .DB     ' CIDCO PCB-C TESTER ',$00
CODE_MSG1:      .DB     'Error code:'
CODE_MSG1A:     .DB                  '      #'
CODE_MSG1B:     .DB                       '  ',$00
CODE_MSG2:      .DB     '                    ',$00
CODE_MSG3:      .DB     ' ESC  INC  DEC  --- ',$00

TEST1_MSG0:     .DB     ' Model CT25CWi      ',$00
TEST1_MSG1:     .DB     'Test in progress....',$00
TEST1_MSG2:     .DB     ' Test #'
TEST1_MSG2A:    .DB           '              ',$00
TEST1_MSG3:     .DB     '                    ',$00

TEST2_MSG0:     .DB     ' Model CT102        ',$00
TEST2_MSG1:     .DB     'Test in progress....',$00
TEST2_MSG2:     .DB     'Test #              ',$00
TEST2_MSG3:     .DB     '                    ',$00

ERR_INTERNAL    .DB    'INTERNAL ERROR #'
ERR_INTERNALA   .DB                    '    ',$00
```

```
;                                       12345678901234567890  20 Column LCD Display.
ERR_MSG:
        .DB BOOT_UP_TEST_NUM,$01,    'STEALTH Boot-up err ',$00

        .DB PWR_TST_NUM,$01,         '-EPS not active  AC ',$00
        .DB PWR_TST_NUM,$02,         'VADPT not in spec AC',$00
        .DB PWR_TST_NUM,$03,         'VDD not in spec AC  ',$00
        .DB PWR_TST_NUM,$04,         'VCD not in spec AC  ',$00
        .DB PWR_TST_NUM,$05,         'VCCA not in spec AC ',$00
        .DB PWR_TST_NUM,$06,         'VCCAX not in spec AC',$00
        .DB PWR_TST_NUM,$07,         'VDDX not in spec AC ',$00
        .DB PWR_TST_NUM,$08,         'VMID not in spec AC ',$00
        .DB PWR_TST_NUM,$09,         'VMID2 not in spec AC',$00
        .DB PWR_TST_NUM,$0A,         'VCD not in spec Batt',$00
        .DB PWR_TST_NUM,$0B,         'VCD not in spec POTs',$00

        .DB DC_RES_TST_NUM,$01,      'DC resistance error ',$00
        .DB DC_RES_TST_NUM,$02,      'Line reversal error ',$00

        .DB RXD_GAIN_TST_NUM,$01,    'RxD Gain (HS) 500Hz ',$00
        .DB RXD_GAIN_TST_NUM,$02,    'RxD Gain (HS) 1200Hz',$00
        .DB RXD_GAIN_TST_NUM,$03,    'RxD Gain (HS) 2000Hz',$00

        .DB TXD_GAIN_TST_NUM,$01,    'TxD Gain (HS) 500Hz ',$00
        .DB TXD_GAIN_TST_NUM,$02,    'TxD Gain (HS) 1200Hz',$00
        .DB TXD_GAIN_TST_NUM,$03,    'TxD Gain (HS) 2000Hz',$00

        .DB LIU_TST_NUM,$01,         '-LIU stuck low      ',$00
        .DB LIU_TST_NUM,$02,         '-LIU not active     ',$00

        .DB KEY_MATRIX_TST_NUM,$01, 'Ky mtrx dst buf nclr',$00
        .DB KEY_MATRIX_TST_NUM,$02, 'DTMF key# receive er',$00
        .DB KEY_MATRIX_TST_NUM,$03, 'DTMF key# detect err',$00
        .DB KEY_MATRIX_TST_NUM,$04, 'DTMF key* receive er',$00
        .DB KEY_MATRIX_TST_NUM,$05, 'DTMF key* detect err',$00
        .DB KEY_MATRIX_TST_NUM,$06, 'DTMF key0 receive er',$00
        .DB KEY_MATRIX_TST_NUM,$07, 'DTMF key0 detect err',$00
        .DB KEY_MATRIX_TST_NUM,$08, 'DTMF key9 receive er',$00
        .DB KEY_MATRIX_TST_NUM,$09, 'DTMF key9 detect err',$00
        .DB KEY_MATRIX_TST_NUM,$0A, 'DTMF key8 receive er',$00
        .DB KEY_MATRIX_TST_NUM,$0B, 'DTMF key8 detect err',$00
        .DB KEY_MATRIX_TST_NUM,$0C, 'DTMF key7 receive er',$00
        .DB KEY_MATRIX_TST_NUM,$0D, 'DTMF key7 detect err',$00
        .DB KEY_MATRIX_TST_NUM,$0E, 'DTMF key6 receive er',$00
        .DB KEY_MATRIX_TST_NUM,$0F, 'DTMF key6 detect err',$00
        .DB KEY_MATRIX_TST_NUM,$10, 'DTMF key5 receive er',$00
        .DB KEY_MATRIX_TST_NUM,$11, 'DTMF key5 detect err',$00
        .DB KEY_MATRIX_TST_NUM,$12, 'DTMF key4 receive er',$00
        .DB KEY_MATRIX_TST_NUM,$13, 'DTMF key4 detect err',$00
        .DB KEY_MATRIX_TST_NUM,$14, 'DTMF key3 receive er',$00
        .DB KEY_MATRIX_TST_NUM,$15, 'DTMF key3 detect err',$00
        .DB KEY_MATRIX_TST_NUM,$16, 'DTMF key2 receive er',$00
        .DB KEY_MATRIX_TST_NUM,$17, 'DTMF key2 detect err',$00
        .DB KEY_MATRIX_TST_NUM,$18, 'DTMF key1 receive er',$00
        .DB KEY_MATRIX_TST_NUM,$19, 'DTMF key1 detect err',$00
        .DB KEY_MATRIX_TST_NUM,$1A, 'DTMF key# receive er',$00

        .DB DTMF_GAIN_TST_NUM,$01,   'DTMF gain adjust err',$00
        .DB DTMF_GAIN_TST_NUM,$02,   'MUTE.DIAL not muting',$00

        .DB FSK_GAIN_TST_NUM,$01,    '-CD stuck low error ',$00
        .DB FSK_GAIN_TST_NUM,$02,    'TONE.OUT gain error ',$00
        .DB FSK_GAIN_TST_NUM,$03,    '-CD not active error',$00
        .DB FSK_GAIN_TST_NUM,$04,    'FSK Gain error      ',$00
        .DB FSK_GAIN_TST_NUM,$05,    'FSK duty cycle error',$00

        .DB CAL_2130F1_TST_NUM,$01, '2130 filter sweep   ',$00

        .DB CAL_2750_TST_NUM,$01,    '2750 filter sweep   ',$00

        .DB CAS_DET_TST_NUM,$01,     'CASDET stuck high   ',$00
        .DB CAS_DET_TST_NUM,$02,     '-CASDET stuck low   ',$00
        .DB CAS_DET_TST_NUM,$03,     'CASDET not high CAS ',$00
        .DB CAS_DET_TST_NUM,$04,     '-CASDET not low CAS ',$00

        .DB RING_TST_NUM,$01,        '-RING stuck low     ',$00
        .DB RING_TST_NUM,$02,        '-RING not active    ',$00
        .DB RING_TST_NUM,$03,        'Piezo signal low err',$00
```

```
            .DB DIAL_TONE_TST_NUM,$01,   'DIAL.TONE stuck high',$00
            .DB DIAL_TONE_TST_NUM,$02,   'DIAL.TONE not active',$00
ERR_END:
                .DB     $00
INITIAL_END:
                .DB     $00
.END
```