TITLE PAGE

INTERACTIVE TOY

(FURBY.ASM - Version 25)

INVENTOR: Dave Hampton

Attorney Docket No. 64799
FITCH, EVEN, TABIN & FLANNERY
Suite 900
135 South LaSalle Street
Chicago, Illinois 60603-4277
Telephone (312) 372-7842

```
;ÉÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍ
ÍÍÍÍ»
;º
º
;º      SPC81A Source Code   (Version 25)
º
;º
º
;º      Written by: Dave Hampton  /  W   e Schulz
º
;º      Date:         July 30, 1998
º
;º
º
;º      Copyright (C) 1996,1997,1998 by Sounds Amazing!
º
;º      All rights reserved.
º
;ÈÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍ
ÍÍÍͼ
;
;
;*********************************************************************


;  remember   SBC    if there is a borrow carry is CLEARED
;  also SBC   if the two numbers are equal you still get a negative
result


;
;
;ÉÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍ
ÍÍÍÍ»
;º   MODIFICATION LIST :
º

;


; Furby29/30/31/32
;      Final testing for shipment of code on 8/2/98.
;      Tables updated,   tor speed updated, wake up/name fix
;      sequential tables never getting first entry,fixed.
;      New diag5.asm, Light3.asm (if light osc stalls it wont hang
system).
;
; Furby33
;      In motor brake routine, turn motors off before turning reverse
;      braking pulse on to save transistors.
;
; Furby34
;      Cleanup start code and wake routines.
;      Light sensor goes max dark and stays there to reff time, then
;      call sleep macro and shut down.
;
; Furby35
;      Adds four new easter eggs,BURP ATTACK, SAY NAME, TWINkLE SONG,
;      and ROOSTER LOVES YOU. Also add new names.
;
;
;
;::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::
```

```
; Release 3

;; File "testR3a"

; 1. Light sensor has a hysteresis point of continually triggering
sensor.
; 2. Light sensor decrements two instead of one on hungry counter.
; 3. Diagnostic mode for light sensor wont trigger very easily.
; 4. When a furby receives the I.R. sleep command he sends the same
command
;     out before going to sleep.
;
; 5. When hungry is low enough to trigger sick counter, each sensor
;     deducts two instead of one for each hit.
;
; 6. When diagnostics complete clear memory, reset hungry & sick to FF
;     randomly choose new name and voice, then write EEPROM before
;     going to sleep. Also extend EEPROM diagnostic to test all locations
;     for pass/fail of device.
;
; 7. Add new light routine

; 8. Change hide and seek egg to light,light,light,tummy.

; 9. Change sick/hungry counter so that it can only get so sick and
;     not continue down to zero. (MAX_SICK)

;10. In diagnostics, motor position test ,,,, first goes forward
continuously
;     until the front switch is pressed, then goes reverse continuously
;     until the front switch is pressed again, and then does normal
position
;     calibration stopping at the calibration switch.

;11. On power up we still use tilt and invert to generate startup random
;     numbers, but if feed switch is pressed for cold boot, we use it to
;     generate random numbers, because it is controlled by the user where
;     the tilt and invert are more flaky.

;12. No matter what age, 25% of time he randomly pulls speech from age 1
;     to generate more Furbish at older ages.

;13. Twinkle song egg
;     When song is complete, if both front and back switches are pressed
;     we goto deep sleep. That means only the invert can wake us up, not
;     the tilt switch.


;º
º
;ÈÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍ
ÍÍÍͼ
;****************************************************************
;****************************************************************
;****************************************************************
;****************************************************************
;****************************************************************
```

```
; Actual numeric value for TI pitch control

;  bit 7 set = subtract value from current course value
;        clr = add value to current course value
;  bit 6 set = select music pitch table
;        clr = select normal speech pitch table
;  bit 0-5 value to change course value (no change = 0)

; A math routine in 'say_0' converts the value for + or -
; if <80 then subtracts from 80 to get the minus version of 00
; ie, if number is 70 then TI gets sent 10 (which is -10)
; If number is 80 or > 80 then get sent literal as positive.

; NOTE: MAX POSITIVE IS 8F (+16 from normal voice of 00)
;       MAX NEGATIVE is 2F (-47 from normal voice of 00)

;This is a difference of 80h - 2Fh or 51h


; 8Fh is hi voice  (8f is very squeeeeeke)
; 2Fh lo voice ( very low)


; The math routine in 'Say_0' allows a +-decimal number in the speech
table.
; A value of 80 = no change or 00 sent to TI
; 81 = +1
; 8f = +16
;
;Avalue of 7F = -1 from normal voice
;70 = -16

; The voice selection should take into consideration that the hi voice
; selection plus an aditional offset is never greater than 8f
; Or a low voice minus offset never less than 2f.

Voice1      EQU   83h   ;(+3) hi voice
Voice2      EQU   7Ah   ;(-6) mid voice
Voice3      EQU   71h   ;(-15) low voice

;;;; we converted to a random selection table, but since all voice
tables
;    use the equates plus some offset, we make the change in the SAY_0
;    routine. We always assign voice 3 which is the lowest, and based on
;    the random power up pitch selection, the ram location 'Rvoice'
holds
;    the number to add to the voice+offset received from the macro
table.

Voice EQU   Voice3          ;pitch (choose Voice1, Voice2,
Voice3)(voice2=norm)

; Select Voice3 since it is the lowest and then add the difference to
get
; Voice2 or Voice3. Here we assign that difference to an equate to be
; used in the voice table that is randomly selected on power up.

S_voice1  EQU  18 ;Voice3 + 18d = Voice1
S_voice2  EQU  09 ;Voice3 + 09d = Voice2
```

```
S_voice3  EQU  0  ;Voice3 + 00d = Voice3


;*************************************************************************

; Motor speed pulse width :
; Motor_on = power to motor, Motor_off is none.


Mpulse_on   EQU   16  ;
Mpulse_off  EQU   16  ;


Cal_pos_fwd EQU   134   ;calibration switch forward direction
Cal_pos_rev EQU   134   ;calibration switch forward direction

;*************************************************************************
;*************************************************************************
;*************************************************************************
;*************************************************************************
;*************************************************************************
;
;ÚÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄ¿
;³                          PORTS                                      ³
;³ SPC40A has : 16 I/O pins                                            ³
;³ PORT_A 4 I/O pins  0-3                                              ³
;³ PORT_C 4 I/O pins  0-3                                              ³
;³ PORT_D 8 I/O pins  0-7                                              ³
;³                                                                     ³
;³                          RAM                                        ³
;³                                                                     ³
;³ SPC40A has : 128 bytes of RAM                                       ³
;³ from $80 - $FF                                                      ³
;³                                                                     ³
;³                          ROM                                        ³
;³ SPC40A has :                                                        ³
;³ BANK0 user ROM from $0600 - $7FFF                                   ³
;³ BANK1 user ROM from $8000 - $FFF9                                   ³
;³                                                                     ³
;³                                                                     ³
;³                          VECTORS                                    ³
;³ NMI   vector  $7FFA / $7FFB                                         ³
;³ RESET vector  $7FFC / $7FFD                                         ³
;³ IRQ   vector  $7FFE / $7FFF                                         ³
;ÀÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÙ
;ÚÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄ¿
;³                          PORTS                                      ³
;³ SPC120A has : 17 I/O pins                                           ³
;³ PORT_A 4 I/O pins  0-3                                              ³
;³ PORT_B 4 I/O pins  0,1,2,4,5                                        ³
;³ PORT_C 4 I/O pins  0-3 input only                                   ³
;³ PORT_D 8 I/O pins  0-7                                              ³
;³                                                                     ³
;³                          RAM                                        ³
;³ SPC120A has : 128 bytes of RAM                                      ³
;³ from $80 - $FF                                                      ³
;³                                                                     ³
;³                          ROM                                        ³
;³ SPC120A has :
```

```
;³ BANK0 user RO        $0600 - $7FFA·                          ³
;³ BANK1 user RC        $8000 - $FFFF                           ³
;³ BANK2 user RC        $10000 - $17FFF                         ³
;³ BANK3 user RC        $1A000 - $1FFFF                         ³
;³                                                              ³
;³                                                              ³
;³                        VECTORS                               ³
;³ NMI   vector  $7FFA / $7FFB                                  ³
;³ RESET vector  $7FFC / $7FFD                                  ³
;³ IRQ   vector  $7FFE / $7FFF                                  ³
;ÀÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÙ


; unuseable areas in rom

;SPC40A:      8000H ÄÄ  DFFFH should be skiped (Dummy area)
;  bank 0 = 600 - 7FFA
;  bank 1 = 8000 - DFFF reserved , start @ E000 - FFFA

;SPC80A:      10000H ÄÄ 13FFFH should be skiped (Dummy area)
;  bank 0 = 600 - 7FFA
;  bank 1 = 8000 - FFFA
;  bank 2 = 10000-13FFF reserved ,  start at 14000 - 17FFF

;SPC120A: ;SPC120A: 18000H ÄÄ 19FFFH should be skiped (Dummy area)
;  bank 0 = 600 - 7FFA
;  bank 1 = 8000 - FFFA
;  bank 2 = 10000 - 17FFF
;  bank 3 = 18000 - 19FFF reserved ,  start at 1A000 - 1FFFA

;SPC256A: ;SPC256A: Non dummy area

;SPC512A: ;SPC512A: Non dummy area

;*****************************************************************

.CODE
.SYNTAX  6502
.LINKLIST
.SYMBOLS


;ÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄ PORT DIRECTION CONTROL REGISTER
ÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄ
Ports_dir        EQU     00      ; (write only)
;
; (4 I/O pins) controlled with each bit of this register
; you can't control each pin separately, only as a nibble
; 0 = input / 1 = output
;
; 7      6      5      4      3      2      1      0         (REGISTER
BITS)
; D      D      C      C      B      B      A      A         (PORT)
; 7654   3210   7654   3210   7654   3210   7654   3210      (PORT BITS)
;ÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄ
ÄÄÄÄÄ

; ÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄ PORT CONFIGURATION CONTROL REGISTER
ÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄ
```

```
;                    based on if the port pin is input or output
;
Ports_con          EQU     01      ; (write only)
;
; (4 I/O pins) controlled with each bit of this register
; 7       6       5       4       3       2       1       0           (REGISTER
BITS)
; D       D       C       C       B       B       A       A           (PORT)
; 7654    3210    7654    3210    7654    3210    7654    3210        (PORT BITS)

; port_a INPUTS can be either:
; 0 = float    1 = pulled high

; port_a OUTPUTS can be either:
; 0 = buffer    1 = upper (4) bits Open drain Pmos (source)
;                   lower (4) bits Open drain Nmos (sink)
;
; port_b INPUTS can be either:
; 0 = float    1 = pulled low

; port_b OUTPUTS can be either:
; 0 = buffer    1 = upper (4) bits Open drain Nmos (sink)
;                   lower (4) bits Open drain Nmos (sink)
;
; port_c INPUTS can be either:
; 0 = float    1 = pulled high
; port_c OUTPUTS can be either:
; 0 = buffer    1 = upper (4) bits Open drain Pmos (source)
;                   lower (4) bits Open drain Nmos (sink)
;
; port_d INPUTS can be either:
; 0 = float    1 = pulled low
; port_d OUTPUTS can be either:
; 0 = buffer    1 = Open drain Pmos (source)

;AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAA

;AAAAAAAAAAAAAAAAAAAAAAAAAAA I/O PORTS
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA

Port_A             EQU     02H      ; (read/write) for TI & speech recgn
CPU's
Data_D0                 EQU     01H     ;bit 0 data nible port
Data_D1                 EQU     02H     ;
Data_D2                 EQU     04H     ;
Data_D3                 EQU     08H     ;

Port_B                  EQU     03H     ;b0/b1 = I/O  b4/b5 = inp only
TI_init                 EQU     01H     ;B0 - TI reset control
TI_CTS                  EQU     02H     ;B1 - hand shake to TI
IR_IN          EQU      10H     ;B4 - I.R. Recv data
TI_RTS                  EQU     20H     ;B5 - TI wants data

Port_C             EQU     04H      ; (read/write)
Motor_cal      EQU      01H     ;C0 - lo when mot   crosses switch
Pos_sen                 EQU     02H     ;C1 - moto    ical sensor (intt C1)
Touch_bck      EQU      04H     ;C2 - back touch
Touch_frnt     EQU      08H     ;C3 - front touch
```

```
Port_D          EQU     05H     ; (read/write)
Ball_side    EQU    01H    ;D0 - hi when on any side (TILT)
Ball_invert  EQU    02H    ;D1 - hi when inverted
Light_in     EQU    04H    ;D2 - hi when bright light hits sensor
Mic_in              EQU    08H    ;D3 - hi pulse microphone input
Power_on     EQU    10H    ;D4 - power to rest of circuit
Motor_led    EQU    20H    ;D5 - motor I.R. led driver
Motor_lt     EQU    40H    ;D6 - motor drive left (forward)
Motor_rt     EQU    80H    ;D7 - motor drive right (reverse)

;AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAA

;AAAAAAAAAAAAAAAAAAAAAAAAAA DATA LATCH  PORT_D
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
Latch_D          EQU     06H     ; (read)
; read to latch data from port_d, used for wake-up on pin change
;AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAA

;AAAAAAAAAAAAAAAAAAAAAAAAAA BANK SELECTION REGISTER
AAAAAAAAAAAAAAAAAAAAAAAAAAA
Bank             EQU     07H     ; (read/write)  x x x x x x x b
; 0 = bank 0, 1 = bank 1         ;               7 6 5 4 3 2 1 0
; only two banks in SPC40a
;AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAA

;AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA WAKE UP
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
Wake_up          EQU     08H     ; (read/write) x x x x x x x w
;                                               7 6 5 4 3 2 1.0

; w=(0=disable, 1=enable wake-up on port_d change)
; read to see if wake-up, or normal reset
; this is the only source for a wake-up
; Always reset stack on wake up.
;AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAA

;AAAAAAAAAAAAAA  AAAAAAAAAAAAAA SLEEP
AAAAAAAAAAAAAAAAA. AAAAAAAAAAA  AAAAAAAAAA
Sleep            E          09H     ; (write)       x x x x x x x s
;                                   ;               7 6 5 4 3 2 1 0
; s=(0=don't care, 1=sl e
; writting 1 to bit0, f   es sleep
;AAAAAAAAAAAAAAAAAAAAAA    AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAA

;AAAAAAAAAAAAAAAAAAAAAAAAA TIMER A CONTROL REGISTER
AAAAAAAAAAAAAAAAAAAAAAAAAAA
; this needs more work to understand DMH
TMA_CON             EQU     0BH     ; (write)
;
;
;                 7 6 5 4 3 2 1 0
;                 m x x x
;
;                 m= Timer one mode (0=Timer,1=Counter)
```

```
;
;           Bit3: IE1 Ä¿ IE1= 0: Counter clock= external clock from IOC2
;   .       Bit2: T1  Ä´     = 1, T1= 0: counter clock= CPUCLK/8192
;           Bit1: IE0 Ä´          T1= 1: counter clock= CPUCLK/65536
;           Bit0: T0  ÄÙ IE0= 0: Counter clock= external clock from IOC2
;                            = 1, T0= 0: counter clock= CPUCLK/4
;                                 T0= 1: counter clock= CPUCLK/64
;
;ÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄ
ÄÄÄÄÄ

;ÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄ INTERRUPTS
ÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄ
Interrupts        EQU      0DH     ; (read/write)
;
;         7 6 5 4 3 2 1 0
;         w m a b 3 2 1 e
;
;         w = (0=watch dog ON, power-on default) (1=watch dog OFF)
;         m = (0=Timer A generates NMI INT, 1=Timer A generates IRQ INT)
;         a = (0=Timer A interrupt off, 1=Timer A interrupt on)
;         b = (0=Timer B interrupt off, 1=Timer B interrupt on)
;         3 = (0=CPU CLK/1024 interrupt off,  1=CPU CLK/1024 interrupt
on)
;         2 = (0=CPU CLK/8192 interrupt off,  1=CPU CLK/8192 interrupt
on)
;         1 = (0=CPU CLK/65536 interrupt off, 1=CPU CLK/65536 interrupt
on)
;         e = (0=external interrupt off, 1=external interrupt on)
;              rising edge, from port_c bit1
;ÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄ
ÄÄÄÄÄ

;ÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄ TIMERS
ÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄ
; There are two 12bits timers.
; Timer A can be either a timer or a counter. (as set by TIMER_CON)
; Timer B can only be used as a timer.
;
; Timers count-up and on overflow from 0FFF to 0000, this carry bit will
; create an interrupt if the corresponding bit is set in INTERRUPTS
register.
; The timer will be auto reloaded with the user setup value, and
start,,,
; count-up again.
;
; Counter will reset by user loading #00 into register TMA_LSB and
TMA_MSB.
; Counter registers can be read on-the-fly, this will not affect
register,,,
; values, or reset them.
;
;ÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄ
ÄÄÄÄÄ

;ÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄ TIMER A (low byte
)ÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄ
TMA_LSB           EQU      10H     (read/write)
;
; all 8bits valid (lower 8bits of 12bit timer)
```

```
;AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAA

;AAAAAAAAAAAAAAAAAAAAAAAAA TIMER A (high byte)
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
TMA_MSB          EQU      11H      (read/write)
; read          x  x  x  x  11 10 9  8    timer upper 4bits
;               7  6  5  4  3  2  1  0
;
  rite          x  x  t  c  11 10 9  8    timer upper 4bits
                7  6  5  4  3  2  1  0    register bit

;                t=(0=speech mode, 1=Tone mode)
;                this connects the AUDA pin to either
;                the DAC , or Timer generated square wave
;
;                c=(0=CPU clock,   1=CPU clock/4:
;AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAA

;AAAAAAAAAAAAAAAAAAAAAAAAA TIMER B (low byte
)AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
TMB_LSB          EQU      12H
;
; all 8bits valid (lower 8bits of 12bit timer)
;AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAA

;AAAAAAAAAAAAAAAAAAAAAAAAA TIMER B (high byte)
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
TMB_MSB          EQU      13H
; read          x  x  x  x  11 10 9  8    timer upper 4bits
;               7  6  5  4  3  2  1  0
;
; write         x  x  t  c  11 10 9  8    timer upper 4bits
;               7  6  5  4  3  2  1  0    register bit
;
;                t=(0=speech mode, 1=Tone mode)
;                this connects the AUDB pin to either
;                the DAC2, or Timer generated square wave
;
;                c=(0=CPU clock,   1=CPU clock/4:
;AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAA

;AAAAAAAAAAAAAAAAAAAAAAAAAA  D/A converters
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
DAC1             EQU      14H      ; (write)
DAC2             EQU      15H      ; (write)
;AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAA

;AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAA
; this needs more work to understand DMH
;    16H    ADCoutputPort16H:

DAC_ctrl    EQU   16H

;
```

```
;            Bit7: I/O 0: Disable ADC; 1: Enable ADC
;            Bit6: I/O
;            Bit5: I/O
;            Bit4: I/O
;            Bit3: I/O
;            Bit2: I/O
;            Bit1: I/O
;            Bit0: I/O
;ÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄ
ÄÄÄÄÄ
;ÚÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄ¿
;³ Operating equate definition
;ÀÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÙ
;EQdef

; to calculate sampl
; CPU clk/sample rat         or
; Hi & Lo timer reg co       = FFF
; FFF - divisor = valu       oa  hi & lo reg.

;ex: 6mHZ clk = 166nSEC

;******** start Tracker


;/* here is som  definition chnge of time interrupt constant */Tracker

;SystemClock:    EQU     6000000          ;Select 6000000Hz it will be the
same
                                          ;as before
SystemClock:    EQU     3579545          ;Select 3579545Hz while we are
use that
                                          ;crystal

TimeA_low:      EQU     <(4096-(SystemClock/5859))       ;put constant
definition
TimeA_hi:       EQU     >(4096-(SystemClock/5859))

TimeB_low:      EQU     <(4096-(SystemClock/1465))
TimeB_hi:       EQU     >(4096-(SystemClock/1465))


;********* end Tracker


Port_def    EQU   A7h   ;D hi=out,D lo=inp / C hi=out,C lo=inp
                        ;B hi=inp,B lo=out / A hi=out,A lo=out

Con_def          EQU    50H   ;D hi=out buffer, D lo=in pull lo
;                       ;C hi=out buffer, C lo=in pull hi
;                       ;B hi=in  hi-Z  , B lo=out buffer
;                       ;A hi=out buffer, A lo=out buffer
;

Intt_dflt   EQU   D0h   ;sets interrupt reg = no watchdog,irq
                        ; timer B , and EXt port C bit 1 = off

;***** run EQU's

;*********************************************************************
```

```
; Send a braking pulse to stop motor drift, and this EQU is a decimal
number
; that determines how many times through the 2.9 mSec loop (how many
loops)
; the brake pulse is on. If attempting to make single count jumps, the
; brake pulse needs to be between 26 and 30. For any jump greater than
10
; braking between 22 and 80 is acceptable. ( Long jumps are not critical
; but short jump will begin to oscillate if braking is too great.)


; 60 long & 20 short work at 3.6v and no pulse width

Drift_long  EQU   60 ;number times thru intt before clearing pulse
Drift_short EQU   25 ;

;*************************************************************************

; set this with a number from 0 - 255 to determine timeout of all
sensors
; for the sequential increments. If it times out the table pointer
; goes back to the start, else each trigger increments through the
table.

; NOTE: this time includes the motor/speech execution time !!!

Global_time EQU   16    ; 1= 742 mSEC ;; 255 = 189.3 seconds

;*************************************************************************
; This determines how long Firby waits with no sensor activity, then
; calls the Bored_table for a random speech selection.
; Use a number between 1 & 255. Should probably not be less than 10.

; SHOULD BE > 10 SEC TO ALLOW TIME FOR TRAINING OF SENSORS

Bored_held  EQU   40    ; 1= 742 mSEC ;; 255 = 189.3 seconds

;*************************************************************************
;
; Each sensor has a sequential random split which must equal 16.
; Each sensor has a different assignment.
; The tables are formatted with the first X assignments random
; and the remaining as sequential.

Seq_front   EQU   8
Ran_front   EQU   8

Seq_back    EQU   9
Ran_back    EQU   7

Seq_tilt    EQU   10
Ran_tilt    EQU   6

Seq_invert  EQU   8
Ran_invert  EQU   8

Seq_sound   EQU   0
Ran_sound   EQU   16
```

```
Seq_light    EQU   0
Ran_light    EQU   16

Seq_feed     EQU   8
Ran_feed     EQU   8

Seq_wake     EQU   0
Ran_wake     EQU   16

Seq_bored    EQU   7
Ran_bored    EQU   9

Seq_hunger   EQU   5
Ran_hunger   EQU   11

Seq_sick     EQU   4
Ran_sick     EQU   12


; rev  furb11ja

; Each sensor also determines how often it is random or sequential
; as in 50/50  or 60/40  etc.
; These entries are subtracted from the random number generated
; and determine the split. (the larger here, the more likely sequential
pick)

Tilt_split   EQU   80h   ;
Invert_split       EQU   80h   ;
Front_split  EQU   80h   ;
Back_split   EQU   80h   ;
Feed_split   EQU   80h   ;
Sound_split  EQU   80h   ;
Light_split  EQU   80h   ;
Bored_split  EQU   80h   ;
Hunger_split       EQU   80h   ;
Sick_split   EQU   80h   ;

;********************************** **********************************

Random_age   EQU   30h   ;at any age, below this number when a
;                        random number is picked will cause him
;                        to pull from the age 1 table. More Furbish.

;*****************************************************************************

Learn_chg    EQU   31    ;amount to inc or dec training of words
;--------------------------------
Food         EQU   20h   ;amount to increase 'Hungry' for each feeding
Need_food    EQU   80h   ;below this starts complaining about hunger
Sick_reff    EQU   60h   ;below this starts complaining about sickness
Really_sick  EQU   C0h   ;below this only complains about sickness
Max_sick     EQU   80h   ;cant go below this when really sick

Hungry_dec   EQU   01    ;subtract X amount for each sensor trigger
Sick_dec     EQU   01    ;subtract X amount for each sensor trigger
;--------------------------------
Nt_word            EQU   FEH   ;turn speech word active off
Nt_last            EQU   FBH   ;bit 2 off - last word sent to TI
```

```
Nt_term             EQU   F7h   ;bit 3 off -terminator to speech TI
Clr_spch    EQU     FCH   ;clears spch_activ & word_activ
CTS_lo              EQU   FDH   ;makes TI_CTS go lo
;--------
Motor_rev   EQU     FDH   ;clears motor fwd bit
Motor_inactv        EQU   FEh   ;kill motor activ bit
Motor_ntseek        EQU   FBh   ;kill motor seek bit
Motor_off   EQU     C0h   ;turns both motor lines off (hi)
Motor_revs  EQU     7FH   ;bit 7 lo
Motor_fwds  EQU     BFh   ;bit 6 lo
Ntmot_on    EQU     DFh   ;clears motor pulse on req
Nt_IRQdn    EQU     F7h   ;clear IRQ stat
Nt_Motor_led        EQU   DFh   ;motor opto led off
Motor_led_rst       EQU   100   ;X * 2.9 millSec for shut off time

Nt_Init_motor       EQU   FBh   ;cks motor speed only on wake up
NT_Init_Mspeed      EQU   F7h   ;clears 2nd part of motor speed test

Opto_spd_reld       EQU   80    ;number of IRQ to count opto pulse speed
Speed_reff  EQU     30    ;value to adjust speed to

Nt_macro_actv       EQU   7Fh   ;clears request
;--------
Not_bside   EQU     F7h   ;clear ball side done flag
Not_binvrt  EQU     EFh   ;clear ball invert done flag
Not_tch_bk  EQU     BFh   ;clear touch back sense done flag
Not_tch_ft  EQU     DFh   ;clear touch back sense done flag
Not_feed    EQU     FDh   ;clear feed sense done flag
Sound_reload        EQU   05    ;X * 742 milisec time between trigger
Snd_cycle_rled      EQU   02    ;sound sense referrence cycle timer
;--------
Light_reload        EQU   07    ;X * 742 millsec until new reff level set
;--------
Nt_Slot_dn  EQU     FEh   ;clr IR slot low detected

Nt_lt_reff  EQU     EFh   ;turns reff off
Nt_lght_stat        EQU   FEh   ;clears light bright status to dim status

;;; Bright & Dim equates have been moved to the light include file.

;;;Bright   EQU     05    ;light sensor trigger > reff level
;;;Dim              EQU   05    ;Light sensor trigger < reff level

;--------
;Qik_snd_reload     EQU   01    ;
;Nt_snd_reff        EQU   DFh   ;kill sound reff level bit
Nt_do_snd   EQU     FEh   ;clears sound state change req
Nt_snd_stat EQU     FBh   ;clears Sound_stat
;--------
Nt_fortune  EQU     FEh   ;kills fortune teller mode
Nt_Rap              EQU   FDh   ;kills Rap mode
Nt_hideseek EQU     FBh   ;kills Hide & seek game mode
Nt_simon    EQU     F7h   ;kills simon say game mode
;--------
Nt_do_tummy EQU     F7h   ;clears sensor change req
Nt_do_back  EQU     EFh   ;clears sensor change req
Nt_do_feed  EQU     DFh   ;clears sensor change req
Nt_do_tilt  EQU     BFh   ;clears sensor change req
Nt_do_invert        EQU   7Fh   ;clears sensor change req
Nt_do_lt_brt        EQU   FDh   ;clears sensor change req
```

```
Nt_do_lt_dim        EQU   FBh   :clears sensor change req
;--------
Nt_temp_gam1        EQU   FEh   ;clears game mode bits
Nt_half_age EQU     BFh   ;clears req for 2 table instead of 4
Nt_randm    EQU     7Fh   ;clears random/sequential status

GameT_reload        EQU   24    ; 1= 742 mSEC ;; 255 = 189.3 seconds

;ÚÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄ¿
;³ Variable definition        (Ram = $80 to $FF)
;ÀÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÙ
;Rdef

;***** DO NOT CHANGE RAM ASSIGNMENTS (X pointer used as offsett)

;**************  The next group of RAM locations can be used by any
;             sensor routine but cannot be used to save data.
;             TEMP ONLY !
;******************** koball
TEMP0         equ   80h
TEMP1         equ   81h
TEMP2         equ   82h
TEMP3         equ   83h
TEMP4         equ   84h
IN_DAT              equ   85h
;******************** end koball
;* END TEMP RAM   **************

Task_ptr    EQU     86h   ;what function is in process
Port_A_image        EQU   87h   ;
Port_B_Image        EQU   88H   ;output port image
Port_D_Image        EQU   89H   ;output port image
;-------
Word_lo             EQU   8Ah   ;speech word lo adrs
Word_hi             EQU   8Bh   ; "         "   hi "
Saysent_lo  EQU     8CH   ;saysent word pointer
Saysent_hi  EQU     8DH   ; "         "      "
Bank_ptr    EQU     8EH   ;which bank words are in
Which_word  EQU     8FH   ;which word or saysent to call
Sgroup              EQU   90H   ;which saysent group table
Tx_data             EQU   91H   ;
;-------
Which_motor EQU     92h   ;holds table number of motor positon
Mgroup              EQU   93H   ;which motor group table
Motor_lo    EQU     94H   ;
Motptr_lo   EQU     95h   ;table pointer to get motor position
Motptr_hi   EQU     96H   ;
Which_delay EQU     97H   ;how much time between motor calls
Intt_Temp   EQU     98H   ;
Drift_fwd   EQU     99h   ;time motor reverses to stop drift
Drift_rev   EQU     9Ah   ;
Pot_timeL   EQU     9Bh   ;motor uses to compare against current positon

; moved to hi ram that is not cleared on power up
;Pot_timeL2

Moff_len    EQU     9Ch   ;holds motor power off pulse time
Mon_len             EQU   9Dh   ;holds motor power on pulse time
Motor_pulse1        EQU   9Eh   ;motor pulse timer
Slot_vote   EQU     9Fh   ;need majority cnt to declare a valid slot
```

```
motor_led_timer   EQU   A0h   ;how long after motion done led on for IR
Mot_speed_cnt     EQU   A1h   ;motor speed test
Mot_opto_cnt      EQU   A2h   ; "
Cal_switch_cnt    EQU   A3h   ;used to eliminate noisy reads
motorstoped equ   A4h   ;times wheel count when stopping
Drift_counter     EQU   A5h   ;decides how much braking pulse to apply
;-------
Mili_sec    EQU   A6h   ;used in calc pot position by timer
Cycle_timer EQU   A7h   ;bypasses intt port c updates to motor
Sensor_timer      EQU   A8h   ;times between sensor trigger
Bored_timer EQU   A9h   ;time with no activity to random speech
;-------
Invrt_count EQU   AAh   ;which speech/motor call is next
Tilt_count  EQU   ABh   ;which speech/motor call is next
Tchfrnt_count     EQU   ACh   ;which speech/motor call is next
Tchbck_count      EQU   ADh   ;which speech/motor call is next
Feed_count  EQU   AEh   ;which speech/motor call is next
;-------
Last_IR           EQU   AFh   ;last IR sample data to compare to next
Wait_time   EQU   B0h   ;used in IRQ to create 2.8mSec timers
;-------
Light_timer EQU   B1h   ;Light sensor routines
Lght_count  EQU   B2h   ;which speech/motor call is next
Light_reff  EQU   B3h   ;holds previous sample
;---------------
Sound_timer EQU   B4h   ;time to set new reff level
Sound_count EQU   B5h   ;which speech/motor call is next
;---------------
Milisec_flag      EQU   B6h   ;set every 742 miliseconds
Macro_Lo    EQU   B7h   ;table pointer
Macro_Hi    EQU   B8h   ;  "     "
Egg_cnt           EQU   B9h   ;easter egg table count pointer

;******************** Koball code rev B

HCEL_LO           EQU   BAh   ;
HCEL_HI           EQU   BBh   ;
BIT_CT            EQU   BCh   ;

;******************** end koball

Ligh_shift EQU    BDh   ;( was TMA_INT ) used for threshold change

;********************

Prev_random EQU   BEh   ;prevents random number twice in a row
Bored_count EQU   BFh   ;sequential selection for bored table
TEMP5       EQU   C0h   ;general use also used for wake up

Temp_ID2    EQU   C1h   ;use in sensor training routines
Temp_ID           EQU   C2h   ;use in sensor training routines
Learn_temp  EQU   C3h   ;use in sensor training routines


Req_macro_lo      EQU   C4h   ;holds last call to see if sleep or IR req
Req_macro_hi      EQU   C5h   ;

Sickr_count EQU   C6h   ;sequential counter for sick speech table
Hungr_count EQU   C7h   ;sequential counter for hunger speech table
```

```
Motor_pulse2        EQU   C8h   ;motor pulse timer


;***** DO NOT CHANGE BIT ORDER ******


Stat_0              Equ   C9h   ;System status
Want_name    EQU    01H   ;bit 0 =set forces system to say Furby's name
Lt_prev_dn   EQU    02H   ;bit 0 = done flag for quick light changes
Init_motor   EQU    04H   ;bit 1 = on wakeup do motor speed/batt test
Init_Mspeed EQU     08H   ;bit 3 = 2nd part of motor speed test
Train_Bk_prev       EQU   10H    ;bit 4 = set when 2 back sw hit in a row
Say_new_name        EQU   20H    ;bit 5 = only happens on cold boot
REQ_dark_sleep      EQU   40H    ;bit 6 = set -dark level sends to sleep
Dark_sleep_prev     EQU   80H    ;bit 7 = if set on wake up thendont
gotosleep
;
Stat_1              EQU   CAH    ;system status
Word_activ   EQU    01H   ;bit 0 = set during any speech
Say_activ    EQU    02H   ;bit 1 = when saysent is in process
Word_end     EQU    04H   ;bit 2 = set when sending FF word end to TI
Word_term    EQU    08H   ;bit 3 = set to send 3 #ffh to end speech
Up_light     EQU    10H   ;bit 4 =set when shift is incrmntg
Snd_reff     EQU    20H   ;bit 5 = set for new referrenc cycle
Half_age     EQU    40H   ;bit 6 = set for 2 tables of age instead of 4.
Randm_sel    EQU    80H   ;bit 7 =decides random/sequential for tables

Stat_2              EQU   CBH    ;system status more
Motor_actv   EQU    01H   ;bit 0 = set = motor in motion
Motor_fwd    EQU    02H   ;bit 1 = set=fwd clr=rev
Motor_seek   EQU    04H   ;bit 2 = seeking to next position
Bside_dn     EQU    08H   ;bit 3 = set = previously flaged
Binvrt_dn    EQU    10H   ;bit 4 = set- prev done
Tchft_dn     EQU    20H   ;bit 5 =  "     "
Tchbk_dn     EQU    40H   ;bit 6 =  "     "
Macro_actv   EQU    80H   ;bit 7 =set when macro in process
;
Stat_3              EQU   CCh    ;system status
Lght_stat    EQU    01H   ;bit 0 = set=bright clr = dim
Feed_dn             EQU   02H    ;bit 1 = set- prev done
Sound_stat   EQU    04H   ;bit 2 =  "     "
IRQ_dn              EQU   08H    ;bit 3 = set when IRQ occurs by IRQ
Lt_reff             EQU   10H    ;bit 4 =set for light sense reff cycle
Motor_on     EQU    20H   ;bit 5 = set=motor pulse power on
M_forward    EQU    40H   ;bit 6 = 1r = move motor forward
M_reverse    EQU    80H   ;bit 7 =clr = move motor reverse
;

;*****************************************************************

; Following bit maps are reserved for easter egg / games


Stat_4              EQU   CDh    ;system task request state
Do_snd              EQU   01H    ;bit 0 = set when sound > prev reff level
Do_lght_brt EQU     02H   ;bit 1 = set when light > prev reff level
Do_lght_dim EQU     04H   ;bit 2 = set when light < prev reff level
Do_tummy     EQU    08H   ;bit 3 = set when front touch triggered
Do_back             EQU   10H    ;bit 4 = set when back touch triggered
```

```
Do_feed             EQU   20H   ;bit 5 = set when feed sensor triggered
Do_tilt             EQU   40H   ;bit 6 = set when tilt sensor triggered
Do_invert    EQU    80H   ;bit 7 = set when inverted sensor triggered
;
Stat_5              Equ   CEh   ;game status
temp_gam1    EQU    01H   ;bit 0 =used in game play
temp_gam2    EQU    02H   ;bit 0 = "    "      "   "
temp_gam3    EQU    04H   ;bit 1 =
temp_gam4    EQU    08H   ;bit 3 =
temp_gam5    EQU    10H   ;bit 4 =
temp_gam6    EQU    20H   ;bit 5 =
temp_gam7    EQU    40H   ;bit 6 =
temp_gam8    EQU    80H   ;bit 7 =
;
Game_1              EQU   CFh   ;system game status
Fortune_mode        EQU   01H   ;bit 0 =set = furby in fortune teller mode
Rap_mode     EQU    02H   ;bit 0 =set = furby in RAP SONG mode
Hideseek_mode       EQU   04H   ;bit 1 =set = furby in hide & seek game
mode
Simonsay_mode       EQU   08H   ;bit 3 =set = furby in simon says game
mode
Burp_mode    EQU    10H   ;bit 4 =set = mode
Name_mode    EQU    20H   ;bit 5 =
Twinkle_mode        EQU   40H   ;bit 6 =
Rooster_mode        EQU   80H   ;bit 7 =
;
Qualify1:    EQU    D0h   ;easter egg disqualified when clear
DQ_fortune   EQU    01h   ;bit 0 = fortune teller
DQ_rap              EQU   02h   ;bit 1 = rap song
DQ_hide             EQU   04h   ;bit 2 = hide and seek
DQ_simon     EQU    08h   ;bit 3 = simon says
DQ_burp             EQU   10h   ;bit 4 = burp attack
DQ_name             EQU   20h   ;bit 5 = says his name
DQ_twinkle   EQU    40h   ;bit 6 = sings song
DQ_rooster   EQU    80h   ;bit 7 = rooster loves you
;

; ********** THIS GROUP OF RAM IS SAVED IN EEPROM


; Need to read these from EEPROM and do test for false data

; "age" uses bit 7 to extend the "age_counter" to 9 bits, and this
; is saved in EERPOm also.

;"AGE" MUST BE IN D1h BECAUSE EEPROM READ & WRITE USE THE EQU FOR START
RAM.

Age          EQU    D1h   ;age = 0-3 (4 total)
Age_counter  EQU    D2h   ;inc on motor action,rolls over & inc age

Name         EQU    D3h   ;holds 1-6 pointer to firby's name
Rvoice              EQU   D4h   ;which one of three voices
Pot_timeL2   EQU    D5h   ;counter from wheel I.R. sensor
Hungry_counter      EQU   D6h   ;holds hungry/full counter
Sick_counter        EQU   D7h   ;healthy/sick counter
Seed_1              EQU   D8h   ;only seed 1 & seed 2 are saved
Seed_2              EQU   D9h   ; "     "     "

; These are used for training each sensor. There is a word number which
```

```
; is 1-16 for the sesnor table macro list and a ram for count which
; determines how often to call the learned word.

; *** DO NOT CHANGE ORDER----- RAM adrs by Xreg offset

Tilt_learned      EQU   DAh   ;which word trained                  1
Tilt_lrn_cnt      EQU   DBh   ;count determines how often called  2

Feed_learned      EQU   DCh   ;which word trained                  3
Feed_lrn_cnt      EQU   DDh   ;count determines how often called  4

Light_learned     EQU   DEh   ;which word trained                  5
Light_lrn_cnt     EQU   DFh   ;count determine  how often called  6

Dark_learned      EQU   E0h   ;which word trained                  7
Dark_lrn_cnt      EQU   E1h   ;count determines how often called  8

Front_learned     EQU   E2h   ;which word trained                  9
Front_lrn_cnt     EQU   E3h   ;count determines how often called  10

Sound_learned     EQU   E4h   ;which word trained                  11
Sound_lrn_cnt     EQU   E5h   ;count determines how often called  12

Wake_learned      EQU   E6h   ;which word trained                  13
Wake_lrn_cnt      EQU   E7h   ;count determines how often called  14

Invert_learned    EQU   E8h   ;which word trained                  15
Invert_lrn_cnt    EQU   E9h   ;count determines how often called  16

; next is equates defining which ram to use for each sensor
; according to the sensor ram defined above. (compare to numbers above)

Tilt_ID           EQU   00    ;defines what offset for above ram
definitions
Feed_ID           EQU   02    ; "
Light_ID    EQU   04    ; "
Dark_ID           EQU   06    ; "
Front_ID    EQU   08    ; "
Sound_ID    EQU   10    ; "
Wake_ID           EQU   12    ; "
Invert_ID   EQU   14    ; "
Back_ID           EQU   EEh   ;special value triggers learn mode

;*****************************************************************************
*
; For power on test, WE only clear ram to E9h and use EAh for a
; messenger to the warm boot routine. We always clear ram and initialize
; registers on power up, but if it is a warm boot then read EEPROM
; and setup ram locations. Location EAH is set or cleared during power
up
; and then the stack can use it during normal run.


Warm_cold   EQU   EDh   ;
Spcl_seed1  EQU   EEh   ;
Spcl_seed2  EQU   EFh   ;
Deep_sleep  EQU   F0h   ;0=no deep sleep 11h is. (tilt wont wakeup)


;********** Need to allow stack growth down  ( EAh- FFH ) **********
```

```
Stacktop    EQU   FFH   ;Stack Top


;**********************************************************************
****
;**********************************************************************
****
;**********************************************************************
****
;**********************************************************************
****

        ORG     00H
        BLKW    300H,00H          ;Fill 0000 ÄÄÄ 05FFH= 00

;ÚÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄ¿
;³                                                        ³
;³       P R O G R A M    S T A R T S   H E R E           ³
;³                                                        ³
;ÀÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÙ

        ORG     0600H

RESET:


      Include           Wake2.asm   ;asm file



;********* end Tracker


; For power on test, WE only clear ram to E9h and use EAh for a
; messenger to the warm boot routine. We always clear ram and initialize
; registers on power up, but if it is a warm boot then read E"PROM
; and setup ram locations. Location EAH is set or cleared during power
up
; and then the stack can use it during normal run.


; Clear RAM to 00H
; ---------------------------------------------------------------------
----
        LDA     #00H              ; data for fill
        LDX     #E9H             ; start at ram location

RAMClear:
        STA     00,X              ; base 00, offset x
        DEX                       ; next ram location
        CPX     #7FH             ; check for end
        BNE     RAMClear          ; branch, not finished
                                  ; fill done
```

```
; ------------------------------------  ------------------------------
----


Main:

InitIO:
      LDA   #01          ;turn DAC on
      STA   DAC_ctrl     ;DAC control

      LDA   #Port_def    ;set direction control
      STA   Ports_dir    ;load reg

      LDA   #Con_def     ;set configuration
      STA   Ports_con    ;load reg

      LDA   #00          ;set for bank 0
      STA   Bank         ;set it
      LDA   #00H         ;disable wakeup control
      STA   Wake_up            ;
      LDA   #00h         ;disable sleep control
      STA   Sleep        ;set dont care

        LDA     #Intt_dflt     ;Initialize timers, etc.
        STA     Interrupts     ;load reg

        LDA     #00H           ;set timer mode
        STA     TMA_CON        ;set reg
      LDA   #TimeA_low  ;get preset timer for interrupts
      STA   TMA_LSB            ;load

      LDA   #TimeA_hi    ;get hi byte for preset
      STA   TMA_MSB            ;load it

      LDA   #TimeB_low   ;get preset timer for interrupts
      STA   TMB_LSB            ;load
      LDA   #TimeB_hi    ;get hi byte for preset
      STA   TMB_MSB            ;load it

      LDA   #C0h         ;preset status for motors off
      STA   Stat_3             ;

      LDA   #00H         ;init ports
      STA   Port_A             ;output

      LDA   #33H         ;init ports
      STA   Port_B_Image       ;ram image
      STA   Port_B             ;output

      LDA   #0FH         ;init ports
      STA   Port_C             ;output

      LDA   #D0H         ;init ports
      STA   Port_D_Image       ;ram image
      STA   Port_D             ;output

      LDA   #FFh         ;milisec timer reload value
      STA   Mili_sec     ;also preset IRQ timer

        CLI              ;Enable IRQ
```

```
      JSR   Kick_IRQ     ;wait for interrupt to restart

      JSR   TI_reset     ;go init TI  (uses 'Cycle_timer')


; Preset motor speed, assuming mid battery life, we set the pulse width
; so that the motor wont be running at 6 volts and burn out. We then
; predict what the pulse width should be for any voltage.

;     LDA   #Mpulse_on   ;preset motor speed
      LDA   #11
      STA   Mon_len            ;set motor on pulse timing

      LDA   #05          ;
      STA   Moff_len     ;set motor off pulse timing


;ÉÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍ»  ÉÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍ
;º 'Diagnostics and calibration Routine                                    º
;ÈÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍ¼  ÈÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍ
;

      Include            Diag7.asm   ;asm file


; ****** Only called by diagnostic speech routines *********

; Be sure to set 'MACRO_HI' and all calls are in that 128 byte block.

Diag_macro:
      STA   Macro_Lo     ;save lo byte of Macro table entry
        LDA     #0b8h    ;#90h              ;hex offset to adrs 400 added
to diag call
      CLC
      ADC   Macro_Lo     ;add in offset
      STA   Macro_Lo     ;update
      LDA   #01          ;get hi byte adrs 400 = 190h
      STA   Macro_Hi     ;save hi byte of Macro table entry
      JSR   Get_macro    ;go start motor/speech
      JSR   Notrdy             ;Do / get  status for speech and motor
      RTS                ;yo !


; Enter with Areg holding how many 30 mili second delay cycles

Half_delay:
      STA   TEMP1        ;save timer
Half_d2:
      LDA   #10          ;set 1/2 sec       (y * 2.9 mSec)
      STA   Cycle_timer  ;set it
Half_d3:
      LDA   Cycle_timer  ;ck if done
      BNE   Half_d3            ;loop
      DEC   TEMP1        ;
      BNE   Half_d2            ;loop
      RTS                ; done
```

```
Test_byp:   ;We assume diagnostic only runs on coldboot

;*****************************************************************

      LDA   #FFh          ;initialize word training variable
      STA   Temp_ID             ;

      LDA   #FFh          ;
      STA   Hungry_counter      ;preset furby's health
      STA   Sick_counter

;*****************************************************************

; We sit here and wait for tilt to go away, and just keep incrementing
; counter until it does. This becomes the new random generator seed.

Init_rnd:
      INC   TEMP1         ;random counter
      LDA   Port_D              ;get switches
      AND   #03           ;check tilt & invert sw
      BNE   Init_rnd      ;loop til gone
      LDA   TEMP1         ;get new seed
      STA   Spcl_seed1    ;stuff it
      STA   Seed_1              ;also load for cold boot

;*****************************************************************

; Use feed sw to generate a better random number

      JSR   Get_feed      ;go test sensor
      LDA   Stat_4              ;get system
      AND   #Do_feed      ;ck sw
      BNE   Feed_rnd      ;if feed sw then cold boot
      JMP   End_coldinit        ;else do warm boot
Feed_rnd:
      INC   TEMP1         ;random counter
      LDA   Stat_4              ;system
      AND   #DFh          ;clear any prev feed sw senses
      STA   Stat_4              ;update
      JSR   Get_feed      ;go test sensor
      LDA   Stat_4              ;get system
      AND   #Do_feed      ;ck sw
      BNE   Feed_rnd      ;wait for feed to go away
      LDA   TEMP1         ;get new seed
      STA   Spcl_seed1    ;stuff it
      STA   Seed_1              ;also load for cold boot


;*****************************************************************

;; IF this is a cold boot , reset command then clear EEPROM and
;  chose a new name and voice.

Do_cold_boot:

      LDA   #00
      STA   Warm_cold     ;flag cold boot
```

```
        LDA     Stat_0              ;system
        ORA     #Say_new_name       ;make system say new name
        STA     Stat_0              ;

;*******  NOTE :::::
;
;   VOICE AND NAME SLECTION MUST HAPPEN BEFORE EEPROM WRITE OR
;   THEY WILL ALWAYS COME UP  00    because ram just got cleared!!!!!!


; Random voice selection here

        LDA     #80h        ;get random/sequential split
        STA     IN_DAT              ;save for random routine

        LDX     #00         ;make sure only gives random
        LDA     #10h        ;get number of random selections
        JSR     Ran_seq             ;go get random selection

        TAX
        LDA     Voice_table,X       ;get new voice
        STA     Rvoice              ;set new voice pitch


;*****************************************************************************
*


; On power up or reset, Furby must go select a new name ,,, ahw how
cute.

        JSR     Random              ;
        AND     #1Fh        ;get 32 possible
        STA     Name        ;set new name pointer
        JSR     Do_EE_write ;write the EEPROM

End_coldinit:



;£ííííííííííííííííííííííííííííííííííííííííííííííííííííííííííííííííííí
;º 'Special initialization prior to normal run mode                  º
;º  Jump to Warm_boot when portD wakes us up
;Èíííííííííííííííííííííííííííííííííííííííííííííííííííííííííííííííííííí
;

Warm_boot:  ;normal start when Port_D wakes us up.

        JSR     S_EEPROM_READ       ;read data to ram

;Eprom_read_byp:

;*****************************************************************
; If light osc fails, or too dark and that sends us to sleep, we
; set 'Dark_sleep_prev' and save it in EEPROM in 'Seed_2'.
; when the sleep routine executes,(00 01 based on this bit)
; When we wake up we recover this bit and it becomes the previous done
; flag back in 'Stat_0', so that if the osc is
```

```
; still dark or failed, Furby wont go back to sleep.

      LDA   Seed_2             ;from EEPROM
      BEQ   No_prevsleep       ;jump if none
      LDA   Stat_0             ;system
      ORA   #Dark_sleep_prev ;prev done
      STA   Stat_0             ;update

No_prevsleep:

;*****************************************************************


      LDA   Spcl_seed1  ;recover start up random number
      STA   Seed_1             ;set generator

;*****************************************************************


; Pot_timeL2 is save in ram through sleep mode and then reloaded t
; Pot_timeL which is the working register for the motor position.
; This allows startup routines to clear ram without forgetting the
; last motor position.

      LDA   Pot_timeL2  ;get current count
      STA   Pot_timeL   ;save in motor routine counter

;*************************

; Get age and make sure it is not greater than 3 (age4)

      LDA   Age         ;get current age
      AND   #83h     ;preserve bit 7 which is 9th age counter bit
;;;;;              and insure age not >3

      STA   Age         ;set system

;*************************

      LDA   #Bored_reld ;reset timer
      STA   Bored_timer ;


      LDA   #03         ;set timer
      STA   Last_IR           ;timer stops IR from hearing own IR xmit

      JSR   Get_light   ;go get light level sample
      LDA   TEMP1       ;get new count
      STA   Light_reff  ;update system

;*****************************************************************
*

      LDA   Warm_cold   ;decide if warm or cold boot
      CMP   #11h        ;ck for warm boot
      BEQ   No_zero           ;jump if is
```

```
        LDA   #00           ;point to macro 0 (SENDS TO SLEEP POSITION)
        STA   Macro_Lo
        STA   Macro_Hi
        JSR   Get_macro     ;go start motor/speech
        JSR   Notrdy              ;Do / get  status for speech and motor

No_zero:


        LDA   #11           ;preset motor speed
        STA   Mon_len             ;set motor on pulse timing

        LDA   #05           ;set motor to 3/4 speed for speed test
        STA   Moff_len      ;set motor off pulse timing
;
;
        LDA   #00           ;clear all system sensor requests
        STA   Stat_4              ;update


;  Currently uses 4 tables, one for each age.

        LDA   Stat_0              ;system
        ORA   #Init_motor ;flag motor to do speed test
        ORA   #Init_Mspeed        ;2nd part of test
        STA   Stat_0              ;update


;*************************************************************

; Do wake up routine :

        lda   #Global_time        ;reset timer to trigger sensor learning
        STA   Sensor_timer        ;

        LDA   #80h          ;get random/sequential split
        STA   IN_DAT              ;save for random routine

        LDX   #00h          ;make sure only gives random
        LDA   #10h          ;get number of random selections
        JSR   Ran_seq             ;go get random selection
        LDA   TEMP1         ;get decision

        STA   IN_DAT              ;save decision
        LDA   #Wake_ID      ;which ram location for learned word count
(offset)
        JSR   Start_learn ;go record training info
        LDA   IN_DAT              ;get back word to speak

        JSR   Decid_age     ;do age calculation for table entry
        LDX   TEMP0         ;age offset
        LDA   Wakeup_S1,X ;get new sound/word
        STA   Macro_Lo      ;save lo byte of Macro table entry
        INX                 ;
        LDA   Wakeup_S1,X ;get new sound/word
        STA   Macro_Hi      ;save hi byte of Macro table entry
        JMP   Start_macro ;go start speech

;*************************************************************
```

```
;É'IDLE Routine                                                    º
;ÈÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍ
;
Idle:

; Idle routine is the time slice task master  (TSTM) ugh!
; We must call each routine and interleave with a call to speech
; to insure we never miss a TI request for data.

      JSR   Notrdy            ;Do / get  status for speech and motor

;*************************************************************
; THis bit is set when light sensor is darker than 'Dark_sleep'

      LDA   Stat_0            ;system
      AND   #REQ_dark_sleep   ;ck for req
      BEQ   No_dark_req ;jump if not

      LDA   Stat_0            ;system
      AND   #BFh        ;kill req
      STA   Stat_0            ;update

      LDA   #A6h        ;sleep macro
      STA   Macro_Lo
      LDA   #00h        ;sleep macro
      STA   Macro_Hi    ;
      JMP   Start_macro ;go say it


No_dark_req:

;*************************************************************

; When any sensor or timer calls the "start_macro" routine, the
; Macro_Lo & Macro_Hi are saved. Everyone jumps back to Idle and when
; speech/motor routines are finished, this routine will look at the
; macros that were used and execute another function if a match is
found.
;
; Checks for his name first, then any IR to send, and finally, the sleep
; commands. THe temp macro buffers are cleared before


;
Spcl_Name1:
      LDX   #00         ;offset
Spcl_Name2:
      LDA   Ck_Name_table,X   ;ck lo byte
      CMP   #FFh        ;ck for end of table (note 255 cant execute)
      BEQ   Spcl_IR1    ;done if is
      CMP   Req_macro_lo      ;ck against last speech request
      BNE   Not_Name2   ;jump if not
      INX               ;to hi byte
      LDA   Ck_Name_table,X   ;ck hi byte
      CMP   Req_macro_hi      ;ck against last speech request
```

```
      BNE   Not_Name3   ;jump if not
      JMP   Say_Sname   ;speak it
Not_Name2:
      INX               ;
Not_Name3:
      INX               ;
      JMP   Spcl_Name2  ;loop til done

Say_Sname:
      LDA   Stat_0
      AND   #DFh        ;kill req for startup new name
      STA   Stat_0            ;update

      LDA   Name        ;current setting for table offset
      CLC
      ROL   A           ;2's comp
      TAX
      LDA   Name_table,X      ;get lo byte
      STA   Macro_Lo    ;save lo byte of Macro table entry
      INX               ;
      LDA   Name_table,X      ;get hi byte
      STA   Macro_Hi    ;save hi byte of Macro table entry
      JSR   Get_macro   ;go start motor/speech
      JSR   Notrdy            ;Do / get  status for speech and motor
;
Spcl_IR1:
      LDX   #00         ;offset
Spcl_IR2:
      LDA   IRxmit_table,X    ;ck lo byte
      CMP   #FFh        ;ck for end of table (note 255 cant execute)
      BEQ   Spcl_IR_dn  ;done if is
      CMP   Req_macro_lo      ;ck against last speech request
      BNE   Not_IRxmit2 ;jump if not
      INX               ;to hi byte
      LDA   IRxmit_table,X    ;ck hi byte
      CMP   Req_macro_hi      ;ck against last speech request
      BNE   Not_IRxmit3 ;jump if not
      INY               ;point to IR table
      LDA   IRxmit_table,X    ;
      STA   TEMP2       ;xmit temp ram
      LDA   #FDh        ;TI command for IR xmit
      STA   TEMP1       ;
      JSR   Xmit_TI           ;go send it

      LDA   #Bored_reld ;reset bored timer
      STA   Bored_timer ;

      LDA   #03         ;set timer
      STA   Last_IR           ;timer stops IR from hearing its own IR
xmit

      JMP   Spcl_IR_dn  ;done - ola ......
Not_IRxmit2:
      INX               ;lo byte
Not_IRxmit3:
      INX               ;hi byte
      INX               ;xmit pointer
      JMP   Spcl_IR2    ;loop til done
Spcl_IR_dn:
;
```

```
;
Spcl_macro1:
      LDX   #00          ;offset
Spcl_sleep1:
      LDA   Sleepy_table,X    ;ck lo byte
      CMP   #FFh         ;ck for end of table (note 255 cant execute)
      BEQ   Ck_macro_dn ;done if is
      CMP   Req_macro_lo      ;ck against last speech request
      BNE   Not_sleepy2 ;jump if not
      INX               ;to hi byte
      LDA   Sleepy_table,X    ;ck hi byte
      CMP   Req_macro_hi      ;ck against last speech request
      BNE   Not_sleepy3 ;jump if not
      LDA   #00          ;clear macro pointers for wake up
      STA   Req_macro_lo
      STA   Req_macro_hi

;mod F-rels2 ;
;   Before going to sleep send sleep cmnd to all others.

      LDA   #15          ;
      STA   TEMP2        ;xmit temp ram
      LDA   #FDh         ;TI command for IR xmit
      STA   TEMP1        ;
      JSR   Xmit_TI           ;go send it

;need to wait >600 milisec before going to sleep because we arent using
;busy flags from TI and need to make sure it is done transmitting the
;I.R. code, the sleep routine kills the TI and it would never send the
cmnd.

      LDA   #25          ;how many 30 milisec cycles to call
      JSR   Half_delay   ;do 30milisec delay cycles

;end mod

      JMP   GoToSleep    ;nity-night

Not_sleepy2:
      INX                ;
Not_sleepy3:
      INX                ;
      JMP   Spcl_sleep1 ;loop til done
;
Ck_macro_dn:
      LDA   #00          ;clear macro pointers for wake up
      STA   Req_macro_lo
      STA   Req_macro_hi
      JMP   Test_new_name     ;on to task master
;

;;;;;;; SLEEP TABLE & IR table ..... MOVE TO INCLUDE FILE LATER

Sleepy_table:
      DW    91    ;hangout

      DW    166   ;wake up
      DW    167   ;wake up
      DW    168   ;wake up
      DW    169   ;wake up
```

```
        DW      258     ;Back sw
        DW      259     ;Back sw
        DW      260     ;Back sw

        DW      403     ;IR
        DW      413     ;IR
        DW      429     ;IR

        DB      FFh,FFh     ;FF FF   is table terminator

IRxmit_table:
        DW      [illegible]     ;trigger macro
        DB      00      ;which IR command to call ( 0 - 0f )
        DW      13      ;trigger macro
        DB      00      ;which IR command to call ( 0 - 0f )
        DW      17      ;trigger macro
        DB      00      ;which IR command to call ( 0 - 0f )
        DW      19      ;trigger macro
        DB      00      ;which IR command to call ( 0 - 0f )
        DW      26      ;trigger macro
        DB      00      ;which IR command to call ( 0 - 0f )
        DW      29      ;trigger macro
        DB      00      ;which IR command to call ( 0 - 0f )
        DW      33      ;trigger macro
        DB      00      ;which IR command to call ( 0 - 0f )
        DW      34      ;trigger macro
        DB      00      ;which IR command to call ( 0 - 0f )
        DW      44      ;trigger macro
        DB      00      ;which IR command to call ( 0 - 0f )
        DW      45      ;trigger macro
        DB      00      ;which IR command to call ( 0 - 0f )
        DW      48      ;trigger macro
        DB      00      ;which IR command to call ( 0 - 0f )
        DW      50      ;trigger macro
        DB      00      ;which IR command to call ( 0 - 0f )
        DW      55      ;trigger macro
        DB      00      ;which IR command to call ( 0 - 0f )
        DW      60      ;trigger macro
        DB      00      ;which IR command to call ( 0 - 0f )
        DW      149     ;from rooster wake up
        DB      00      ;

        DW      352     ;trigger macro
        DB      01      ;which IR command to call ( 0 - 0f )
        DW      363     ;trigger macro
        DB      01      ;which IR command to call ( 0 - 0f )
        DW      393     ;trigger macro
        DB      01      ;which IR command to call ( 0 - 0f )

        DW      248     ;trigger macro
        DB      02      ;which IR command to call ( 0 - 0f )
        DW      313     ;trigger macro
        DB      02      ;which IR command to call ( 0 - 0f )

        DW      86      ;trigger macro
        DB      03      ;which IR command to call ( 0 - 0f )
        DW      93      ;trigger macro
        DB      03      ;which IR command to call ( 0 - 0f )
        DW      339     ;trigger macro
```

```
DB    03    ;which IR command to call ( 0 - 0f )
DW    344   ;trigger macro
DB    03    ;which IR command to call ( 0 - 0f )
DW    351   ;trigger macro
DB    03    ;which IR command to call ( 0 - 0f )

DW    404   ;trigger macro
DB    04    ;which IR command to call ( 0 - 0f )
DW    405   ;trigger macro
DB    04    ;which IR command to call ( 0 - 0f )

DW    293   ;trigger macro
DB    05    ;which IR command to call ( 0 - 0f )
DW    394   ;trigger macro
DB    05    ;which IR command to call ( 0 - 0f )
DW    406   ;trigger macro
DB    05    ;which IR command to call ( 0 - 0f )
DW    414   ;trigger macro
DB    05    ;which IR command to call ( 0 - 0f )
DW    422   ;trigger macro
DB    05    ;which IR command to call ( 0 - 0f )

DW    395   ;trigger macro
DB    06    ;which IR command to call ( 0 - 0f )
DW    421   ;trigger macro
DB    06    ;which IR command to call ( 0 - 0f )
DW    423   ;trigger macro
DB    06    ;which IR command to call ( 0 - 0f )

DW    296   ;trigger macro
DB    07    ;which IR command to call ( 0 - 0f )
DW    415   ;trigger macro
DB    07    ;which IR command to call ( 0 - 0f )
DW    416   ;trigger macro
DB    07    ;which IR command to call ( 0 - 0f )

DW    288   ;trigger macro
DB    08    ;which IR command to call ( 0 - 0f )

DW    11    ;trigger macro
DB    09    ;which IR command to call ( 0 - 0f )
DW    12    ;trigger macro
DB    09    ;which IR command to call ( 0 - 0f )
DW    27    ;trigger macro
DB    09    ;which IR command to call ( 0 - 0f )
DW    42    ;trigger macro
DB    09    ;which IR command to call ( 0 - 0f )
DW    57    ;trigger macro
DB    09    ;which IR command to call ( 0 - 0f )
DW    235   ;trigger macro
DB    09    ;which IR command to call ( 0 - 0f )
DW    236   ;trigger macro
DB    09    ;which IR command to call ( 0 - 0f )
DW    237   ;trigger macro
DB    09    ;which IR command to call ( 0 - 0f )
DW    238   ;trigger macro
DB    09    ;which IR command to call ( 0 - 0f )
DW    261   ;trigger macro
DB    09    ;which IR command to call ( 0 - 0f )
DW    262   ;trigger macro
```

```
DB    09    ;which IR command to call ( 0 - 0f )
DW    396   ;trigger macro
DB    09    ;which IR command to call ( 0 - 0f )
DW    409   ;trigger macro
DB    09    ;which IR command to call ( 0 - 0f )

DW    399   ;trigger macro
DB    10    ;which IR command to call ( 0 - 0f )
DW    407   ;trigger macro
DB    10    ;which IR command to call ( 0 - 0f )
DW    408   ;trigger macro
DB    10    ;which IR command to call ( 0 - 0f )

DW    272   ;trigger macro
DB    11    ;which IR command to call ( 0 - 0f )
DW    273   ;trigger macro
DB    11    ;which IR command to call ( 0 - 0f )
DW    274   ;trigger macro
DB    11    ;which IR command to call ( 0 - 0f )
DW    275   ;trigger macro
DB    11    ;which IR command to call ( 0 - 0f )
DW    400   ;trigger macro
DB    11    ;which IR command to call ( 0 - 0f )
DW    418   ;trigger macro
DB    11    ;which IR command to call ( 0 - 0f )
DW    425   ;trigger macro
DB    11    ;which IR command to call ( 0 - 0f )
DW    426   ;trigger macro
DB    11    ;which IR command to call ( 0 - 0f )

DW    336   ;trigger macro
DB    12    ;which IR command to call ( 0 - 0f )
DW    342   ;trigger macro
DB    12    ;which IR command to call ( 0 - 0f )
DW    401   ;trigger macro
DB    12    ;which IR command to call ( 0 - 0f )

DW    92    ;trigger macro
DB    13    ;which IR command to call ( 0 - 0f )
DW    411   ;trigger macro
DB    13    ;which IR command to call ( 0 - 0f )
DW    419   ;trigger macro
DB    13    ;which IR command to call ( 0 - 0f )
DW    427   ;trigger macro
DB    13    ;which IR command to call ( 0 - 0f )

DW    291   ;trigger macro
DB    14    ;which IR command to call ( 0 - 0f )
DW    402   ;trigger macro
DB    14    ;which IR command to call ( 0 - 0f )
DW    412   ;trigger macro
DB    14    ;which IR command to call ( 0 - 0f )
DW    428   ;trigger macro
DB    14    ;which IR command to call ( 0 - 0f )

DW    256   ;trigger macro
DB    15    ;which IR command to call ( 0 - 0f )
DW    257   ;trigger macro
DB    15    ;which IR command to call ( 0 - 0f )
DW    420   ;trigger macro
```

```
      DB    15    ;which IR command to call ( 0 - 0f )

;mod F-rels2 ; send sleep if recv sleep on IR

      DW    403   ;trigger macro
      DB    15    ;which IR command to call ( 0 - 0f )
      DW    413   ;trigger macro
      DB    15    ;which IR command to call ( 0 - 0f )
; end mod

      DB    FFh,FFh     ;FF FF   is table terminator


Ck_Name_table:

      DW    97
      DW    248
      DW    393
      DW    414
      DW    149
      DW    305
      DW    404
      DW    421

      DB    FFh,FFh     ;FF FF   is table terminator

;*************************************************************

; Say name

Test_new_name:

      LDA   Stat_0              ;system
      AND   #Say_new_name       ;make system say new name
      BEQ   Nosayname    ;bypass if clear
      LDA   Stat_0
      AND   #DFh          ;kill req for startup new name
      STA   Stat_0              ;update

      LDA   Name          ;current setting for table offset
      CLC
      ROL   A             ;2's comp
      TAX
      LDA   Name_table,X        ;get lo byte
      STA   Macro_Lo     ;save lo byte of Macro table entry
      INX                ;
      LDA   Name_table,X        ;get hi byte
      STA   Macro_Hi     ;save hi byte of Macro table entry
      JSR   Get_macro    ;go start motor/speech
      JSR   Notrdy              ;Do / get  status for speech and motor

Nosayname:


;*************************************************
;
;
; ***** below routines run at 742 mSec loops
; Timer B sets 'Milisec_flag' each 742 miliseconds
```

```
Updt_timer:
      LDA   Milisec_flag      ;if >0 then 742 mili seconds have passed
      BEQ   TimerL_dn   ;bypass if 0
      LDA   #00         ;clear it
      STA   Milisec_flag      ;reset

      LDA   Sensor_timer      ;get current timer * 742mSec sec
      BEQ   TimerL1           ;do nothing if 0
      DEC   Sensor_timer      ;-1
TimerL1:
      LDA   Light_timer ;get current timer * 742mSec sec
      BEQ   TimerL2           ;do nothing if 0
      DEC   Light_timer ;-1
TimerL2:
      LDA   Sound_timer ;get current timer * 742mSec sec
      BEQ   TimerL3           ;do nothing if 0
      DEC   Sound_timer ;-1
TimerL3:
      LDA   Bored_timer ;get current timer * 742mSec
      BEQ   TimerL4           ;do nothing if 0
      DEC   Bored_timer ;-1
TimerL4:
      LDA   Last_IR           ;get current timer * 742mSec
      BEQ   TimerL5           ;do nothing if 0
      DEC   Last_IR           ;-1
TimerL5:

TimerL_dn:

      INC   Task_ptr    ;+1
      LDA   Task_ptr    ;get it
      CLC
      SBC   #08         ;ck if off end
      BCC   Ck_tsk_A    ;jump if <9
      LDA   #01         ;reset pointer
      STA   Task_ptr    ;

Ck_tsk_A:

; If too sick then no game play...

      CLC               ;
      LDA   Sick_counter      ;how sick is he
      SBC   #Really_sick      ;
      BCS   Ck_task_egg ;do egg if not
      JMP   Ck_bored    ;bypass if too sick


; Scan all game mode pointers to determine if any are active.
; Continue to execute the first active game found, and that game always
; allows the task list to be scaned for sensor input. If no games are
; active, then check task 0 to determine if the correct sensor sequence
; is occuring which will initiate the next game.

Ck_task_egg:

      LDA   Game_1            ;get game active bits
      ROR   A           ;move bit 0 to carry
      BCC   Ck_g2       ;check next if not activ
```

```
      JMP   Game_fortune      ;jump if active
Ck_g2:
      ROR   A           ;bit 1
      BCC   Ck_g3       ;check next if not activ
      JMP   Game_Rap    ;jump if active
Ck_g3:
      ROR   A           ;bit 2
      BCC   Ck_g4       ;check next if not activ
      JMP   Game_hideseek     ;jump if active
Ck_g4:
      ROR   A           ;bit 3
      BCC   Ck_g5       ;check next if not activ
      JMP   Game_simon  ;jump if active
Ck_g5:
      ROR   A           ;bit 4
      BCC   Ck_g6       ;check next if not activ
      JMP   Game_Burp   ;jump if active
Ck_g6:
      ROR   A           ;bit 5
      BCC   Ck_g7       ;check next if not activ
      JMP   Game_name   ;jump if active
Ck_g7:
      ROR   A           ;bit 6
      BCC   Ck_g8       ;check next if not activ
      JMP   Game_twinkle      ;jump if active
Ck_g8:
      ROR   A           ;bit 7
      BCC   Ck_g9       ;check next if not activ
      JMP   Game_rooster      ;jump if active

Ck_g9:


;  none active
;
;;************************


; Task 0 : scans all active requests from sensors looking for a trigger.
; If any are set then scan through the game select tables for each game
; looking for a match, and increment the counter each time a succesive
; match is found. If one is not in sequence, then that counter is reset
to
; zero. Since all counters are independent, then the first one to
completion
; wins and all others are zeroed.
;
; All sensor triggers are in one status byte so we can create a number
; based on who has been triggered (we ignore the I.R. sensor).
; The following bits are in Stat_4 and are set when they are triggered
; by the individual sensor routines :

; 00 = none
; 01 = Loud sound
; 02 = Light change brighter
; 04 = Light change darker
; 08 = Front tummy switch
; 10 = Back switch
; 20 = Feed switch
; 40 = Tilt switch
```

```
; 80 = Invert switch

; We assi n a single bit per game or egg senario. Each time a
; sensor is triggered, we increment the counter and test all eggs for
; a match. If a particular sensor doesnt match, then set its
disqualified
; bit and move on. If at any time all bits are set, then clear counter
to
; zero and start over. WHen a table gets an FF then that egg is
executed.
; Each time a sensor is triggered, the system timer is reset. This timer
; called 'Sensor_timer'is reset with 'Global_time' equate. This timer is
also
; used for the random sequential selection of sensor responses. If this
; timer goes to zero before an egg is complete, ie, Furby has not been
played
; with, then clear all disqualified bits and counters.

; Currently there are 24 possible eggs. (3 bytes)

;Qualify1:
;DQ_fortune EQU   01    ;bit 0 = fortune teller
;DQ_rap           EQU   02    ;bit 1 = rap song
;DQ_hide    EQU   04    ;bit 2 = hide and seek
;DQ_simon   EQU   08    ;bit 3 = simon says
;DQ_burp    EQU   10    ;bit 4 = burp attack
;DQ_name    EQU   20    ;bit 5 = say name
;DQ_twinkle EQU   40    ;bit 6 = sing song
;DQ_rooste- EQU   80    ;bit 7 = rooster-love you

;Qualify2:  ;;;;removed due to lack of RAM
;      bit 0 =
;      bit 1 =
;      bit 2 =
;      bit 3 =
;      bit 4 =
;      bit 5 =
;      bit 6 =
;      bit 7 =


; Test triggers here

Ck_game:
;      LDA   Sensor_timer      ;ck if no action for a while
       LDA   Bored_timer ;ck if no action for a while
       BNE   Ck_gamactv  ;jump if system active
       JSR   Clear_games ;go reset all other triggers and game pointers
Ck_gamactv:
       LDA   Qualify1    ;test if all are disqualified
       CMP   #FFh        ;compare activ bits only
       BNE   Ck_anysens  ;jump if some or all still active
;      LDA   Qualify2    ;test if all are disqualified
;      CMP   #00h        ;compare activ bits only
;      BNE   Ck_anysens  ;jump if some or all still active
       JSR   Clear_games ;go reset all other triggers and game pointers
Ck_anysens:
       LDA   Stat_4            ;ck if any sensor is triggered
       BNE   Ck_gam1           ;go ck games if any set
       JMP   Ck_bored    ;bypass if none
```

```
;
Ck_gam1:    ;fortune teller
      LDX   Egg_cnt           ;get current count
      LDA   Qualify1    ;update game qualification
      AND   #DQ_fortune ;check if dis-qualified bit
      BNE   Ck_gam2           ;bail out if is
      LDA   Fortune_table,X   ;get current data
      AND   Stat_4            ;compare against sensor trigger
      BNE   Ck_gam1a    ;if set then good compare
      LDA   Qualify1    ;update game qualification
      ORA   #DQ_fortune ;set dis-qualified bit
      STA   Qualify1    ;update system
      JMP   Ck_gam2           ;check next egg
Ck_gam1a:
      LDA   Fortune_table+1,X ;get current +1 to see if end of egg
      CMP   #FFh        ;test if end of table and start of game
      BNE   Ck_gam2           ;jump if not at end
      JSR   Clear_games ;go reset all other triggers and game pointers
      LDA   Game_1            ;get system
      ORA   #Fortune_mode     ;start game mode
      STA   Game_1            ;update
      JMP   Idle        ;done
;
Ck_gam2:    ; Rap mode
      LDA   Qualify1    ;update game qualification
      AND   #DQ_rap           ;check if dis-qualified bit
      BNE   Ck_gam3           ;bail out if is
      LDA   Rap_table,X ;get current data
      AND   Stat_4            ;compare against sensor trigger
      BNE   Ck_gam2a    ;if set then good compare
      LDA   Qualify1    ;update game qualification
      ORA   #DQ_rap           ;set dis-qualified bit
      STA   Qualify1    ;update system
      JMP   Ck_gam3           ;check next egg
Ck_gam2a:
      LDA   Rap_table+1,X     ;get current data +1 to see if end of egg
      CMP   #FFh        ;test if end of table and start of game
      BNE   Ck_gam3           ;jump if not at end
      JSR   Clear_games ;go reset all other triggers and game pointers
      LDA   Game_1            ;get system
      ORA   #Rap_mode   ;start game mode
      STA   Game_1            ;update
      JMP   Idle        ;done
;
Ck_gam3:    ; Hide and seek
      LDA   Qualify1    ;update game qualification
      AND   #DQ_hide    ;check if dis-qualified bit
      BNE   Ck_gam4           ;bail out if is
      LDA   Hseek_table,X     ;get current data
      AND   Stat_4            ;compare against sensor trigger
      BNE   Ck_gam3a    ;if set then good compare
      LDA   Qualify1    ;update game qualification
      ORA   #DQ_hide    ;set dis-qualified bit
      STA   Qualify1    ;update system
      JMP   Ck_gam4           ;check next egg
Ck_gam3a:
      LDA   Hseek_table+1,X   ;get current data +1 to see if end of egg
      CMP   #FFh        ;test if end of table and start of game
      BNE   Ck_gam4           ;jump if not at end
      JSR   Clear_games ;go reset all other triggers and game pointers
```

```
        LDA   Game_1              ;get system
        ORA   #Hideseek_mode      ;start game mode
        STA   Game_1              ;update
        JMP   Idle          ;done
;
Ck_gam4:    ; Simon says
        LDA   Qualify1      ;update game qualification
        AND   #DQ_simon     ;check if dis-qualified bit
        BNE   Ck_gam5             ;bail out if is
        LDA   Simon_table,X       ;get current data
        AND   Stat_4              ;compare against sensor trigger
        BNE   Ck_gam4a      ;if set then good compare
        LDA   Qualify1      ;update game qualification
        ORA   #DQ_simon     ;set dis-qualified bit
        STA   Qualify1      ;update system
        JMP   Ck_gam5             ;check next egg
Ck_gam4a:
        LDA   Simon_table+1,X     ;get current data +1 to see if end of egg
        CMP   #FFh          ;test if end of table and start of game
        BNE   Ck_gam5             ;jump if not at end
        JSR   Clear_games ;go reset all other triggers and game pointers
        LDA   Game_1              ;get system
        ORA   #Simonsay_mode      ;start game mode
        STA   Game_1              ;update
        LDA   #00           ;clear all pointers
        STA   Stat_5              ;system
        JMP   Idle          ;done

Ck_gam5:    ; Burp attack
        LDA   Qualify1      ;update game qualification
        AND   #DQ_burp      ;check if dis-qualified bit
        BNE   Ck_gam6             ;bail out if is
        LDA   Burp_table,X        ;get current data
        AND   Stat_4              ;compare against sensor trigger
        BNE   Ck_gam5a      ;if set then good compare
        LDA   Qualify1      ;update game qualification
        ORA   #DQ_burp      ;set dis-qualified bit
        STA   Qualify1      ;update system
        JMP   Ck_gam6             ;check next egg
Ck_gam5a:
        LDA   Burp_table+1,X      ;get current data +1 to see if end of egg
        CMP   #FFh          ;test if end of table and start of game
        BNE   Ck_gam6             ;jump if not at end
        JSR   Clear_games ;go reset all other triggers and game pointers
        LDA   Game_1              ;get system
        ORA   #Burp_mode    ;start game mode
        STA   Game_1              ;update
        LDA   #00           ;clear all pointers
        STA   Stat_5              ;system
        JMP   Idle          ;done

Ck_gam6:    ; say name
        LDA   Qualify1      ;update game qualification
        AND   #DQ_name      ;check if dis-qualified bit
        BNE   Ck_gam7             ;bail out if is
        LDA   Name_egg,X    ;get current data
        AND   Stat_4              ;compare against sensor trigger
        BNE   Ck_gam6a      ;if set then good compare
        LDA   Qualify1      ;update game qualification
        ORA   #DQ_name      ;set dis-qualified bit
```

```
        STA   Qualify1     ;update system
        JMP   Ck_gam7              ;check next egg
Ck_gam6a:
        LDA   Name_egg+1,X         ;get current data +1 to see if end of egg
        CMP   #FFh         ;test if end of table and start of game
        BNE   Ck_gam7              ;jump if not at end
        JSR   Clear_games ;go reset all other triggers and game pointers
        LDA   Game_1               ;get system
        ORA   #Name_mode   ;start game mode
        STA   Game_1               ;update
        LDA   #00          ;clear all pointers
        STA   Stat_5               ;system
        JMP   Idle         ;done

Ck_gam7:    ; twinkle song
        LDA   Qualify1     ;update game qualification
        AND   #DQ_twinkle ;check if dis-qualified bit
        BNE   Ck_gam8              ;bail out if is
        LDA   Twinkle_egg,X        ;get current data
        AND   Stat_4               ;compare against sensor trigger
        BNE   Ck_gam7a     ;if set then good compare
        LDA   Qualify1     ;update game qualification
        ORA   #DQ_twinkle ;set dis-qualified bit
        STA   Qualify1     ;update system
        JMP   Ck_gam8              ;check next egg
Ck_gam7a:
        LDA   Twinkle_egg+1,X      ;get current data +1 to see if end of egg
        CMP   #FFh         ;test if end of table and start of game
        BNE   Ck_gam8              ;jump if not at end
        JSR   Clear_games ;go reset all other triggers and game pointers
        LDA   Game_1               ;get system
        ORA   #Twinkle_mode        ;start game mode
        STA   Game_1               ;update
        LDA   #00          ;clear all pointers
        STA   Stat_5               ;system
        JMP   Idle         ;done

Ck_gam8:    ; rooster loves you
        LDA   Qualify1     ;update game qualification
        AND   #DQ_rooster ;check if dis-qualified bit
        BNE   Ck_gam9              ;bail out if is
        LDA   Rooster_egg,X        ;get current data
        AND   Stat_4               ;compare against sensor trigger
        BNE   Ck_gam8a     ;if set then good compare
        LDA   Qualify1     ;update game qualification
        ORA   #DQ_rooster ;set dis-qualified bit
        STA   Qualify1     ;update system
        JMP   Ck_gam9              ;check next egg
Ck_gam8a:
        LDA   Rooster_egg+1,X      ;get current data +1 to see if end of egg
        CMP   #FFh         ;test if end of table and start of game
        BNE   Ck_gam9              ;jump if not at end
        JSR   Clear_games ;go reset all other triggers and game pointers
        LDA   Game_1               ;get system
        ORA   #Rooster_mode        ;start game mode
        STA   Game_1               ;update
        LDA   #00          ;clear all pointers
        STA   Stat_5               ;system
        JMP   Idle         ;done
```

```
Ck_gam9:


Ck_gamend:
      INC   Egg_cnt           ;incs on any sensor trigger
      LDA   Egg_cnt           ;get
      CLC
      SBC   #10          ;limit max to 10 for error checking
      BCC   Cge2         ;continue if less
      JSR   Clear_games  ;reset all
Cge2:
      LDA   #00          ;clear all sensor triggers this pass
      STA   Stat_4            ;ready for next pass of sensor triggers
      JMP   Ck_bored     ;done with easter egg test

;*************************

Clear_all_gam:
      LDA   #00          ;clear all game enabled bits
      STA   Game_1            ;
;     STA   Game_2            ;

Clear_games:
      LDA   #00          ;clear counter
      STA   Egg_cnt           ;
      STA   Stat_4            ;clear game status
      STA   Stat_5            ;clear game status
      STA   Qualify1     ;clear all dis-qualify bits
;     STA   Qualify2     ;clear all dis-qualify bits
      RTS                ;done

;*************************

; 00 = none
; 01 = Loud sound
; 02 = Light change brighter
; 04 = Light change darker
; 08 = Front tummy switch
; 10 = Back switch
; 20 = Feed switch
; 40 = Tilt switch
; 80 = Invert switch

; These look up tables provide the sequence of sensor triggers required
; to enter that specific game mode. (FFh is always the last byte)


Fortune_table:
      DB    04h,04h,10h,FFh          ;lght,lght,back

Rap_table:
      DB    01h,01h,01h,01h,FFh      ;snd,snd,snd,snd

Hseek_table:
      DB    04h,04h,04h,08h,FFh      ;light,light,light,frnt

Simon_table:
      DB    08h,10h,01h,04h,FFh      ;frnt,back,snd,lght

Burp_table:
```

```
      DB    20h,20h,20h,10h,FFh      ;feed,feed,feed,back

Name_egg:
      DB    08h,08h,08h,10h,FFh      ;frnt,frnt,frnt,back

Twinkle_egg:
      DB    01h,01h,01h,10h,FFh      ;snd,snd,snd,back

Rooster_egg:
      DB    04h,04h,04h,10h,FFh      ;light,light,light,back


;*****************************
;
;
; Normal task scan of sensors and timers.
;
Ck_bored:
      LDA   Bored_timer ;ck if bored ... =0
      BNE   Ck_tsk1           ;jump if not bored

;  Currently uses 4 tables, one for each age.

      LDA   #Bored_split      ;get random/sequential split
      STA   IN_DAT            ;save for random routine

      LDX   #Seq_bored  ;get number of sequential selections
      LDA   #Ran_bored  ;get number of randoms
      JSR   Ran_seq           ;go decide random/sequential
      BCS   Bored_ran   ;Random mode when carry SET

      LDX   Bored_count ;save current
      INC   Bored_count ;if not then next table entry
      LDA   Bored_count ;get
      CLC
      SBC   #Seq_bored-1      ;ck if > assignment
      BCC   Bored_side  ;jump if <
      LDA   #00         ;reset to 1st entry of sequential
      STA   Bored_count ;
Bored_side:
      TXA               ;current count

Bored_ran:
      JSR   Decid_age   ;do age calculation for table entry
      LDX   TEMP0       ;age offset
      LDA   Bored_S1,X  ;get new sound/word
      STA   Macro_Lo    ;save lo byte of Macro table entry
      INX               ;
      LDA   Bored_S1,X  ;get new sound/word
      STA   Macro_Hi    ;save hi byte of Macro table entry
      JMP   Start_macro ;go set group/table pointer for motor & spch
;
Ck_tsk1:
      LDA   Task_ptr    ;
      CMP   #01         ;decide which
      BNE   Ck_tsk4           ;jump if not
      JMP   CK_tilt           ;Ck ball switch side sense
Ck_tsk4:
      CMP   #02         ;decide which
      BNE   Ck_tsk5           ;jump if not
```

```
      JMP   Ck_invert   ;Ck ball switch inverted sense
Ck_tsk5:
      CMP   #03         ;decide which
      BNE   Ck_tsk6           ;jump if not
      JMP   Ck_back           ;Ck Touch switch back sensor
Ck_tsk6:
      CMP   #04         ;decide which
      BNE   Ck_tsk7           ;jump if not
      JMP   Ck_IR       ;Ck IR input
Ck_tsk7:
      CMP   #05         ;decide which
      BNE   Ck_tsk8           ;jump if not
      JMP   Ck_feed     ;Ck Feed sensor
Ck_tsk8:
      CMP   #06         ;decide which
      BNE   Ck_tsk9           ;jump if not
      JMP   Ck_light    ;Ck Light sensor
Ck_tsk9:
      CMP   #07         ;decide which
      BNE   Ck_tsk10    ;jump if not
      JMP   Ck_front    ;Ck Front touch switch
Ck_tsk10:
      CMP   #08         ;decide which
      BNE   Ck_tskend   ;jump if not
      JMP   Ck_sound    ;Ck Mic input

Ck_tskend:
      JMP   Idle        ;no task

;*************************************************************
;*************************************************************
;*************************************************************

; This rtn tests for motor and speech activity and only services them
; to allow each request to finish, and then returns to task routine.
; As long as motor is active, we continually reload the motor led timer
; to keep the optical counter alive and when all activity is complete,
; the IRQ will turn led off when timer goes to 00.

Notrdy:
      JSR   Task_1              ;go do speech
      JSR   Task_2              ;go do motor

      LDA   Stat_1              ;get system
      AND   #Word_activ ;Test for spch word active
      BNE   Notrdy2             ;jump if not done
      LDA   Stat_1              ;update
      AND   #Say_activ  ;ck for saysent active
      BNE   Notrdy2

      LDA   Stat_2              ;get system
      AND   #Motor_seek ;ck motor request
      BNE   Notrdy2             ;jump if set
      LDA   Stat_2              ;get system
      AND   #Motor_actv ;ck motor in motion
      BNE   Notrdy2

      LDA   Drift_fwd   ;motor drift counter 0 when done
      BNE   Notrdy2
```

```
      LDA   Drift_rev   ;
      BNE   Notrdy2           ;

      LDA   Stat_2            ;system
      AND   #Macro_actv ;ck for flag request
      BEQ   Notrdy_dn   ;bail  if none
      JSR   Ck_Macro    ;decide if more chaining in process
      JMP   Notrdy2           ;continue
Notrdy_dn:
      RTS               ;only leave when everyone done

Notrdy2:
      LDA   #Motor_led_rst    ;get led timer reload
      STA   Motor_led_timer   ;how long the motor stays on
      JMP   Notrdy            ;loop

;*********************************************************
;*********************************************************
;*********************************************************
;*********************************************************


Task_1:
      LDA   Stat_1            ;get system
      AND   #Word_activ ;Test for spch word active
      BNE   W_activ           ;jump if not done
;More_spch:
      LDA   Stat_1            ;update
      AND   #Say_activ  ;ck for saysent active
      BEQ   EndTask_1   ;nothing going on, ck next task
      JSR   Do_nextsent ;continue on with saysent
      JMP   EndTask_1   ;Next task
W_activ:
      LDA   Port_B            ;get TI req/busy line
      AND   #TI_RTS           ;get bit
      BNE   EndTask_1   ;if no speech then ck motor
      JSR   Do_spch           ;go send next byte to TI
EndTask_1:
      RTS
;
;
Task_2:

;***********  Motor Routines ************
;
; get next motor data

Ck_motor:
      LDA   Stat_2            ;get system
      AND   #Motor_actv ;ck motor in motion
      BEQ   Ck_mot2           ;done
      JMP   Do_motor    ;not done so check position
Ck_mot2:
      LDA   Stat_2            ;get system
      AND   #Motor_seek ;ck motor request
      BEQ   NMM_out           ;jump if none

Next_motor:
;     LDA   Drift_fwd   ;motor drift counter 0 when done
```

```
;       BNE     NMM_out             ;wait til 0
;       LDA     Drift_rev   ;
;       BNE     NMM_out             ;wait til 0


; Set a timer and ck counter 'motorstoped' (incremented with wheel
count)
; to see if it changed. When it stops changing then the motor has
stopped.

        LDA     motorstoped ;ck for 0
        BNE     NMM_out             ;wait till 0
        LDA     TEMP1       ;get last motor count
        CMP     Pot_timeL   ;ck if changed
        BEQ     Motor_done  ;jump if same (motor finally stopped)
        LDA     Pot_timeL   ;get current
        STA     TEMP1       ;
        LDA     #15         ;reset timer (8)
        STA     motorstoped ;
        JMP     NMM_out             ;wait another cycle

Motor_done:
        LDA     Cycle_timer ;get step timer
        BNE     NMM_out             ;wait til 0

        STA     Drift_counter       ;use as a temp register

        JSR     Motor_data  ;get data

        LDA     #00
        STA     TEMP1       ;reset

        LDA     Motor_lo    ;get data (use for 1byte table (DB)).
        CMP     #FFh        ;is it table end (dont inc off end)
        BNE     Motor_pause ;more
        LDA     Stat_2              ;get system
        AND     #Motor_ntseek       ;clear seek flag
        STA     Stat_2              ;update system
NMM_out:
        JMP     Endtask_2   ;seek complete

Motor_pause:
        LDA     Motor_lo    ;check for pause request on this step (00)
        BNE     More_motor  ;more
        JMP     Motor_killend       ;set cycle timer and wait for next motor
step
;
;
; To initialize the motor call table, the originator loads 'Which_motor'
; with the pointer and calls 'Decide_motor'.

Ck_Macro:
        JSR     Next_macro  ;get data
        STA     Which_motor ;save motor seek pointer
        JSR     Next_macro  ;get data
        STA     Mgroup              ;save high byte
        CMP     #00h        ;check for end of macro
        BNE     Got_macro   ;do it if not 0
        LDA     Which_motor ;ck lo byte for 0
        CMP     #00h        ;check for end of macro
```

```
      BNE   Got_macro   ;do it if not 0 else must be end command
End_macro:
      LDA   Stat_2              ;get system
      AND   #Nt_macro_actv      ;clear request
      STA   Stat_2              ;update
;     LDA   #Bored_reld ;reset bored timer
;     STA   Bored_timer ;
No_macro:
      RTS               ;done
;
Next_macro:
        LDX     #00H
      LDA   (Macro_Lo,X)        ;get speech/motor table request
      INC   Macro_Lo    ;next
      BNE   Mac_dat2    ;jmp in no roll over
      INC   Macro_Hi    ;rolled over so hi +1
Mac_dat2:
      RTS               ;
;
Got_macro:

; The speech and motor pointer table pointer from the sensor table ,
are
; a 1-999 decimal number. The assemble converts to two 8 bit numbers and
; this creates a one of four group of 128 byte pointers in each group.
; We also do 2's offset for table lookup.

      CLC               ;do motor
      ROL   Which_motor ;move hi bit to carry
      ROL   Mgroup              ;move carry into one of four group ptr

      LDA   Which_motor ;offset
      STA   Which_word  ;set speech group pointers
      LDA   Mgroup              ;offset
      STA   Sgroup              ;
      JSR   Decide_motor        ;start motor routine
      JSR   Say_0       ;start speech routine
      RTS               ;back to task master

;
;*********************************  ************************

More_motor:
      LDA   Stat_3              ;system
      ORA   #Motor_on   ;flag  on mode
      STA   Stat_3              ;update
;m    LDA   Mon_len             ;get length of on pulse
;m    STA   Motor_pulse ;set timer

      LDA   Stat_2              ;get system
      ORA   #Motor_actv ;set motor in motion
      STA   Stat_2              ;update

Mcalc_lo:

; When motor stops, if the IR detector is on the slot in the wheel, no
; action is reeded. If passed the slot, when the next motion command
occurs,
; if the direction is the same as the last motion, no action is needed.
; If the direction is opposit to last motion then we decrement or
```

```
; increment, based on new direction, to compensate for the slot which
; will be counted twice.

        LDA     Motor_lo     ;get data
        CMP     Pot_timeL    ;ck for same
        BNE     Tst_fwdmore ;jump if not 0
        LDA     Stat_2              ;get system
        AND     #Motor_inactv       ;clear activ flag
        STA     Stat_2              ;update system
        JMP     Endtask_2    ;bail out
Tst_fwdmore:
        CLC
        SBC     Pot_timeL    ;get current position
        BCC     Go_rev              ;if borrow then dec command

Go_fwd:
        LDA     Port_C              ;get IR detector
        AND     #Pos_sen     ;
        BEQ     Go_fwd2             ;bypass if sensor is over slot in wheel
        LDA     Stat_2              ;get system
        AND     #Motor_fwd   ;get direction motor was last headed
        BNE     Go_fwd2             ;if set then new direction is same as last
        DEC     Pot_timeL2   ;compensate for counter direction reversal

Go_fwd2:
        LDA     Stat_2              ;get system
        ORA     #Motor_fwd   ;set = motor fwd (inc)
        ORA     #Motor_actv ;set motor in motion
        STA     Stat_2              ;update system
        LDA     Stat_3              ;get current status
        ORA     #Motor_off   ;turn both motors off
        AND     #Motor_fwds ;move motor in fwd dir
        JMP     End_rev             ;go finish port setup
;
Go_rev:
        LDA     Port_C              ;get IR detector
        AND     #Pos_sen     ;
        BEQ     Go_rev2             ;bypass if sensor is over slot in wheel
        LDA     Stat_2              ;get system
        AND     #Motor_fwd   ;get direction motor was last headed
        BEQ     Go_rev2             ;if clr then new direction is same as last
        INC     Pot_timeL2   ;compensate for counter direction reverzal

Go_rev2:
        LDA     Stat_2              ;get system
        AND     #Motor_rev   ;clear fwd flag
        ORA     #Motor_actv ;set motor in motion
        STA     Stat_2              ;update system
        LDA     Stat_3              ;get current status
        ORA     #Motor_off   ;turn both motors off
        AND     #Motor_revs ;move motor in rev dir
End_rev:
        STA     Stat_3
        JMP     Endtask_2    ;done

Do_motor:

;((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((
; motor speed control
```

```
;  jmp Byp_motorS3

       LDA    Stat_0              ;system
       AND    #Init_Mspeed        ;ck if motor to do speed test
       BEQ    Byp_motorS3 ;only runs on wake up
       LDA    Stat_0              ;system
       AND    #Init_motor ;ck if motor to do speed test
       BEQ    Byp_motorS2 ;only runs on wake up
       LDA    Stat_0              ;system
       AND    #Nt_Init_motor      ;done
       STA    Stat_0              ;update

       LDA    #00          ;reset opto speed counter
       STA    Mot_opto_cnt        ;setit
       LDA    #Opto_spd_reld      ;get timer value for speed test
       STA    Mot_speed_cnt       ;set it

Byp_motorS2:

       LDA    Mot_speed_cnt       ;get timer
       BNE    Byp_motorS3 ;do nothing if >0

       LDX    Mot_opto_cnt        ;get wheel count during speed test
       LDA    Motor_speed,X       ;get motor on pulse width
       STA    Mon_len             ;on time
       LDA    #Mpulse_on+1        ;max cycle time on+off
       CLC
       SBC    Mon_len             ;get cmplmnt
       STA    Moff_len    ;
       BCS    Byp_motorS3 ;jump if not neg
       LDA    #00
       STA    Moff_len    ;

       LDA    Stat_0              ;system
       AND    #NT_Init_Mspeed     ;clear motor to do speed test
       STA    Stat_0              ;update

Byp_motorS3:

;))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))

; On power up we preset Mon_len to 11 and Moff_len to 5. This prevents
; the motor from destroying itself when the batteries are 6.4v.
; This also gives a timed count on the speed test of -7 difference.
; so I adjusted the table to compensate for the shift.


; Compare motor position to see if at destination yet

       LDA    Stat_2              ;get direction
       AND    #Motor_fwd  ;set=inc  clr=dec
       BEQ    Motor_dec   ;

;bit was set so motor in inc condition

FCalc_lo:
       LDA    Motor_lo    ;get data
       CLC                ;carry=0
```

```
        SBC   Pot_timeL    ;table - current cap time
        BCC   Motor_killfwd      ;jump if result is negative
        JMP   Endmotor     ;wait till there & pulse for speed

; Reverse direction......
Motor_dec:                 ; go reverse
        LDA   Pot_timeL    ;destination
        CLC                ;carry=0
        SBC   Motor_lo     ;table position to seek to
        BCC   Motor_killrev      ;jump if result negative
        JMP   Endmotor     ;wait till there & pulse for speed

Motor_killfwd:
        LDA   Drift_counter      ;ck how far we travled
        TAX                ;prep for drift table
        CLC                ;
        SBC   #20          ;ck if less than 20 steps
        BCC   M_killf2     ;jump if less
        LDA   #Drift_long  ;long delay if >10 steps
        JMP   M_killf3     ;go fini
M_killf2:
        LDA   Drift_table,X      ;get brake pulse
;       LDA   #Drift_short       ;short delay if < 10 steps
M_killf3:
        STA   Drift_rev    ;save
        JMP   Motor_killend      ;go shut down motor
;
Motor_killrev:
        LDA   Drift_counter      ;ck how far we travled
        TAX                ;prep for drift table
        CLC                ;
        SBC   #20          ;ck if less than 20 steps
        BCC   M_killr2     ;jump if less
        LDA   #Drift_long  ;long delay if >10 steps
        JMP   M_killr3     ;go fini
M_killr2:
        LDA   Drift_table,X      ;get brake pulse
;       LDA   #Drift_short       ;short delay if < 10 steps
M_killr3:
        STA   Drift_fwd    ;save
Motor_killend:
        LDA   Stat_3             ;get current status
        ORA   #Motor_off   ;turn both motors off
        STA   Stat_3             ;update
        LDA   Stat_2             ;get system
        AND   #Motor_inactv      ;clear activ flag
        STA   Stat_2             ;update system
        LDA   Which_delay  ;time til next read
        STA   Cycle_timer  ;reset it
        LDA   #00
        STA   TEMP1        ;used to test motor drift between seeks
        JMP   Endtask_2    ;

; Drift table controls the magnitude of braking pulse applied.
; If the distance just travled is less than 20 then use that number
; to point into table and get new brake pulse length.

Drift_table:

;       DB    24,30,32,34,35,38,40,44,48,54,56
```

```
;       DB      58,60,60,60,60,60,60,60,60,60,60

;       DB      20,22,24,27,30,32,34,36,38
;       DB      46,48,50,52,54,56,58,60,60,60,60,60

        DB      25,26,27,28,30,32,34,36,38,42,45
        DB      48,51,54,57,60,60,60,60,60,60,60

;
; On wake up when the motor moves from position 10 to 134, we
; time it and increment a counter which is used to access this table
; and get the motor on pulse value.


; Refer to power up preset pulse width for table pointers.

Motor_speed:

    DB   Mpulse_on,Mpulse_on,Mpulse_on
    DB Mpulse_on,Mpulse_on,Mpulse_on
    DB Mpulse_on,Mpulse_on,Mpulse_on
    DB   Mpulse_on,Mpulse_on,Mpulse_on
    DB   Mpulse_on,Mpulse_on,Mpulse_on
    DB   Mpulse_on,Mpulse_on,Mpulse_on      ;f,10
    DB Mpulse_on,Mpulse_on,Mpulse_on
    DB   Mpulse_on,Mpulse_on,Mpulse_on
    DB   Mpulse_on,Mpulse_on,Mpulse_on-1
    DB Mpulse_on-2,Mpulse_on-3,Mpulse_on-4     ;1b,1c
    DB Mpulse_on-5,Mpulse_on-5,Mpulse_on-6
    DB Mpulse_on-7,Mpulse_on-8,Mpulse_on-9
    DB Mpulse_on-9,Mpulse_on-9,Mpulse_on-9
    DB Mpulse_on-9,Mpulse_on-9,Mpulse_on-9
    DB Mpulse_on-9,Mpulse_on-9,Mpulse_on-9
    DB Mpulse_on-9,Mpulse_on-9,Mpulse_on-9
    DB Mpulse_on-9,Mpulse_on-9,Mpulse_on-9
    DB Mpulse_on-9,Mpulse_on-9,Mpulse_on-9



;
;
; This finds the 16 bit adrs of the table and points the motor

Decide_motor:
        LDX    Which_motor ;offset ptr
        LDA    Mgroup              ;get current group pointer
        CMP    #03          ;is it table group 4
        BEQ    Dec_mot4     ;jump if is
        CMP    #02          ;is it table group 3
        BEQ    Dec_mot3     ;jump if is
        CMP    #01          ;is it table group 2
        BEQ    Dec_mot2     ;jump if is
Dec_mot1:           ;table group 1
        LDA    Motor_grp1,X       ;get lo pointer
        STA    Motptr_lo    ;working buffer
        INX                 ;X+1
        LDA    Motor_grp1,X       ;get hi pointer
        JMP    Dec_mot_end ;go finish load
Dec_mot2:           ;
```

```
        LDA   Motor_grp2,X      ;get lo pointer
        STA   Motptr_lo   ;working buffer
        INX               ;X+1
        LDA   Motor_grp2,X      ;get hi pointer
        JMP   Dec_mot_end ;go finish load
Dec_mot3:
        LDA   Motor_grp3,X      ;get lo pointer
        STA   Motptr_lo   ;working buffer
        INX               ;X+1
        LDA   Motor_grp3,X      ;get hi pointer
        JMP   Dec_mot_end ;go finish load
Dec_mot4:
        LDA   Motor_grp4,X      ;get lo pointer
        STA   Motptr_lo   ;working buffer
        INX               ;X+1
        LDA   Motor_grp4,X      ;get hi pointer
Dec_mot_end:
        STA   Motptr_hi   ;working buffer
        LDA   Stat_2              ;system
        ORA   #Motor_seek ;flag system
        STA   Stat_2              ;update
;       LDA   #Motor_led_rst    ;get moto led timer reload
;       STA   Motor_led_timer   ;how long the motor IR led stays on
More_multi_m:
        JSR   Motor_data  ;1st time only get 1st byte (delay)
        LDA   Motor_lo    ;get data
        STA   Which_delay ;motor delay control
        RTS               ;done
;
;
; Get next motor data from table according to indirect pointer.

; NOTE: we are now using DB statments in the motor table
;       so were back to single byte format.

Motor_data:
        LDX     #00H
        LDA     (Motptr_lo,X) ;Get the motor data
        STA   Motor_lo    ;lo byte
        INC   Motptr_lo   ;next
        BNE   Mot_dat2    ;jmp in no roll over
        INC   Motptr_hi   ;rolled over so hi +1
Mot_dat2:
        RTS


; Test motor pulse timer and alternate on & off to keep motor speed
; constant through battery deterioration.

Endmotor:
;m      LDA   Motor_pulse ;ck pulse timer
;m      BNE   Endtask_2   ;jump if not done
;m      LDA   Stat_3              ;system
;m      AND   #Motor_on   ;is it an power on pulse
;m      BNE   Emotor_off  ;jump if on pulse (set)
;m      LDA   Stat_3              ;system
;m      ORA   #Motor_on   ;flag  on mode
;m      STA   Stat_3              ;update
;m      LDA   Mon_len             ;get length of on pulse
;m      STA   Motor_pulse ;set timer
```

```
;mPls_fwd:
;m    LDA   Stat_2            ;get system
;m    AND   #Motor_fwd  ;ck if set = motor fwd (inc)
;m    BEQ   Pls_rev           ;else go reverse
;m    LDA   Stat_3            ;get current status
;m    ORA   #Motor_off  ;turn both motors off
;m    AND   #Motor_fwds ;move motor in fwd dir
;m    JMP   Plsend            ;go finish port setup
;mPls_rev:
;m    LDA   Stat_3            ;get current status
;m    ORA   #Motor_off  ;turn both motors off
;m    AND   #Motor_revs ;move motor in rev dir
;mPlsend:
;m    STA   Stat_3
;m    JMP   Endtask_2   ;done
;mEmotor_off:     ;must be on so turn off
;m    LDA   Stat_3            ;system
;m    AND   #Ntmot_on   ;set to power off pulse
;m    STA   Stat_3            ;update
;m    LDA   Moff_len    ;get length of off pulse
;m    STA   Motor_pulse ;set timer
;m    LDA   Stat_3            ;get current status
;m    ORA   #Motor_off  ;turn both motors off
;m    STA   Stat_3            ;update
Endtask_2:
      RTS               ;back to Idle rtn

;*****************************************************************
;*****************************************************************
; Start motor/speech from macro table

; Because of conflicts in diagnostic routines, this routine has been
; changed to a subroutine. All normal sensors jump here, diags call
; direct.

Start_macro:
      LDA   #Bored_reld ;reset bored timer
      STA   Bored_timer ;

      LDA   Macro_Lo    ;save for sleepy & IR tests
      STA   Req_macro_lo      ;
      LDA   Macro_Hi    ;save for sleepy & IR tests
      STA   Req_macro_hi      ;

      JSR   Get_macro   ;
      JMP   Idle        ;done

Get_macro:

; Motor noise is triggering sound sensor hardware, so this sets the
; previously sound done flag, and the system will not respond to the
; sound sensor until the sound trigger line goes low and clears prev
done.

      LDA   Stat_3            ;system
      ORA   #Sound_stat ;
      STA   Stat_3            ;set prev dn

;------------------- end sound flag
```

```
      INC   Age_counter ;rolls over to inc age
      BNE   Same_age    ;jump if no roll over

;_____________________________________________________

; AGE INCRMNT uses bit 7 to double age counter
      LDA   Age         ;get bit 7 - set = counter rolled over twice
      AND   #80h              ;get bit 7
      BNE   Roll_age    ;bit 7 set so inc age
      LDA   Age
      ORA   #80h        ;set bit 7 for next counter roll over
      STA   Age         ;update
      JMP   Same_age    ;done

Roll_age:
      INC   Age         ;just grew up some
      LDA   Age
      AND   #07h        ;clear bit 7
      STA   Age
      CLC
      SBC   #03         ;make sure it isnt > 3  (0-3 age)
      BCC   Same_age    ;jump if <4
      LDA   #03         ;max age
      STA   Age         ;

Same_age:
;------------------- end age


      LDA   Stat_2            ;system
      ORA   #Macro_actv ;flag request
      STA   Stat_2            ;update
      CLC               ;do speech
      ROL   Macro_Lo    ;move hi bit to carry & get 2's offset
      ROL   Macro_Hi    ;move carry into one of four group ptr

      LDX   Macro_Lo    ;offset ptr
      LDA   Macro_Hi    ;get current group pointer
      CMP   #03         ;is it table group 4
      BEQ   Dec_macro4  ;jump if is
      CMP   #02         ;is it table group 3
      BEQ   Dec_macro3  ;jump if is
      CMP   #01         ;is it table group 2
      BEQ   Dec_macro2  ;jump if is
Dec_macro1:       ;table group 1
      LDA   Macro_grp1,X      ;get lo pointer
      STA   Macro_Lo    ;working buffer
      INX               ;X+1
      LDA   Macro_grp1,X      ;get hi pointer
      JMP   Dec_macro_end     ;go finish load
Dec_macro2:       ;
      LDA   Macro_grp2,X      ;get lo pointer
      STA   Macro_Lo    ;working buffer
      INX               ;X+1
      LDA   Macro_grp2,X      ;get hi pointer
      JMP   Dec_macro_end     ;go finish load
Dec_macro3:       ;
      LDA   Macro_grp3,X      ;get lo pointer
      STA   Macro_Lo    ;working buffer
      INX               ;X+1
```

```
      LDA   Macro_grp3,X      ;get hi pointer
      JMP   Dec_macro_end     ;go finish load
Dec_macro4:       ;
      LDA   Macro_grp4,X      ;get lo pointer
      STA   Macro_Lo    ;working buffer
      INX               ;X+1
      LDA   Macro_grp4,X      ;get hi pointer
Dec_macro_end:
      STA   Macro_Hi    ;working buffer
      RTS               ;

;
;
;*****************************************************************
;*****************************************************************
;*****************************************************************

;
; This group of speech & misc routines are used for the various game
; play modes, triggered by the easter egg.


;*****************************************************************
;*****************************************************************
;*****************************************************************

; REMEMBER TO CLEAR GAME ACTIVE STATUS WHEN DONE

; NOTE: Otomah should have a delay before the word to seperate this game
;       from the speech generated by the last sensor that triggered
;       this game.

Otomah_lo   EQU   #54h  ;using macro 84 for 1st word
Otomah_hi   EQU   #00   ;hi byte adrs 84 = 054h

Fortdelay_lo      EQU   #66h  ;using macro 102 for delay between speech
Fortdelay_hi      EQU   #00h  ;hi byte adrs 102 = 066h

Game_fortune:
      LDA   Stat_5            ;flag used at start of game
      AND   #temp_gam1  ;see if prev done
      BNE   Gam_fort2   ;jump if done
      LDA   Stat_5            ;flag used at start of game
      ORA   #temp_gam1  ;set prev done
      STA   Stat_5            ;update

      LDA   #Otomah_lo  ;get macro lo byte
      STA   Macro_Lo    ;save lo byte of Macro table entry
      LDA   #Otomah_hi  ;get macro hi byte
      STA   Macro_Hi    ;save hi byte of Macro table entry
      JSR   Get_macro   ;go start motor/speech
      JSR   Notrdy            ;Do / get  status for speech and motor

      LDA   #GameT_reload     ;reset game timer
      STA   Sensor_timer      ;

Gam_fort2:
      JSR   Test_all_sens     ;go check all sensors
```

```
        LDA   Stat_4              ;get sensor status
        AND   #Do_back     ;ck if back sw req
        BNE   Gam_fort4     jump if requested

        LDA   Stat_4              ;get sensor status
        AND   #Do_invert   ;ck if tilt sw req
        BEQ   Gam_fort3    ;jump if not requested
Gam_fort2a:
        JSR   Clear_all_gam       ;go clear all status, cancle game
        JMP   End_all_games       ;done go say "me done"

Gam_fort3:
        LDA   Sensor_timer        ;ck for no action timeout
        BEQ   Gam_fort2a  ;clear all if timed out
        JMP   Idle        ;wait for switch

Gam_fort4:
        LDA   Stat_4              ;get sensor status
        AND   #Nt_do_back ;back sw req
        STA   Stat_4              ;clear req

        LDA   #GameT_reload       ;reset game timer
        STA   Sensor_timer        ;

        LDA   #Fortdelay_lo       ;get macro lo byte
        STA   Macro_Lo     ;save lo byte of Macro table entry
        LDA   #Fortdelay_hi       ;get macro hi byte
        STA   Macro_Hi     ;save hi byte of Macro table entry
        JSR   Get_macro    ;go start motor/speech
        JSR   Notrdy              ;Do / get  status for speech and motor

        LDA   Stat_1              ;get system
        ORA   #Half_age    ;force table 1 or 2 in "Decid_age"
        STA   Stat_1              ;update

        LDA   #80h        ;get random/sequential split
        STA   IN_LAT              ;save for random routine

        LDX   #00         ;make sure only gives random
        LDA   #10h        ;get number of random selections
        JSR     n_seq             ;go decide random/sequential

;;;;;;;;;;;;;;;;;;;     Acc holds random number 0-F

        JSR   Decid_age    ;do age calculation for table entry
        LDX   TEMP0        ;age offset
        LDA   Fortyes_S1,X        ;get lo byte
        STA   Macro_Lo     ;save lo byte of Macro table entry
        STA   Req_macro_lo        ;save for game
        INX               ;
        LDA   Fortyes_S1,X        ;get hi byte
        STA   Macro_Hi     ;save hi byte of Macro table entry
        STA   Req_macro_hi        ;save for game

        LDX   #00         ;offset
Fort_Name2:
        LDA   Ck_Fort_name,X      ;ck lo byte
        CMP   #FFh        ;ck for end of table (note 255 cant execute)
        BEQ   Fort_Name_dn        ;done if is
        CMP   Macro_Lo     ;ck against last speech request
```

```
      BNE   Not_Fort2    ;jump if not
      INX                ;to hi byte
      LDA   Ck_Fort_name,X    ;ck hi byte
      CMP   Macro_Hi     ;ck against last speech request
      BNE   Not_Fort3    ;jump if not
      JMP   Say_Fortname      ;speak it
Not_Fort2:
      INX                ;
Not_Fort3:
      INX                ;
      JMP   Fort_Name2   ;loop til done

Say_Fortname:
      LDA   Name         ;current setting for table offs
      CLC
      ROL   A            ;2's comp
      TAX
      LDA   Name_table,X      ;get lo byte
      STA   Macro_Lo     ;save lo byte of Macro table entry
      INX                ;
      LDA   Name_table,X      ;get hi byte
      STA   Macro_Hi     ;save hi byte of Macro table entry
      JSR   Get_macro    ;go start new or speech
      JSR   Notrdy            ;Do / get  status for speech and motor

      LDA   Req_macro_lo      ;recover for game
      STA   Macro_Lo     ;set game speech
      LDA   Req_macro_hi      ;recover for game
      STA   Macro_Hi     ;set game speech

Fort_Name_dn:
      JMP   Start_macro ;go set group/table pointer for motor & spch

; compare macro to see if we are going to call Furby's name first.

Ck_Fort_name:
      DW    69
      DW    77

      DB    FFh,FFh     ;FF FF   is table terminator


;*****************************************************************
;
Game_Rap:
      JMP   Do_rap            ;1st time thru
Grap_2:
      JSR   Simon_timer ;decrement bored timer
      LDA   Bored_timer ;system elapsed time
      BEQ   Rap_over    ;jump if 0
      JSR   Test_all_sens     ;go check all sensors
      LDA   Stat_4            ;get sensors
      BEQ   Grap_2            ;loop if none
      AND   #Do_snd           ;ck for mic
      BNE   Do_rap            ;any other sensor stops game
Rap_over:
      JSR   Clear_all_gam     ;go clear all status, cancle games
      JMP   End_all_games     ;done go say "me done"
```

```
Do_rap:
      LDA   #00          ;clear all sensor flags
      STA   Stat_4             ;
      LDA   #GameT_reload      ;get reload
      STA   Bored_timer ;reset
      LDA   #80h         ;get random/sequential split
      STA   IN_DAT             ;save for random routine
      LDX   #00h         ;make sure only gives random
      LDA   #10h         ;get number of random selections
      JSR   Ran_seq            ;go get random selection
      LDA   TEMP1        ;get decision
      AND   #03h         ;got 1 of 4 decision
      CLC
      ROL   A            ;2's offsett
      TAX
      LDA   Rapsong,X    ;get macro lo byte
      STA   Macro_Lo     ;save lo byte of Macro table entry
      INX
      LDA   Rapsong,X    ;get macro hi byte
      STA   Macro_Hi     ;save hi byte of Macro table entry
      JSR   Get_macro    ;go start motor/speech
      JSR   Notrdy             ;Do / get  status for speech and motor
      JMP   Grap_2             ;loop

Rapsong:
      DW    395          ;macro RAP song pointer
      DW    396          ;
      DW    407          ;
      DW    416          ;

;*****************************************************************
;

HidePeek_lo EQU   #DBh  ;using macro 475 for startp "hide me" spch
HidePeek_hi EQU   #01h  ;hi byte adrs 475 = 1DBh

Hidsklost_lo      EQU   #D8h  ;using macro 472 for "nana nana nana
Hidsklost_hi      EQU   #01h  ;hi byte adrs 472 = 1D8h

Hidskwon_lo EQU   #B7h  ;using macro 439 for "whopee
Hidskwon_hi EQU   #01h  ;hi byte adrs 439 = 1B7h


Game_hideseek:

      LDA   #80          ;set timer for 1 min (80 * .742)
      STA   HCEL_LO            ;use temp ram for timer

      LDA   Name         ;current setting for table offset
      CLC
      ROL   A            ;2's comp
      TAX
      LDA   Name_table,X       ;get lo byte
      STA   Macro_Lo     ;save lo byte of Macro table entry
      INX                ;
      LDA   Name_table,X       ;get hi byte
      STA   Macro_Hi     ;save hi byte of Macro table entry
      JSR   Get_macro    ;go start motor/speech
      JSR   Notrdy             ;Do / get  status for speech and motor
```

```
        LDA   #HidePeek_lo      ;get macro lo byte
        STA   Macro_Lo    ;save lo byte of Macro table entry
        LDA   #HidePeek_hi      ;get macro hi byte
        STA   Macro_Hi    ;save hi byte of Macro table entry
        JSR   Get_macro   ;go start motor/speech
        JSR   Notrdy            ;Do / get  status for speech and motor

Gam_hide2:
        JSR   HideS_timer ;go dec bored timer without Idle

        JSR   Test_all_sens     ;go check all sensors
        LDA   Stat_4            ;get all switches
        AND   #Do_invert  ;ck if inverted
        BEQ   Gam_hide2a  ;jump if not inverted
;       JMP   Gam_hide9   ;abort game and call game lost speech
        JSR   Clear_all_gam     ;go clear all status, cancle games
        JMP   End_all_games     ;done go say "me done"

Gam_hide2a:
        LDA   HCEL_LO           ;ck for no action timeout
        BNE   Gam_hide2   ;wait till done to start game

        LDA   #00         ;clear all sensor flags
        STA   Stat_4            ;

        LDA   #242        ;set timer for 3 min (242 * .742)
        STA   HCEL_LO           ;reset

Gam_hide4:
        LDA   #80h        ;get random/sequential split
        STA   IN_DAT            ;save for random routine
        LDX   #00         ;make sure only gives random
        LDA   #10h        ;get number of random selections (0-6f)
        JSR   Ran_seq           ;go decide random
        AND   #0Fh        ;and nnot >16
        TAX
        LDA   Hide_time,X ;get random timer for speech
        STA   Sensor_timer      ;

Gam_hide5:
        JSR   Test_all_sens     ;go check all sensors
        LDA   Stat_4            ;get sensor status
        AND   #Do_tilt    ;ck if tilt sw req
        BNE   Gam_hide8   ;jump if requested

        JSR   HideS_timer ;go dec bored timer & sensor_timer
        LDA   HCEL_LO           ;get elapsed
        BEQ   Gam_hide9   ;game over

        LDA   Sensor_timer      ;get random speech timer
        BNE   Gam_hide5   ;loop till done

; GO SAY RANDOM WORDS TO HELP FIND HIM

        LDA   #80h        ;get random/sequential split
        STA   IN_DAT            ;save for random routine
        LDX   #00h        ;make sure only gives random
        LDA   #10h        ;get number of random selections
        JSR   Ran_seq           ;go get random selection
        LDA   TEMP1       ;get decision
```

```
      CLC
      ROL   A           ;2's offsett
      TAX
      LDA   Hideseek,X  ;get macro lo byte
      STA   Macro_Lo    ;save lo byte of Macro table entry
      INX
      LDA   Hideseek,X  ;get macro hi byte
      STA   Macro_Hi    ;save hi byte of Macro table entry
      JSR   Get_macro   ;go start motor/speech
      JSR   Notrdy            ;Do / get  status for speech and motor
      JMP   Gam_hide4

Gam_hide8:  ;GAME WON SPEECH

      JSR   Clear_all_gam     ;go clear all status, cancel game

      LDA   #Hidskwon_lo      ;get macro lo byte
      STA   Macro_Lo    ;save lo byte of Macro table entry
      LDA   #Hidskwon_hi      ;get macro hi byte
      STA   Macro_Hi    ;save hi byte of Macro table entry
      JMP   Start_macro ;go set group/table pointer for motor & spch

Gam_hide9:  ;GAME LOST SPEECH

      JSR   Clear_all_gam     ;go clear all status, cancel game
      LDA   #03         ;number of times to call "nana"
      STA   HCEL_HI
Gam_hide9a:
      LDA   #Hidsklost_lo     ;get macro lo byte
      STA   Macro_Lo    ;save lo byte of Macro table entry
      LDA   #Hidsklost_hi     ;get macro hi byte
      STA   Macro_Hi    ;save hi byte of Macro table entry
      JSR   Get_macro   ;go start motor/speech
      JSR   Notrdy            ;Do / get  status for speech and motor
      DEC   HCEL_HI           ;loop
      BNE   Gam_hide9a  ;
      JMP   Idle        ;done

HideS_timer:
      LDA   Milisec_flag      ;if >0 then 742 mili seconds have passed
      BEQ   HideS_tdn   ;bypass if 0
      LDA   #00         ;clear it
      STA   Milisec_flag      ;reset
      LDA   HCEL_LO           ;get current timer * 742mSec sec
      BEQ   HideS_t2    ;do nothing if 0
      DEC     HCEL_LO         ;-1
HideS_t2:
      LDA   Sensor_timer      ;get current timer * 742mSec sec
      BEQ   HideS_tdn   ;do nothing if 0
      DEC   Sensor_timer      ;-1
HideS_tdn:
      RTS               ;


Hide_time:  ;for random time between calls when hiding
      DB    6     ;5 sec  (x * .742)
      DB    7
      DB    8
      DB    9
      DB    10
```

```
        DB      11
        DB      12
        DB      13
        DB      14
        DB      15
        DB      16
        DB      17
        DB      18
        DB      19
        DB      20      ;15 sec
        DB      10


Hideseek:       ;table of sound when Furby is hiding & waiting to be found
        DW      437     ;
        DW      438
        DW      95
        DW      96
        DW      97
        DW      451
        DW      452
        DW      437
        DW      437
        DW      438
        DW      95
        DW      96
        DW      97
        DW      451
        DW      452
        DW      438


;*****************************************************************
;
; Furby - Says ;;;;

; Four byte of ram allocated for game and 5th byte is game counter.
; On start, get 4 random numbers and set the game counter to 4
sequences.
; Furby plays the 4 sounds and waits for the sensors to respond. If its
; wrong, then start over at beginning and if it is right then say
whoppee
; and increment to 5 sounds,,,,,, until all 16. If 16 correct then get
; 4 new random numbers and continue with 16 sequences.
; The invert switch bails out of the game.


Simondelay_lo       EQU     #66h    ;using macro 102 for delay between speech
Simondelay_hi       EQU     #00h    ;hi byte adrs 102 = 066h

Listen_me_lo        EQU     DAh     ;on start up he say "Listen Me"
Listen_me_hi        EQU     01h     ;macro 474 = 1DAh

Simon_frnt_lo       EQU     #AEh    ;using macro 430 for simon chooses
"tickle"
Simon_frnt_hi       EQU     #01h    ;hi byte adrs 430 = 1AEh

Simon_back_lo       EQU     #AFh    ;using macro 431 for simon chooses  "pet
Simon_back_hi       EQU     #01h    ;hi byte adrs 431 = 1AFh
```

```
Simon_snd_lo      EQU   #B0h  ;using macro 432 for simon chooses "sound
Simon_snd_hi      EQU   #01h  ;hi byte adrs 432 = 1B0h

Simon_lght_lo     EQU   #B1h  ;using macro 433 for simon chooses "light
Simon_lght_hi     EQU   #01h  ;hi byte adrs 433 = 1B1h

Skeyfrnt_lo EQU   #0Fh  ;using macro 15 for user feed back
Skeyfrnt_hi EQU   #00h  ;use for "front"

Skeybck_lo  EQU   #B2h  ;using macro 434 for user feed back
Skeybck_hi  EQU   #01h  ;use for "back"

Skeylght_lo EQU   #B3h  ;using macro 435 for user feed back
Skeylght_hi EQU   #01h  ;use for "light"

Skeysnd_lo  EQU   #B4h  ;using macro 436 for user feed back
Skeysnd_hi  EQU   #01h  ;use for "sound"

Simonlost_lo      EQU   #D8h  ; lost game is macro 472
Simonlost_hi      EQU   #01


; Available ram not in use during this game

;HCEL_LO    Counter of which sensor were on

;HCEL_HI    Random play ram 1
;BIT_CT     Random play ram 2
;Task_ptr   Random play ram 3
;Bored_count       Random play ram 4

;TEMP5             Random save ram 1  ( was TMA_INT ) TEMP5 used in
'RAN_SEQ'
;Temp_ID2   Random save ram 2
;Temp_ID    Random save ram 3
;Learn_temp Random save ram 4


Game_simon:
; do delay before start of game

      LDA   #Simondelay_lo    ;get macro lo byte
      STA   Macro_Lo    ;save lo byte of Macro table entry
      LDA   #Simondelay_hi    ;get macro lo byte
      STA   Macro_Hi    ;save hi byte of Macro table entry
      JSR   Get_macro   ;go start motor/speech
      JSR   Notrdy            ;Do / get  status for speech and motor

      LDA   Name        ;current setting for table offset
      CLC
      ROL   A           ;2's comp
      TAX
      LDA   Name_table,X      ;get lo byte
      STA   Macro_Lo    ;save lo byte of Macro table entry
      INX               ;
      LDA   Name_table,X      ;get hi byte
      STA   Macro_Hi    ;save hi byte of Macro table entry
      JSR   Get_macro   ;go start motor/speech
      JSR   Notrdy            ;Do / get  status for speech and motor
```

```
      LDA   #Listen_me_lo     ;get macro lo byte
      STA   Macro_Lo    ;save lo byte of Macro table entry
      LDA   #Listen_me_hi     ;get macro lo byte
      STA   Macro_Hi    ;save hi byte of Macro table entry
      JSR   Get_macro   ;go start motor/speech
      JSR   Notrdy            ;Do / get  status for speech and motor

      LDA   #Simondelay_lo    ;get macro lo byte
      STA   Macro_Lo    ;save lo byte of Macro table entry
      LDA   #Simondelay_hi    ;get macro lo byte
      STA   Macro_Hi    ;save hi byte of Macro table entry
      JSR   Get_macro   ;go start motor/speech
      JSR   Notrdy            ;Do / get  status for speech and motor

      LDA   #04         ;number of sensors in 1st game
GS_rentr:
      STA   HCEL_LO           ;load counter
      STA   IN_DAT            ;save for later use
      JSR   Simon_random      ;go load 2 grps of 4 ram locations
Simon1:
            LDA   HCEL_HI           ;get  1st ram location
      JSR   Simon_sensor      ;go to speech
      JSR   Rotate_play ;get next 2 bits for sensor choice
      DEC   IN_DAT            ;-1 (number of sensors played this game)
      BNE   Simon1            ;loop til all speech done

      JSR   Recover_play      ;reset random rams
      LDA   #GameT_reload     ;reset timer
      STA   Bored_timer ;set
      LDA   #00
      STA   Stat_4            ;clear all sensors
      LDA   HCEL_LO           ;get counter
      STA   IN_DAT            ;reset it
Simon2:
      JSR   Test_all_sens     ;go check all sensors
      LDA   Stat_4            ;get em
      BNE   Simon3            ;jump if any triggered
      JSR   Simon_timer ;go check for timeout
      LDA   Bored_timer ;
      BNE   Simon2            ;loop if not
      JMP   Simon_over  ;bailout if 0
Simon3:
; do to lack of time I resort to brute force ... YUK....
      LDA   Stat_4            ;get which sensor
      CMP   #08h        ;front sw
      BNE   Simon3a           ;jump if not
      LDA   #Skeyfrnt_lo      ;get macro lo byte
      STA   Macro_Lo    ;save lo byte of Macro table entry
      LDA   #Skeyfrnt_hi      ;get macro hi byte
      JMP   Simon3dn    ;go speak it
Simon3a:
      CMP   #10h        ;back sw
      BNE   Simon3b           ;jump if not
      LDA   #Skeybck_lo ;get macro lo byte
      STA   Macro_Lo    ;save lo byte of Macro table entry
      LDA   #Skeybck_hi ;get macro hi byte
      JMP   Simon3dn    ;go speak it
Simon3b:
        CMP #04h        ;light
```

```
        BNE   Simon3c           ;jump if not
        LDA   #Skeylght_lo      ;get macro lo byte
        STA   Macro_Lo    ;save lo byte of Macro table entry
        LDA   #Skeylght_hi      ;get macro hi byte
        JMP   Simon3dn    ;go speak it
Simon3c:
        CMP   #01h        ;sound
        BNE   Simon3d           ;jump if not
        LDA   #Skeysnd_lo ;get macro lo byte
        STA   Macro_Lo    ;save lo byte of Macro table entry
        LDA   #Skeysnd_hi ;get macro hi byte
        JMP   Simon3dn    ;go speak it
Simon3d:
        CMP   #Do_invert  ;?
        BEQ   Simon3e           ;jump if is invert
        LDA   #00         ;
        STA   Stat_4            ;clear sensor flags
        JMP   Simon2            ;ignore all other sensors loop up
Simon3e:
        JMP   Simon_over  ;bail out if is

Simon3dn:
        STA   Macro_Hi    ;save for macro call
        JSR   Get_macro   ;go start motor/speech
        JSR   Notrdy            ;Do / get  status for speech and motor

              LDA   HCEL_HI           ;get  1st ram location
        AND   #03         ;bit 0 & 1
        TAX               ;point to interpret table entry
        LDA   Simon_convert,X   ;translat game to sensors
        CMP   Stat_4            ;ck for correct sensor
        BNE   Simon_lost  ;done if wrong
        LDA   #00
        STA   Stat_4            ;clear all sensors
        JSR   Rotate_play ;get next 2 bits for sensor choice
        DEC   IN_DAT            ;-1 (number of sensors played this game)
        BNE   Simon2            ;loop til all sensors done
        JSR   Simon_won   ;game won
        JSR   Recover_play      ;reset random rams
        INC   HCEL_LO           ;increase number of sensors in next game
        CLC
        LDA   HCEL_LO           ;get current
        STA   IN_DAT            ;reset game sensor counter
        SBC   #16         ;ck if max number of sensors
        BCS   Simon4            ;
        JMP   Simon1            ;loop up
Simon4:
        LDA   #16         ;set to max
        JMP   GS_rentr    ;start next round


;;;;;; Simon subroutines

Simon_lost:
;       LDA   Stat_4            ;ck for invert sw to end game
;       CMP   #Do_invert  ;?
;       BEQ   Simon_over  ;bail out if is

        LDA   #Simonlost_lo     ;get macro lo byte
```

```
      STA   Macro_Lo     ;save lo byte of Macro table entry
      LDA   #Simonlost_hi      ;get macro hi byte
      STA   Macro_Hi     ;save hi byte of Macro table entry
      JSR   Get_macro    ;go start motor/speech
      JSR   Notrdy             ;Do / get  status for speech and motor
      JMP   Game_simon   ;start at beginning

Simon_won:
      LDA   HCEL_LO            ;game number (how many steps)
      CLC
      ROL   A            ;2's offsett for speech win table
      TAX                ;
      LDA   Simon_won_tbl,X    ;get lo byte
      STA   Macro_Lo     ;save lo byte of Macro table entry
      INX                ;
      LDA   Simon_won_tbl,X    ;get hi byte
      STA   Macro_Hi     ;save hi byte of Macro table entry
      JSR   Get_macro    ;go start motor/speech
      JSR   Notrdy             ;Do / get  status for speech and motor
      RTS


Rotate_play:
      ROR   Bored_count ;shfl to carry
      ROR   Task_ptr     ;carry & shfl to carry
      ROR   BIT_CT             ;carry & shfl to carry
      ROR   HCEL_HI            ;carry & shfl to carry throw away lo bit
      ROR   Bored_count ;shfl to carry
      ROR   Task_ptr     ;carry & shfl to carry
      ROR   BIT_CT             ;carry & shfl to carry
      ROR   HCEL_HI            ;carry & shfl to carry throw away lo bit
      RTS                ;

Recover_play:
      LDA   TEMP5        ;recover random data
      STA   HCEL_HI
      LDA   Temp_ID2
      STA   BIT_CT
      LDA   Temp_ID
      STA   Task_ptr
      LDA   Learn_temp
      STA   Bored_count
      RTS                ;
;
Simon_over:
      JSR   Clear_all_gam      ;go clear all status, cancel game
      LDA   #00          ;
      STA   Task_ptr     ;reset for normal use
      JMP   End_all_games      ;done go say "me done"
;
Simon_sensor:
      AND   #03h         ;get senosr
      CLC
      ROL   A            ;2s offset
      TAX                ;offset
      LDA   Psimon_table,X     ;
      STA   Macro_Lo     ;
      INX                ;
      LDA   Psimon_table,X     ;
      STA   Macro_Hi     ;save hi byte of Macro table entry
```

```
      JSR   Get_macro    ;go start motor/speech
      JSR   Notrdy             ;Do / get  status for speech and motor
      RTS                ;
;
Simon_delay:
      LDA   #Simondelay_lo     ;get macro lo byte
      STA   Macro_Lo     ;save lo byte of Macro table entry
      LDA   #Simondelay_hi     ;get macro hi byte
      STA   Macro_Hi     ;save hi byte of Macro table entry
      JSR   Get_macro    ;go start motor/speech
      JSR   Notrdy             ;Do / get  status for speech and motor
      RTS                ;
;
Simon_random:
      JSR   Random             ;get random number (0-255)
      STA   TEMP5        ;   "
      STA   HCEL_HI
      JSR   Random             ;get random number (0-255)
      STA   Temp_ID2     ;   "   "
      STA   BIT_CT
      JSR   Random             ;get random number (0-255)
      STA   Temp_ID            ;   "
      STA   Task_ptr
      JSR   Random             ;get random number (0-255)
      STA   Learn_temp   ;   "
      STA   Bored_count
      RTS
;
Simon_timer:
      LDA   Milisec_flag       ;if >0 then 742 mili seconds have passed
      BEQ   Simon_tdn    ;bypass if 0
      LDA   #00          ;clear it
      STA   Milisec_flag       ;reset

      LDA   Bored_timer ;get current timer * 742mSec sec
      BEQ   Simon_tdn    ;do nothing if 0
      DEC   Bored_timer ;-1
Simon_tdn:
      RTS                ;
;
Psimon_table:
      DW    430    ;front switch   ( 00 )
      DW    431    ;back switch    ( 01 )
      DW    433    ;sound sensor   ( 11 ) (lt & snd swaped in table)
      DW    432    ;light sensor   ( 10 )
;
Simon_convert:     ;converts game table to sensor table
      DB    08h    ;front sw
      DB    10h    ;back sw
      DB    04h    ;light
      DB    01h    ;sound
;
Simon_won_tbl:     ;for each game won there is a macro (or re-use them)
      DW    72     ; 0 (not used,,,, place holder)
      DW    72     ; 1 (not used,,,, place holder)
      DW    72     ; 2 (not used,,,, place holder)
      DW    72     ; 3 (not used,,,, place holder)

      DW    72     ; 4  (1st game has 4 sensors, each game adds one)
      DW    72     ; 5
```

```
      DW    72    ; 6
      DW    72    ; 7
      DW    380   ; 8
      DW    380   ; 9
      DW    380   ; 10
      DW    380   ; 11
      DW    471   ; 12
      DW    471   ; 13
      DW    471   ; 14
      DW    471   ; 15
      DW    439   ; 16

;

End_all_games:    ;when any game ends, they jump here and say done

Saygamdn_lo EQU   #D9h  ;using macro 473 for game over speech
Saygamdn_hi EQU   #01h  ;


      LDA   #Bored_reld ;reset bored timer
      STA   Bored_timer ;

      LDA   #Saygamdn_lo      ;get macro lo byte
      STA   Macro_Lo    ;save lo byte of Macro table entry
      LDA   #Saygamdn_hi      ;get macro hi byte
      STA   Macro_Hi    ;save hi byte of Macro table entry
      JMP   Start_macro ;go set group/table pointer for motor & spch

;*****************************************************************

;Burp attack egg

Burpsnd_lo  EQU   #D6h  ;using macro 470 for user feed back
Burpsnd_hi  EQU   #01h  ;


Game_Burp:

      JSR   Clear_all_gam

      LDA   #Bored_reld ;reset bored timer
      STA   Bored_timer ;

      LDA   #Burpsnd_lo ;get macro lo byte
      STA   Macro_Lo    ;save lo byte of Macro table entry
      LDA   #Burpsnd_hi ;get macro hi byte
      STA   Macro_Hi    ;save hi byte of Macro table entry
      JMP   Start_macro ;go set group/table pointer for motor & spch



;
;*****************************************************************

;easter egg says NAME

Game_name:
```

```
      JSR   Clear_all_gam

      LDA   #Bored_reld ;reset bored timer
      STA   Bored_timer ;

      LDA   Name        ;current setting for table offset
      CLC
      ROL   A           ;2's comp
      TAX
      LDA   Name_table,X      ;get lo byte
      STA   Macro_Lo    ;save lo byte of Macro table entry
      INX               ;
      LDA   Name_table,X      ;get hi byte
      STA   Macro_Hi    ;save hi byte of Macro table entry
      JMP   Start_macro ;go set group/table pointer for motor & spch
;
;*************************************************************

;Twinkle song egg

; When song is complete, if both front and back switches are pressed
; we goto deep sleep. That means only the invert can wake us up, not
; the invert switch.


Twinklsnd_lo      EQU   #D5h  ;using macro 469
Twinklsnd_hi      EQU   #01h  ;

Sleep_lo    EQU   #A6h  ;using macro 166 (before going to sleep)
Sleep_hi    EQU   #00h  ;

Game_twinkle:

      JSR   Clear_all_gam
      LDA   #03         ;song counter
      STA   HCEL_LO           ;set
Gtwnk:
      DEC   HCEL_LO           ;-1
      LDA   Stat_2            ;Get system clear done flags
      AND   #Not_tch_ft ;clear previously inverted flag
      AND   #Not_tch_bk ;clear previously inverted flag
      STA   Stat_2            ;update

      LDA   #Bored_reld ;reset bored timer
      STA   Bored_timer ;

      LDA   #Twinklsnd_lo     ;get macro lo byte
      STA   Macro_Lo    ;save lo byte of Macro table entry
      LDA   #Twinklsnd_hi     ;get macro hi byte
      STA   Macro_Hi    ;save hi byte of Macro table entry
      JSR   Get_macro   ;go start motor/speech
      JSR   Notrdy            ;Do / get  status for speech and motor
      JSR   Test_all_sens     ;get status
      JSR   Test_all_sens     ;get status 2nd time for debounce
      LDA   Stat_4            ;switch status
      AND   #18h        ;isolate front and back switches
      CMP   #18h        ;
      BEQ   Start_sleep ;if both switches pressed, goto sleep
      LDA   HCEL_LO           ;get song loop counter
      BNE   Gtwnk       ;loop
```

```
      JMP   Idle          ;not so egg complete

Start_sleep:
      LDA   #Sleep_lo     ;get macro lo byte
      STA   Macro_Lo      ;save lo byte of Macro table entry
      LDA   #Sleep_hi     ;get macro hi byte
      STA   Macro_Hi      ;save hi byte of Macro table entry
      JSR   Get_macro     ;go start motor/speech
      JSR   Notrdy              ;Do / get  status for speech and motor
      LDA   #11h          ;set deep sleep mode
      STA   Deep_sleep    ;
      JMP   GoToSleep     ;nity-night
;
;*************************************************************

;Rooster loves you  egg

Roostersnd_lo     EQU   #D4h  ;using macro 468
Roostersnd_hi     EQU   #01h  ;


Game_rooster:

      JSR   Clear_all_gam

      LDA   #Bored_reld ;reset bored timer
      STA   Bored_timer ;

      LDA   #Roostersnd_lo    ;get macro lo byte
      STA   Macro_Lo    ;save lo byte of Macro table entry
      LDA   #Roostersnd_hi    ;get macro hi byte
      STA   Macro_Hi    ;save hi byte of Macro table entry
      JMP   Start_macro ;go set group/table pointer for motor & spch

;*************************************************************
;

; If a game requires sensor input without triggering the normal
; sensor cycle for speech, then this rtn will check all sensors for
; change and the calling game can check for the appropriate trigger
; DO NOT USE I.R. SENSOR SINCE ITS RAM LOCATIONS ARE USED IN GAMES

Test_all_sens:
      JSR   Get_back    ;
      JSR   Get_Tilt    ;
      JSR   Get_invert  ;
      JSR   Get_front   ;
      JSR   Get_light   ;
      JSR   Get_sound   ;
      JSR   Get_feed    ;
      RTS               ;back to game



;*************************************************************
;*************************************************************
;*************************************************************

;***** Side  all switch triggers when ball falls off center and I/O goes
```

```
hi.

CK_tilt:          ;tilt sensor
      JSR   Get_Tilt     ;go ck for sensor trigger
      BCS   Normal_tilt ;go fini normal spch/motor table
      JMP   Idle         ;no request

Get_Tilt:   ;this is the subroutine entry point.
      LDA   Port_D             ;get I/O
      AND   #Ball_side   ;ck if we tilted on side
      BNE   Do_bside     ;jump if hi

      LDA   Stat_2             ;Get system
      AND   #Not_bside   ;clear previously on side flag
      STA   Stat_2             ;update
Side_out:
      CLC                ;clear indicates no request
      RTS                ;

Do_bside:
      LDA   Stat_2             ;system
      AND   #Bside_dn    ;ck if previously done
      BNE   Side_out     ;jump if was
      LDA   Stat_2             ;get system
      ORA   #Bside_dn    ;flag set ,only execute once
      STA   Stat_2             ;update system

      LDA   Stat_4             ;game mode status
      ORA   #Do_tilt     ;flag sensor is active
      STA   Stat_4             ;update
      SEC                ;carry set indicates sensor is triggered
      RTS                ;

Normal_tilt:      ;Idle rtn jumps here to complete speech/motor table


;;;;;;;  also for testing, when tilt is triggered, it resets all
;        easter egg routines to allow easy entry of eggs.


      JSR   Clear_all_gam      ;


;*****************************************************************

      JSR   Life         ;go tweek health/hungry counters
      BCS   More_tilt    ;if clear then do sensor else bail
      JMP   Idle         ;done
More_tilt:

;*****************************************************************

      LDA   #Tilt_split ;get random/sequential split
      STA   IN_DAT             ;save for random routine

      LDX   #Seq_tilt    ;get how many sequential selections
      LDA   #Ran_tilt    ;get number of random slections
      JSR   Ran_seq            ;go decide random/sequential
```

```
      LDX   Sensor_timer      ;get current for training subroutine

      BCS   Tilt_ran    ;Random mode when carry SET

      LDA   Sensor_timer      ;ck if timed out since last action
      BEQ   Tilt_reset  ;yep

      LDA   Tilt_count  ;save current
      STA   BIT_CT            ;temp store

      INC   Tilt_count  ;if not then next table entry
      LDA   Tilt_count  ;get
      CLC
      SBC   #Seq_tilt-1 ;ck if > assignment
      BCC   Tilt_side   ;jump if <
      LDA   #Seq_tilt-1 ;dont inc off end
      STA   Tilt_count  ;
      JMP   Tilt_side   ;do it
Tilt_reset:
      LDA   #00         ;reset to 1st entry of sequential
      STA   BIT_CT            ;temp store
      STA   Tilt_count  ;
Tilt_side:
      LDA   #Global_time      ;get timer reset value
      STA   Sensor_timer      ;reset it
      LDA   BIT_CT            ;Acc holds value for subroutine

Tilt_ran:
      STA   IN_DAT            ;save decision
      LDA   #Tilt_ID    ;which ram location for learned word count
(offset)
      JSR   Start_learn ;go record training info
      LDA   IN_DAT            ;get decision

      JSR   Decid_age   ;do age calculation for table entry
      LDX   TEMP0       ;age offset
      LDA   Tilt_S1,X   ;get lo byte
      STA   Macro_Lo    ;save lo byte of Macro table entry
      INX               ;
      LDA   Tilt_S1,X   ;get hi byte
      STA   Macro_Hi    ;save hi byte of Macro table entry
      JMP   Start_macro ;go set group/table pointer for motor & spch
;
;
;*****************************************************************
;*****************************************************************
;*****************************************************************
;
;
;
;***** Inverted ball switch triggers when ball touches top and I/O goes
hi.

Ck_invert:              ; upside down sense

      JSR   Get_invert  ;go ck for sensor trigger
      BCS   Normal_invert     ;go fini normal spch/motor table
      JMP   Idle        ;no request

Get_invert: ;this is the subroutine entry point.
```

```
      LDA   Port_D              ;get I/O
      AND   #Ball_invert        ;ck if we upside down

      BNE   Do_binvrt   ;jump if inverted (hi)

      LDA   Stat_2              ;Get system
      AND   #Not_binvrt ;clear previously inverted flag
      STA   Stat_2              ;update
Invrt_out:
      CLC               ;clear carry indicates no sensor change
      RTS               ;

Do_binvrt:
      LDA   Stat_2              ;get system
      AND   #Binvrt_dn  ;ck if prev done
      BNE   Invrt_out   ;jump if was
      LDA   Stat_2              ;get system
      ORA   #Binvrt_dn  ;flag set ,only execute once
      STA   Stat_2              ;update system

      LDA   Stat_4              ;game mode status
      ORA   #Do_invert  ;flag sensor is active
      STA   Stat_4              ;update

      SEC               ;set indicates sensor is triggered
      RTS               ;

Normal_invert:

;*****************************************************************

      JSR   Life        ;go tweek health/hungry counters
      BCS   More_invert ;if clear then do sensor else bail
      JMP   Idle        ;done
More_invert:

;*****************************************************************


      LDA   #Invert_split       ;get random/sequential split
      STA   IN_DAT              ;save for random routine

      LDX   #Seq_invert ;get how many sequential selections
      LDA   #Ran_invert ;get number of random slections
      JSR   Ran_seq             ;go decide random/sequential

      LDX   Sensor_timer        ;get current  for training subroutine

      BCS   Invrt_rnd   ;Random mode when carry SET

      LDA   Sensor_timer        ;ck if timed out since last action
      BEQ   Invrt_reset ;yep

      LDA   Invrt_count ;save current
      STA   BIT_CT              ;temp store

      INC   Invrt_count ;if not then next table entry
      LDA   Invrt_count ;get
```

```
      CLC
      SBC   #Seq_invert-1     ;ck if > assignment
      BCC   Invrt_set   ;jump if <
      LDA   #Seq_invert-1     ;dont inc off end
      STA   Invrt_count ;
      JMP   Invrt_set   ;do it
Invrt_reset:
      LDA   #00         ;reset to 1st entry of sequential
      STA   BIT_CT            ;temp store
      STA   Invrt_count ;
Invrt_set:
      LDA   #Global_time      ;get timer reset value
      STA   Sensor_timer      ;reset it
      LDA   BIT_CT            ;speech to call

Invrt_rnd:

      STA   IN_DAT            ;save decision
      LDA   #Invert_ID  ;which ram location for learned word count
(offset)
      JSR   Start_learn ;go record training info
      LDA   IN_DAT            ;get back word to speak

      JSR   Decid_age   ;do age calculation for table entry
      LDX   TEMP0       ;age offset
      LDA   Invrt_S1,X  ;get lo byte
      STA   Macro_Lo    ;save lo byte of Macro table entry
      INX               ;
      LDA   Invrt_S1,X  ;get hi byte
      STA   Macro_Hi    ;save hi byte of Macro table entry
      JMP   Start_macro ;go set group/table pointer for motor & spch
;
;
;*****************************************************************
;*****************************************************************
;*****************************************************************
;*****************************************************************
;
Ck_back:          ;Back touch sensor

      JSR   Get_back    ;go ck for sensor trigger
      BCS   Normal_back ;go fini normal spch/motor table
      JMP   Idle        ;no request

Get_back:   ;this is the subroutine entry point.

      LDA   Port_C            ;get I/O
      AND   #Touch_bck  ;ck if Firby's back is rubbed
      BEQ   Do_tch_bk   ;jump if lo
      LDA   Stat_2            ;Get system
      AND   #Not_tch_bk ;clear previously inverted flag
      STA   Stat_2            ;update
Tch1_out:
      CLC               ;clear carry for no sensor request
      RTS               ;

Do_tch_bk:
      LDA   Stat_2            ;get system
      AND   #Tchbk_dn   ;ck if prev done
      BNE   Tch1_out    ;jump if was
```

```
        LDA   Stat_2              ;get system
        ORA   #Tchbk_dn    ;flag set ,only execute once
        STA   Stat_2              ;update system

        LDA   Stat_4              ;game mode status
        ORA   #Do_back     ;flag sensor is active
        STA   Stat_4              ;update
        SEC                ;set indicates sensor is triggered
        RTS                ;

Normal_back:       ;enter here to complere sensor speech/motor

;*****************************************************************

        JSR   Life         ;go tweak health/hungry counters
        BCS   More_back    ;if clear then do sensor else bail
        JMP   Idle         ;done
More_back:

;*****************************************************************


        LDA   #Back_split ;get random/sequential split
        STA   IN_DAT              ;sa   for random routine

        LDX   #Seq_back    ;get how many sequential selections
        LDA   #Ran_back    ;get number of random slections
        JSR   Ran_seq             ;go decide random/sequential

        LDX   Sensor_timer        ;get current  for training subroutine

        BCS   Back_rnd     ;Random mode when carry SET

        LDA   Sensor_timer        ;ck if timed out since last action
        BEQ   Back_reset  ;yep

        LDA   Tchbck_count        ;save current
        STA   BIT_CT              ;temp store

        INC   Tchbck_count        ;if not then next table entry
        LDA   Tchbck_count        ;get
        CLC
        SBC   #Seq_back-1 ;ck if > assignment
        BCC   Back_set     ;jump if <
        LDA   #Seq_back-1 ;dont inc off end
        STA   Tchbck_count        ;
        JMP   Back_set     ;do it
Back_reset:
        LDA   #00          ;reset to 1st entry of sequential
        STA   BIT_CT              ;temp store
        STA   Tchbck_count        ;
Back_set:
        LDA   #Global_time        ;get timer reset value
        STA   Sensor_timer        ;reset it
        LDA   BIT_CT              ;get current pointer to tables

Back_rnd:

        STA   IN_DAT              ;save decision
        LDA   #Back_ID     ;which ram location for learned word count
```

```
(offset)
      JSR   Start_learn ;go record training info
      LDA   IN_DAT            ;get back word to speak

      JSR   Decid_age   ;do age calculation for table entry
      LDX   TEMP0       ;age offset
      LDA   Tback_S1,X  ;get lo byte
      STA   Macro_Lo    ;save lo byte of Macro table entry
      INX               ;
      LDA   Tback_S1,X  ;get hi byte
      STA   Macro_Hi    ;save hi byte of Macro table entry
      JMP   Start_macro ;go set group/table pointer for motor & spch
;
;
;*****************************************************************
;*****************************************************************
;*****************************************************************
;*****************************************************************
;
; The IR routine turns interrupts off for 100 Msec, which stops the
; timing chain (multiplies time by 100). This front end leaves
; interrupts on and sits in a loop for 5 msec to determine if I.R. is
; active and if so, executes normal I.R. routine, else exits.

;********* start Tracker


;The way to include the IR program, I list as the following:
;It shows the program prargraph from Ck_IR: to Ck_front:
;of couse, It also attach the IR.asm file
;the IR.asm file I just make a little bit change, to make they work at
;any system clock assume by constant SystemClock:
;please advise.. :>

Ck_IR:
      LDA   Last_IR           ;timer stops IR from hearing own IR xmit
      BEQ   CKIR_S            ;jump if timer 0
      JMP   Idle        ;abort if >0
CKIR_S:
      LDA   #FFh        ;set loop timer
      STA   TEMP1       ;
      LDA   #10h        ;set gross timer
      STA   TEMP2       ;
IR_req:
      LDA   Port_B            ;ck if IR signal active (hi)
      AND   #IR_IN            ;get port pin
      BNE   Got_IR            ;go do input if active
      LDA   Port_B            ;ck if IR signal active (hi)
      AND   #IR_IN            ;get port pin
      BNE   Got_IR            ;go do input if active
      DEC   TEMP1       ;inside loop
      BNE   IR_req            ;
      LDA   #FFh        ;reset loop timer
      STA   TEMP1       ;
      DEC   TEMP2       ;outside loop
      BNE   IR_req            ;loop thru
      JMP   Idle        ;no activity found

Got_IR:
      LDA   #05         ;number of times to ck for IR reception
```

```
      STA   TEMP4       ;
Got_IR2:
        JSR     D_IR_test       ;used as a subroutine for diags
        BCS     New_IR          ;jump if found data
      DEC   TEMP4       ;
      BNE   Got_IR2           ;loop
        JMP     Idle            ;bail out if not
New_IR:
        JMP     Normal_IR

;*************************
; Begin Koball's code
;*************************

D_IR_test:
        SEI                     ;;Tracker
        JSR     GBYTE           ;;Tracker       First time to read
        LDA     #Intt_dflt      ;Initialize timers, etc.
;;Tracker
        STA     Interrupts      ;load reg
;;Tracker
        LDA     IN_DAT          ;;load result to ACC
        CLI                     ;;Tracker

        RTS

Normal_IR:
; There are 4 I.R. table arranged as all other tables, one for each age.
; But here we get a random number which determines which one of the
; four tables we point to and the actual number received is the one of
; sixteen selection.


        LDA     IN_DAT          ;;Tracker add
      AND   #0Fh        ;kill hi nibble (compliment of lo nibble)
      STA   IN_DAT            ;save

      CMP   #08         ;test for special sneeze command
      BNE   No_sneeze   ;jump if not
      LDA   #Really_sick-30   ;force Furby to get sick
      STA   Sick_counter      ;update

No_sneeze:
      LDA   Bored_timer ;get current count
      STA   TEMP1       ;save
Get_IR_rnd:
      JSR   Random            ;get something
      DEC   TEMP1       ;-1
      BNE   Get_IR_rnd  ;loop getting random numbers
      LDA   Seed_1            ;get new random pointer
      AND   #0Fh        ;kill hi nibble
      STA   TEMP1       ;save
      CLC
      SBC   #11         ;ck if > 11
      BCC   NormIR_2    ;jump if not
      LDA   #96         ;point to table 4
      JMP   Got_normIR  ;
NormIR_2:
      LDA   TEMP1       ;recover random number
      CLC
```

```
      SBC   #07          ;ck if > 7
      BCC   NormIR_3     ;jump if not
      LDA   #64          ;point to table 3
      JMP   Got_normIR   ;
NormIR_3:
      LDA   TEMP1        ;recover random number
      CLC
      SBC   #03          ;ck if > 03
      BCC   NormIR_4     ;jump if not
      LDA   #32          ;point to table 2
      JMP   Got_normIR   ;
NormIR_4:
      LDA   #00          ;force table 1

Got_normIR:

      CLC
      ROL   IN_DAT             ;16 bit offset for speech
      CLC
      ADC   IN_DAT             ;create speech field ofsett pointer
      TAX                ;set offset

        LDA     IR_S1,X          ;get lo byte
        STA     Macro_Lo         ;save lo byte of Macro table entry
        INX                      ;
        LDA     IR_S1,X          ;get hi byte
        STA     Macro_Hi         ;save hi byte of Macro table entry
        JMP     Start_macro      ;go set group/table pointer for motor &
spch

        Include         IR2.Asm            ;asm file

;********* end Tracker

;*****************************************************************
;*****************************************************************
;*****************************************************************
;*****************************************************************

Ck_front:          ; touch front  (tummy)

      JSR   Get_front    ;go ck for sensor trigger
      BCS   Normal_front       ;go fini normal spch/motor table
      JMP   Idle         ;no request

Get_front:  ;this is the subroutine entry point.

      LDA   Port_C             ;get I/O
      AND   #Touch_frnt ;ck if Firby's chest is rubbed
      BEQ   Do_tch_ft    ;jump if lo
      LDA   Stat_2             ;Get system
      AND   #Not_tch_ft ;clear previously inverted flag
      STA   Stat_2             ;update
Touch_end:
      CLC                ;clear indicates no sensor request
      RTS                ;
Do_tch_ft:
      LDA   Stat_2             ;get system
      AND   #Tchft_dn    ;ck if prev done
      BNE   Touch_end    ;jump if was
```

```
      LDA   Stat_2              ;get system
      ORA   #Tchft_dn    ;flag set ,only execute once
      STA   Stat_2              ;update system

      LDA   Stat_4              ;game mode status
      ORA   #Do_tummy    ;flag sensor is active
      STA   Stat_4              ;update
      SEC                ;set indicates sensor is triggered
      RTS                ;

Normal_front:            ;enter here to complete sensor speech/motor

;*****************************************************************

      JSR   Life         ;go tweek health/hungry counters
      BCS   More_front   ;if clear then do sensor else bail
      JMP   Idle         ;done
More_front:

;*****************************************************************


      LDA   #Front_split       ;get random/sequential split
      STA   IN_DAT             ;save for random routine

      LDX   #Seq_front   ;get how many sequential selections
      LDA   #Ran_front   ;get sequential split
      JSR   Ran_seq            ;go decide random/sequential

      LDX   Sensor_timer       ;get current  for training subroutine

      BCS   Front_rnd    ;Random mode when carry set

      LDA   Sensor_timer       ;ck if timed out since last action
      BEQ   Front_reset ;yep

      LDA   Tchfrnt_count      ;save current
      STA   BIT_CT             ;temp store

      INC   Tchfrnt_count      ;if not then next table entry
      LDA   Tchfrnt_count      ;get
      CLC
      SBC   #Seq_front-1       ;ck if > assignment
      BCC   Front_set    ;jump if <
      LDA   #Seq_front-1       ;dont inc off end
      STA   Tchfrnt_count      ;
      JMP   Front_set    ;do it
Front_reset:
      LDA   #00          ;reset to 1st entry of sequential
      STA   BIT_CT             ;temp store
      STA   Tchfrnt_count      ;
Front_set:
      LDA   #Global_time       ;get timer reset value
      STA   Sensor_timer       ;reset it
      LDA   BIT_CT             ;get current pointer to tables

Front_rnd:

      STA   IN_DAT             ;save decision
```

```
      LDA   #Front_ID   ;which ram location for learned word count
(offset)
      JSR   Start_learn ;go record training info
      LDA   IN_DAT            ;get back word to speak

      JSR   Decid_age   ;do age calculation for table entry
      LDX   TEMP0       ;age offset
      LDA   Tfrnt_S1,X  ;get lo byte
      STA   Macro_Lo    ;save lo byte of Macro table entry
      INX               ;
      LDA   Tfrnt_S1,X  ;get hi byte
      STA   Macro_Hi    ;save hi byte of Macro table entry
      JMP   Start_macro ;go set group/table pointer for motor & spch

;
;*****************************************************************
;*****************************************************************
;*****************************************************************
;*****************************************************************
;
;
Ck_feed:          ; food sensor
;
      JSR   Get_feed    ;go ck for sensor trigger
      BCS   Normal_feed ;go fini normal spch/motor table

      JMP   Idle        ;no request

Get_feed:   ;this is the subroutine entry point.

; Each trigger increments the health status at a greater rate

; Special enable routine to share port pin D1 with invert switch.
; Feed switch is pulled hi by the DAC1 (aud-a) output only after
; we test the invert line. If invert is not hi, then turn on
; DAC1 and ck feed line on same port D1.


      LDA   Port_D          ;get I/O
      AND   #Ball_invert    ;ck if we are inverted
      BEQ   St_feed         ;jump if not inverted (lo=not inverted)
      CLC             ;indicates no request
      RTS             ;if inverted then bypass
St_feed:
      LDA   #FFh      ;turn DAC2 on to enable feed switch
      STA   DAC2      ;out
      LDA   Port_D          ;get I/O
      AND   #Ball_invert    ;ck if feed switch closed
      BNE   Start_feed  ;jump if hi
      LDA   #00
      STA   DAC2      ;clear feed sw enable
      LDA   Stat_3          ;Get system
      AND   #Not_feed   ;clear previously inverted flag
      STA   Stat_3          ;update
Feed_out:
      CLC             ;clear indicates no request
      RTS             ;go test next

Start_feed:
      LDA   #00
```

```
        STA   DAC2          ;clear feed sw enable

;       LDA   Stat_3              ;get system
;       AND   #Feed_dn      ;ck if prev done
;       BNE   Feed_out      ;jump if was
;       LDA   Stat_3              ;get system
;       ORA   #Feed_dn      ;flag set ,only execute once
;       STA   Stat_3              ;update system

        LDA   Stat_4              ;game mode status
        ORA   #Do_feed      ;flag sensor is active
        STA   Stat_4              ;update
        SEC                 ;set when sensor is triggered
        RTS                 ;

Normal_feed:                ;enter here to complete speech/motor

;*****************************************************************

;  health table calls here and decision for which speech pattern

        LDA   #Food         ;each feeding increments hunger counter
        CLC
        ADC   Hungry_counter      ;feed him!
        BCC   Feeding_dn    ;jump if no roll over
        LDA   #FEh          ;max count
Feeding_dn:
        STA   Hungry_counter      ;update

;;;;; JSR   Life          ;go finish sick/hungry speech


;*****************************************************************

        LDA   #Feed_split ;get random/sequential split
        STA   IN_DAT              ;save for random routine

        LDX   #Seq_feed   ;get how many sequential selections
        LDA   #Ran_feed   ;get random assignment
        JSR   Ran_seq             ;go decide random/sequential

        LDX   Sensor_timer        ;get current  for training subroutine

        BCS   Feedrand    ;Random mode when carry set

        LDA   Sensor_timer        ;ck if timed out since last action
        BEQ   Feed_reset  ;yep

        LDA   Feed_count  ;save current
        STA   BIT_CT              ;temp store

        INC   Feed_count  ;if not then next table entry
        LDA   Feed_count  ;get
        CLC
        SBC   #Seq_feed-1 ;ck if > assignment
        BCC   Feed_set    ;jump if <
        LDA   #Seq_feed-1 ;dont inc off end
        STA   Feed_count  ;
        JMP   Feed_set    ;do it
Feed_reset:
```

```
      LDA   #00          ;reset to 1st entry of sequential
      STA   BIT_CT             ;temp store
      STA   Feed_count   ;
Feed_set:
      LDA   #Global_time       ;get timer reset value
      STA   Sensor_timer       ;reset it
      LDA   BIT_CT             ;get current pointer to tables

Feedrand:

      STA   IN_DAT             ;save decision
      LDA   #Feed_ID     ;which ram location for learned word count
(offset)
      JSR   Start_learn ;go record training info
      LDA   IN_DAT             ;get back word to speak

      JSR   Decid_age    ;do age calculation for table entry
      LDX   TEMP0        ;age offset
      LDA   Feed_S1,X    ;get lo byte
      STA   Macro_Lo     ;save lo byte of Macro table entry
      INX                ;
      LDA   Feed_S1,X    ;get hi byte
      STA   Macro_Hi     ;save hi byte of Macro table entry
      JMP   Start_macro ;go set group/table pointer for motor & spch

;
;*****************************************************************
;*****************************************************************
;*****************************************************************
;*****************************************************************
;
Ck_light:          ;Bright light sensor

      JSR   Get_light    ;now handled as a subroutine
      BCC   Ck_light2    ;jump if new level > reff
      JMP   Idle         ;nothing to do
Ck_light2:
      JMP   Normal_light       ;jump if new level > reff


      Include            Light5.asm  ;asm file


Normal_light:

; below routines are jumped to by light exec if > reff

;*****************************************************************

      JSR   Life         ;go tweek health/hungry counters
      BCS   More_light   ;if clear then do sensor else bail
      JMP   Idle         ;done
More_light:

;*****************************************************************


      LDA   #Light_split       ;get random/sequential split
      STA   IN_DAT             ;save for random routine
```

```
        LDX   #Seq_light  ;get how many sequential selections
        LDA   #Ran_light  ;get sensor split table
        JSR   Ran_seq           ;go decide random/sequential

        LDX   Sensor_timer      ;get current  for training subroutine

        BCS   Lghtrand    ;Random mode when carry set

        LDA   Sensor_timer      ;ck if timed out since last action
        BEQ   Lght_reset  ;yep

        LDA   Lght_count  ;save current
        STA   BIT_CT            ;temp store

        INC   Lght_count  ;if not then next table entry
        LDA   Lght_count  ;get
        CLC
        SBC   #Seq_light-1      ;ck if > assignment
        BCC   Lght_set    ;jump if <
        LDA   #Seq_light-1      ;dont inc off end
        STA   Lght_count  ;
        JMP   Lght_set    ;do it
Lght_reset:
        LDA   #00         ;reset to 1st entry of sequential
        STA   BIT_CT            ;save temp store
        STA   Lght_count  ;
Lght_set:
        LDA   #Global_time      ;get timer reset value
        STA   Sensor_timer      ;reset it
        LDA   BIT_CT            ;get current pointer to tables

Lghtrand:

        STA   TEMP4       ;save seq/rand pointer
        LDA   Stat_3            ;system
        AND   #Lght_stat  ;ck bit for light/dark table
        BEQ   Do_dark           ;jump if clear

        LDA   TEMP4       ;get pointer

        STA   IN_DAT            ;save decision
        LDA   #Light_ID   ;which ram location for learned word count
(offset)
        JSR   Start_learn ;go record training info
        LDA   IN_DAT            ;get back word to speak

        JSR   Decid_age   ;do age calculation for table entry
        LDX   TEMP0       ;age offset
        LDA   Light_S1,X  ;get lo byte
        STA   Macro_Lo    ;save lo byte of Macro table entry
        INX               ;
        LDA   Light_S1,X  ;get hi byte
        STA   Macro_Hi    ;save hi byte of Macro table entry
        JMP   Start_macro ;go set group/table pointer for motor & spch

Do_dark:
        LDA   TEMP4       ;get pointer

        STA   IN_DAT            ;save decision
```

```
      LDA   #Dark_ID    ;which ram location for learned word count
(offset)
      JSR   Start_learn ;go record training info
      LDA   IN_DAT            ;get back word to speak

      JSR   Decid_age   ;do age calculation for table entry
      LDX   TEMP0       ;age offset
      LDA   Dark_S1,X   ;get lo byte
      STA   Macro_Lo    ;save lo byte of Macro table entry
      INX               ;
      LDA   Dark_S1,X   ;get hi byte
      STA   Macro_Hi    ;save hi byte of Macro table entry
      JMP   Start_macro ;go set group/table pointer for motor & spch

;
;*****************************************************************
;*****************************************************************
;*****************************************************************
;*****************************************************************
;

Ck_sound:         ;Audio sensor
      JSR   Get_sound   ;now handled as a subroutine
      BCS   Ck_sound2   ;jump if new level > reff
      JMP   Idle        ;nothing to do
Ck_sound2:
      JMP   Normal_sound      ;jump if new level > reff


Get_sound:  ;alt entry for diagnostics


; The microphone interface generates a square wave of 2k to 100k.
; We can loop on the sense line and count time for the
; hi period to determine if sound has changed and compare it to previous
; samples.


      SEI               ;disable interrupts
      LDX   #00         ;clear
      STX   TEMP1       ;clear buffer
      LDX   #FFh        ;load loop timer
      STX   TEMP2       ;
Ck_snd2:
      DEC   TEMP2       ;
      BEQ   Ck_snd4           ;jump if timed out
      LDA   Port_D            ;get I/O
      AND   #Mic_in           ;ck sound clk is hi
      BEQ   Ck_snd2           ;wait for it to go hi
      LDX   #FFh        ;load loop timer
      STX   TEMP2       ;
Ck_snd3:
      INC   TEMP1       ;count during lo clk   +5
      BEQ   Snd_over    ;jump if rolled over   +3
      LDA   Port_D            ;get I/O                +2
      AND   #Mic_in           ;ck if still hi           +2
      BNE   Ck_snd3           ;loop till lo             +3
(15*166ns=2.49uS)
      JMP   Ck_snd4           ;done
Snd_over:
```

```
;   we should never get here so bail back to idle and this will
;   also prevent system lockup when no clk

      LDA   #250          ;never allow roll over
      STA   TEMP1         ;
Ck_snd4:
      CLI                 ;re-enable interrupt
      JSR   Kick_IRQ      ;wait for motor R/C to start working again
      LDA   TEMP1         ;get count
      CLC                 ;clear
      SBC   #05           ;is diff > 5
      BCC   No_snd              ;bail out if not

      LDA   Stat_3              ;system
      AND   #Sound_stat ;ck for prev done
      BNE   No_snd2             ;wait till quiet

      LDA   Stat_3              ;system
      ORA   #Sound_stat ;
      STA   Stat_3              ;set prev dn

      LDA   Stat_4              ,
      ORA   #Do_snd             ;set indicating change > reff level
      STA   Stat_4              ;

      SEC                 ;carry set indicates no change
      RTS


No_snd:
      LDA   Stat_3              ;get system
      AND   #Nt_snd_stat        ;clear prev dn
      STA   Stat_3              ;update
No_snd2:
      CLC                 ;carry clear indicates no sound
      RTS                 ;done

Normal_sound:

; below routines are jumped to if sound pulse detected


;*****************************************************************

      JSR   Life          ;go tweek health/hungry counters
      BCS   More_sound    ;if clear then do sensor else bail
      JMP   Idle          ;done
More_sound:

;*****************************************************************

      LDA   #Sound_split        ;get random/sequential split
      STA   IN_DAT              ;save for random  utine

      LDX   #Seq_sound    ;get how many sequential selections
      LDA   #Ran_sound    ;number of random selections
      JSR   Ran_seq             ;go decide random/sequential
```

```
      LDX   Sensor_timer      ;get current  for training subroutine

      BCS   Sndrand           ;Random mode when carry set

      LDA   Sensor_timer      ;ck if timed out since last action
      BEQ   Snd_reset   ;yep

      LDA   Sound_count ;save current
      STA   BIT_CT            ;temp store

      INC   Sound_count ;if not then next table entry
      LDA   Sound_count ;get
      CLC
      SBC   #Seq_sound-1      ;ck if > max assignment
      BCC   Snd_set           ;jump if <
      LDA   #Seq_sound-1      ;dont inc off end
      STA   Sound_count ;
      JMP   Snd_set           ;do it
Snd_reset:
      LDA   #00         ;reset to 1st entry of sequential
      STA   BIT_CT            ;temp store
      STA   Sound_count ;
Snd_set:
      LDA   #Global_time      ;get timer reset value
      STA   Sensor_timer      ;reset it
      LDA   BIT_CT            ;get current pointer to tables

Sndrand:

      STA   IN_DAT            ;save decision
      LDA   #Sound_ID   ;which ram location for learned word count
(offset)
      JSR   Start_learn ;go record training info
      LDA   IN_DAT            ;get back word to speak

      JSR   Decid_age   ;do age calculation for table entry
      LDX   TEMP0       ;age offset
      LDA   Sound_S1,X  ;get lo byte
      STA   Macro_Lo    ;save lo byte of Macro table entry
      INX               ;
      LDA   Sound_S1,X  ;get hi byte
      STA   Macro_Hi    ;save hi byte of Macro table entry
      JMP   Start_macro ;go set group/table pointer for motor & spch

;*****************************************************************
;*****************************************************************
;*****************************************************************
;*****************************************************************
;*****************************************************************
;
;É
;°Misc Subroutines
;È
; SENSOR TRAINING

; Training for each sensor is set up here and the decision if the
```

```
learned
; word should be played or not.
; Temp_ID hold the ram offset for the last sensor of the learned word.
; Temp_ID2 hold the ram offset for the current sensor of the learned
word.
; IN_DAT holds the current word the sensor chose, and will be loaded
with
; the learned word instead if the sensor count > the random number that
was
; just sampled, ie., force learned word to play.

; ****

; If the sensor timer is at 0 when entering here, then the LEARN_TEMP
; ram location is cleared, else the current learned word is loaded. If
; the learned word is 0 then all entries are cleared.


; When entering, check sensor timer and bail if 0. THen test if this is
; the back switch and if so then move the current sensor to previous
sensor
; ram and increment the counter.
; If this is not the back switch, then get previous sensor ram counter
and
; decrement it. THen move all current sensor information to previous and
; return to caller.

; Because of training difficulties, we now need two back touches to
; increment training counters. If only one occurs then the normal
decrement
; happens. This double back touch helps to prevent accidentally training
; with a new macro by hitting the back sw when it is not the macro you
; have been working with.

Start_learn:
      STA   Temp_ID2     ;sensor ram location of counter (current sensor)
      LDA   Temp_ID2     ;get current sensor ID
      CMP   #EEh         ;EF= this is the back switch (special)
      BNE   Not_BCK            ;jumpif not
      CPX   #00          ;ck if sensor timer timed out
      BNE   Learn_update       ;jump if is back switch and not timed out

Not_BCK:
      LDA   Temp_ID            ;get previous sensor ram offset
      CMP   #EEh         ;ck if last was back sw
      BEQ   Not_learned ;jump if no sensor prev

      LDX   Temp_ID            ;get previous sensor ram offset
      LDA   Tilt_learned,X     ;get learned word counter from ram
      CMP   Learn_temp  ;compare with last word
      BNE   Do_lrn2            ;bail out if different
      LDA   Tilt_lrn_cnt,X  ;prev sensor counter +offset to current
sensor
      CLC
      SBC   #Learn_chg  ;dec learned word counter since not back sw
      STA   Tilt_lrn_cnt,X     ;update
      BCS   Do_lrn2            ;jump if > #Learn_chg
      BPL   Do_lrn2            ;jump if not negative  (rolled over)
      LDA   #00
      STA   Tilt_lrn_cnt,X  ;set to zero, no roll over
```

```
Do_lrn2:
      LDX   Temp_ID             ;get sensor learn ram offset
      JSR   Random              ;get a number
      CLC
      LDA   Tilt_lrn_cnt,X      ;get count
      CMP   #FFh          ;check for max
      BEQ   Do_lrn2a      ;bypass random
      CLC
      SBC   Seed_1              ;random minus learned word counter
      BCC   Not_learned ;if less than random then bail out
Do_lrn2a:
      LDA   Tilt_learned,X      ;get learned word counter from ram
      AND   #0Fh          ;make sure never off end of table
      STA   Tilt_learned,X      ;also in ram
      STA   IN_DAT              ;force learned word for sensor
Not_learned:
      LDA   IN_DAT              ;get curent sensor word
      STA   Learn_temp    ;SAVE FOR NEXT PASS
      LDA   Temp_ID2      ;get current sensor
      STA   Temp_ID             ;save in previous sensor ram

      LDA   Stat_0              ;system
      AND   #EFh          ;"Train_Bk_prev" clear 2nd time thru flag
      STA   Stat_0              ;update

      RTS                 ;done-ola

Learn_update:
      LDA   Temp_ID             ;sensor ram location for last trigger
      CMP   #EEh          ;EE= this is the back switch (special)
      BEQ   Not_learned ;bail out if last trigger was also back sw
      CMP   #FFh          ;only happens on power up
      BEQ   Not_learned ;false call

      LDA   Stat_0              ;system
      AND   #Train_Bk_prev      ;is this the 1st or 2nd time thru
      BNE   Lrn_upd1      ;jump if 2nd back sw hit
      LDA   Stat_0              ;system
      ORA   #Train_Bk_prev      ;this is 1st time
      STA   Stat_0              ;update
      RTS                 ;my job is done here !

Lrn_upd1:
      LDA   Stat_0              ;system
      AND   #EFh          ;"Train_Bk_prev" clear 2nd time thru flag
      STA   Stat_0              ;update

      LDX   Temp_ID             ;sensor ram location for last trigger
      LDA   Tilt_learned,X      ;get learned word from ram
      CMP   Learn_temp    ;ck for training of same word
      BEQ   Lrn_upd2      ;jump if is
      LDA   Learn_temp    ;get new word trainer wants to use
      STA   Tilt_learned,X      ;update new word
      LDA   #00           ;reset to 0 for new word to train
      STA   Tilt_lrn_cnt,X      ;
      JMP   Not_learned ;done for now
Lrn_upd2:
      CLC
      LDA   Tilt_lrn_cnt,X      ;get learned word counter from ram
```

```
; on 1st cycle of new learn, we set counter 1/2 way ..... (chicken)

      BNE   Lrn_upd2a   ;jump if not 0
      LDA   #80h        ;1/2 way point
      STA   Tilt_lrn_cnt,X    ;update sensor counter
      JMP   Clear_learn ;go finish
Lrn_upd2a:
;---------- end 1st cycle preload

      ADC   #Learn_chg  ;add increment value
      BCS   Learn_overflw     ;jump if rolled over
      STA   Tilt_lrn_cnt,X    ;update sensor counter
      JMP   Clear_learn ;go finish
Learn_overflw:
      LDA   #FFh        ;set to max
      STA   Tilt_lrn_cnt,X    ;save it
Clear_learn:
      JMP   Do_lrn2           ;done



;*****************************************************************

;
; When IRQ gets turned off, and then restarted, we wait two complete
; cycle to insure the motor R/C pulses are back in sync.

Kick_IRQ:
      LDA   Stat_3            ;get system
      AND   #Nt_IRQdn   ;clear IRQ occured status
      STA   Stat_3            ;update system
      LDX   #03         ;loop counter
Kick2:
      LDA   Stat_3            ;system
      AND   #IRQ_dn           ;ck if IRQ occured
      BEQ   Kick2       ;wait till IRQ happens
      LDA   Stat_3            ;get system
      AND   #Nt_IRQdn   ;clear IRQ occured status
      STA   Stat_3            ;update system
      DEX               ;-1
      BNE   Kick2       ;loop til done
      RTS               ;is done

;*****************************************************************
;*****************************************************************

;EEPROM READ/WRITE

; Read & write subroutines

;*****************************************************************

Do_EE_write:

; EEPROM WRITE
```

```
; Enter with 'TEMP0' holding adrs of 0-63. Areg holds lo byte and
; Xreg holds hi byte. If carry is clear then it was succesfull, if
; carry is set the write failed.

; MODIFIED eeprom , load lo byte in temp1 and hi byte in temp2
;  and call EEWRIT2.

      LDA   #00          ;use DAC output to put TI in reset
      STA   DAC1         ;
      SEI                ;turn IRQ off

      LDA   #00          ;EEPROM adrs to write data to
      STA   Sgroup             ;save adrs
      LDA   #13          ;number of ram adrs to transfer (x/2)
      STA   Which_delay  ;save
      LDA   #00          ;Xreg offset
      STA   Which_motor  ;save

; Need one read cycle before a write to wake up EEPROM

      LDX   Which_motor  ;eeprom address to read from
      JSR   EEREAD             ;get data (wakes up eeprom)


Write_loop:

      LDA   Sgroup             ;get next EEPROM adrs
      STA   TEMP0        ;buffer
      LDX   Which_motor  ;ram source
      LDA   Age,X        ;lo byte (data byte #1)
      STA   TEMP1        ;save data bytes
      INC   Which_motor  ;
      INX
      LDA   Age,X        ;
      STA   TEMP2        ;hi byte (data byte #2)
      JSR   EEWRIT2            ;send em
;     BCS   EEfail             ;jump if bad

      INC   Sgroup             ;0-63 EEPROM adrs next
      INC   Sgroup             ;0-63 EEPROM adrs next (eeprom writes 2
bytes)
      INC   Which_motor  ;next adrs
      DEC   Which_delay  ;how many to send
      BNE   Write_loop   ;send some more

      RTS                ;done

;*************************************************************


; READ EEPROM HERE AND SETUP RAM

S_EEPROM_READ:

; Xreg is the adrs 0-63, system returns lo byte in Areg & hi byte in
Xreg.

;     on call: X = EEPROM data address (0-63)
;     on return: ACC = EEPROM data (low byte) (also in TEMP0)
;              X = EEPROM data (high byte)  (also in TEMP1)
```

```
        LDA   #00           ;use DAC output to put TI in reset
        STA   DAC1          ;
        SEI                 ;turn IRQ off

        LDX   #00           ;eeprom address to read from
        JSR   EEREAD              ;get data (one read to init system)


        LDA   #00           ;EEPROM adrs to read
        STA   Sgroup              ;save adrs
        LDA   #13           ;number of ram adrs to transfer (x/2)
        STA   Which_delay   ;save
        LDA   #00           ;Xreg offset to write ram data
        STA   Which_motor   ;save

Read_loop:

        LDX   Sgroup              ;EEPROM adrs
        JSR   EEREAD              ;get data

        LDX   Which_motor   ;ram destination
        LDA   TEMP0         ;get data
        STA   Age,X         ;lo byte (data byte #1)
        INC   Which_motor   ;
        INX
        INC   Sgroup              ;0-63 EEPROM adrs next
        LDA   TEMP1         ;get data
        STA   Age,X         ;lo byte (data byte #2)
        INC   Which_motor   ;next adrs
        INC   Sgroup              ;0-63 EEPROM adrs next
        DEC   Which_delay   ;how many to get
        BNE   Read_loop     ;send some more

        LDA   #00           ;clear rams used
        STA   Sgroup              ;
        STA   Which_motor   ;
        STA   Which_delay   ;

          CLI               ;Enable IRQ
        JSR   Kick_IRQ      ;wait for interrupt to restart
        JSR   TI_reset      ;go init TI  (uses 'Cycle_timer')
;*****************************************************************

;
;*****************************************************************************
******
; Begin Koball's code
;*****************************************************************************
******
;
; Enable or Disable EEPROM by setting/clearing CS
;     (CS = B.0)
;
;     on call: --
;     on return: --
;     stack usage: 0
;     RAM usage: B_IMG
;
```

```
;*************************************************************************
******
;
EEENA:
      LDA   Port_B_Image      ;get prev state of port B,
      ORA   #001H       ; turn on B.0
      JMP   EEE02       ;
;
EEDIS:
      LDA   Port_B_Image      ;get prev state of port B,
      AND   #0FEH       ; turn off B.0
;
EEE02:
      STA   Port_B            ;output to port
      STA   Port_B_Image      ; and save port image
      RTS               ;
;
;*************************************************************************
******
;
; Output data bit to EEPROM by placing data bit on
; EEPROM DI line and toggling EEPROM CLK line.
;
;     EEPROM DI = A.1
;     EEPROM CLK = A.0
;
;     on call: C = data bit to be output
;     on return: --
;     stack usage: 0
;     RAM usage: Port_A_image
;
;*************************************************************************
******
;
OUTBIT:
      BCS   OUTB02            ;branch if output bit = 1
;
      LDA   Port_A_image      ;get prev state of port A,
      AND   #0FDH       ; turn off A.1,
      JMP   OUTB04            ;
;
OUTB02:
      LDA   Port_A_image      ;get prev state of port A,
      ORA   #002H       ; turn on A.1,
;
OUTB04:
      STA   Port_A            ; output bit to port
      STA   Port_A_image      ; and save image
;
; toggle EEPROM clock
;
TOGCLK:
      LDA   Port_A_image      ;get prev state of A
      ORA   #001H       ;turn on A.0,
      STA   Port_A            ;output to port
      NOP               ;delay
      NOP               ;
      NOP               ;
      AND   #0FEH       ;turn off A.0
      STA   Port_A            ;output to port
```

```
      STA   Port_A_image      ;save image
      RTS                  ;
;
;*****************************************************************************
*****
;
; Read data 16-bit data word from EEPROM at specified address
;
;     on call: X = EEPROM data address (0-63)
;     on return: ACC = EEPROM data (low byte)
;                X = EEPROM data (high byte)
;     stack usage: 2
;     RAM usage: TEMP0
;
;*****************************************************************************
******
;
EEREAD:
      STX   TEMP0          ;store data addr
      JSR   EEENA          ;turn on CS
;
      SEC                  ;send start bit
      JSR   OUTBIT               ;
;
      SEC                  ;send READ opcode (10)
      JSR   OUTBIT               ;
      CLC                  ;
      JSR   OUTBIT               ;
;
      LDX   #6             ;init addr bit count
      ROL   TEMP0          ;align MS addr bit in bit 7
      ROL   TEMP0          ;
;
EERD02:
      ROL   TEMP0          ;shift address bit into carry
      JSR   OUTBIT               ;send it to EEPROM
      DEX                  ;bump bit counter
      BNE   EERD02               ; and repeat until done
;
      LDX   #16            ;init data bit count
      LDA   #0             ;
      STA   TEMP0          ;init data bit accumulators
      STA   TEMP1          ;
;
EERD04:
      JSR   TOGCLK               ;toggle clock for next bit
      LDA   #020H          ;test data bit (B.5) from EEPROM
      BIT   Port_B               ;
      BNE   EERD08               ;
;
      CLC                  ;EEPROM data bit = 0
      JMP   EERD10               ;
;
EERD08:
      SEC                  ;EEPROM data bit = 1
;
EERD10:
      ROL   TEMP0          ;rotate data bit into 16-bit
      ROL   TEMP1          ; accumulator
      DEX                  ;bump bit counter
```

```
      BNE   EERD04              ; and repeat until done
;
      JSR   EEDIS         ;turn off CS and return
      LDA   TEMP0         ;ret w/data byte in ACC
      LDX   TEMP1         ; and X regs
      RTS                 ;
;
;*************************************************************************
******
;
; Issue ERASE/WRITE ENABLE or DISABLE instruction to EEPROM
;     (instruction = 1001100000)
;
;     on call: --
;     on return: --
;     stack usage: 2
;     RAM usage: TEMP3
;
;*************************************************************************
******
;
EEWEN:
      LDA   #0FFH         ;set up enable inst
      JMP   EEWE02              ;
;
EEWDS:
      LDA   #000H         ;set up disable inst
;
EEWE02:
      STA   TEMP3         ;save instruction
      JSR   EEENA         ;turn on CS
;
      SEC                 ;send start bit
      JSR   OUTBIT              ;
;
      CLC                 ;send ENA/DIS opcode (00)
      JSR   OUTBIT              ;
      CLC                 ;
      JSR   OUTBIT              ;
;
      LDX   #6            ;init instr bit count
;
EEWE04:
      ROL   TEMP3         ;shift instruction bit into carry
      JSR   OUTBIT              ;send it to EEPROM
      DEX                 ;bump bit counter
      BNE   EEWE04              ; and repeat until done
      RTS                 ;
;
;*************************************************************************
******
;
; Write data byte to EEPROM at specified address
;
;     on call: TEMP0 = EEPROM data address (0-63)
;              ACC = data to be written (low byte)
;              X = data to be written (high byte)
;     on return: C = 0 on successful write cycle
;                C = 1 on write cycle time out
;     stack usage: 4
```

```
;     RAM usage: TEMP0, TEMP1, TEMP2
;
;*****************************************************************************
******
;
EEWRIT:
      STA   TEMP1         ;save data bytes
      STX   TEMP2         ;
EEWRIT2:
;
      JSR   EEWEN         ;send write enable inst to EEPROM
      JSR   EEDIS         ;set CS low
      JSR   EEENA         ; then high again
;
      SEC                 ;send start bit
      JSR   OUTBIT              ;
;
      CLC                 ;send WRITE opcode (01)
      JSR   OUTBIT              ;
      SEC                 ;
      JSR   OUTBIT              ;
;
      LDX   #6            ;init addr bit count
      ROL   TEMP0         ;align MS addr bit in bit 7
      ROL   TEMP0         ;
;
EEWR02:
      ROL   TEMP0         ;shift address bit into carry
      JSR   OUTBIT              ;send it to EEPROM
      DEX                 ;bump bit counter
      BNE   EEWR02              ; and repeat until done
;
      LDX   #16           ;init data bit count
;
EEWR06:
      ROL   TEMP1         ;shift data bit into carry
      ROL   TEMP2         ;
      JSR   OUTBIT              ;send it to EEPROM
      DEX                 ;bump bit counter
      BNE   EEWR06              ; and repeat until done
;
      JSR   EEDIS         ;cycle CS low
      JSR   EEENA         ; then high again
;
      LDA   #0            ;init write cycle
      STA   TEMP0         ; time out counter
      STA   TEMP1         ;
;
EEWR08:
      LDA   #020H         ;test READY/BUSY bit (B.5)
      BIT   Port_B              ;  from EEPROM
      BNE   EEWR10              ;wait for write cycle to finish
;
      DEC   TEMP0         ;write cycle time out counter
      BNE   EEWR08              ;
      DEC   TEMP1         ;
      BNE   EEWR08              ;
;
      JSR   EEWR10              ;time out, disable EEPROM and
      SEC                 ; set carry to signal error
```

```
      RTS                ;
;
EEWR10:
      JSR   EEWDS        ;send write disable inst to EEPROM
      JSR   EEDIS        ;set CS low
      CLC                ;clear carry to signal successful write
      RTS                ;
;
;*****************************************************************
;*****************************************************************
;

; Subroutine creates sensor table entry for the selected age.
; One table for each age.
; Enter with Acc holding the 1-16 table selection.
; Exit with Acc & Temp0 holding the offset 0-FF of the 1-4 age entry.

; Special condition where we have only two tables instead of 4
; (where each table is called based on age), if the "half_age" bit is
; set then ages 1 & 2 call table 1 and ages 3 & 4 call table 2.

Decid_age:
      STA   TEMP0        ;save 0-0f selection

      LDA   Stat_1             ;system
      AND   #Half_age    ;test if this is a special 2 table select
      BEQ   Decid_normal       ;jump if not
      LDA   Stat_1             ;
      AND   #Nt_half_age       ;clear req
      STA   Stat_1             ;update system

      LDA   Age          ;

  AND #03h  ;get rid of bit 7 (9th counter bit )

      CLC
      SBC   #01          ;actual age is 0-3, test if <2
      BCC   Dec_age1     ;choose age 1 ( actually 0 here)
      JMP   Spcl_age2    ;choose age 2 ( actually 1 here)
Decid_normal:

;;; mod TestR3a.... 25% of time chose age1 to add more furbish after
;;;                 he is age 4.

      JSR   Random             ;get a number
      CLC
      SBC   #Random_age ;below this level selects age 1
      BCS   Nospcl_age  ;jump if >
      LDA   #00         ;set age 1
      JMP   Do_age             ;go do it
;;; end mod

Nospcl_age:

      LDA   Age          ;get current
      AND   #03h         ;get rid of bit 7 (9th counter bit )
      CMP   #03          ;is it age 4
      BNE   Dec_age3     ;jump if not
      LDA   #96          ;point to 4th field
      JMP   Do_age             ;finish load from table
```

```
Dec_age3:
      CMP   #02          ;is it age 3
      BNE   Dec_age2     ;jump if not
      LDA   #64          ;point to 3rd field
      JMP   Do_age             ;finish load from table
Dec_age2:
      CMP   #01          ;is it age 2
      BNE   Dec_age1     ;jump if not
Spc1_age2:
      LDA   #32          ;point to 2nd field
      JMP   Do_age             ;finish load from table
Dec_age1:                ;age 1
      LDA   #00          ;point to 1st field
Do_age:
      STA   TEMP2        ;save age offset for speech
      CLC
      ROL   TEMP0        ;16 bit offset for speech
      LDA   TEMP2        ;which table entry
      ADC   TEMP0        ;create speech field ofsett pointer
      STA   TEMP0        ;save
      RTS
;
;*****************************************************************************
******
;*****************************************************************************
******
;
; Random/sequential decision control for all sensors.
;
; Enter with Acc holding the number of random selections for sensor.
; Enter with Xreg holding number of sequential selections
; It returns with Acc holding the random selection and the carry will
; be cleared for a sequential mode and set for a random mode.
; NOTE: if the caller has no random selections then carry will be
cleared.

Ran_seq:
      STA   TEMP1        ;save random max
      STX   TEMP5        ;save number of sequentials
      LdA   TEMP1        ;force cpu status ck
      BEQ   Seq_decisn   ;jump if no randoms
      DEC   TEMP1        ;make offset from 0
Ran_loop:
      JSR   Random             ;get n
      ROR   A            ;move hi nible to lo
      ROR   A
      ROR   A
      ROR   A
      AND   #0Fh         ;get lo nible
      STA   TEMP2        ;save
      CLC
      SBC   TEMP1        ;get max random number from sensor
      BCS   Ran_loop     ;loop until =< max value
      LDA   TEMP2        ;get new number
      CMP   Prev_random  ;ck if duplicate from last attempt
      BEQ   Ran_loop     ;loop if is
      STA   Prev_random  ;update for next pass
      STA   TEMP1        ;new

      LDA   TEMP5        ;ck if no sequentials
```

```
      BEQ   Ran_decisn  ;force random if none

      JSR   Random            ;get random/sequential decision

      CMP   IN_DAT            ;random/seqential split
;;;;;       CMP   #80h        ;>80=random else sequential

      BCC   Seq_decisn  ;jump if less

Ran_decisn:
      LDA   TEMP5       ;get number of sequential for this pass
      CLC
      ADC   TEMP1       ;add to random for correct table start point
      STA   TEMP1       ;update
      SEC               ;set carry to indicate random
      RTS               ;done (Acc holds answer)
Seq_decisn:
      CLC               ;clear carry to indicate sequential
      RTS               ;done (Acc holds answer)

;************************************************************
;************************************************************

; Random number generator,
; SEED_1 & SEED_2 are always saved through power down
; TEMP3 & TEMP4 are random temporary files.
; Acc returns with random number, Seed_1 also holds random number.

Random:
      LDA   Seed_1            ;
      STA   TEMP3       ;
      LDA   Seed_2            ;
      STA   TEMP4       ;
      CLC
      ROL   A
      ROL   Seed_1
      CLC
      ROL   A
      ROL   Seed_1
      CLC
      ADC   TEMP4
      STA   Seed_2
      LDA   #00
      ADC   Seed_1
      CLC
      ADC   TEMP3
      STA   Seed_1
      LDA   #00
      INC   Seed_2
      ADC   Seed_1
      STA   Seed_1
      RTS               ;return with random number in Acc & seed_1

;
;*************************************************************************
******
;*************************************************************************
******
```

```
Life:

; Each FEED trigger increments the HUNGRY counter by (EQU = FOOD).

;Hungry >80 (Need_food) + Sick >C0 (Really_sick) = normal sensor
;Hungry >80 (Need_food) + Sick <C0 (Really_sick) = random SICK/SENSOR
;Hungry <80 (Need_food) + Sick >C0 (Really_sick) = random HUNGRY/SENSOR
;Hungry <80 (Need_food) + Sick <C0 (Really_sick) = random
HUNGRY/SICK/SENSOR
;Hungry <60 (Sick_reff) + Sick <C0 (Really_sick) = random HUNGRY/SICK

;Hungry >60 then each sensor motion increments Sick
;Hungry <60 then each sensor motion decrements Sick

; When the system does a cold boot, we set HUNGRY & SICK to FFh......

; When returning from here, carry is set if sensor should execute
; normal routine, and cleared if sensor should do nothing.

;REFF only -----
;Hungry_counter
;Sick_counter

;Food       EQU   20h   ;amount to increase 'Hungry' for each feeding
;Need_food  EQU   80h   ;below this starts complaining about hunger
;Sick_reff  EQU   60h   ;below this starts complaining about sickness
;Really_sick      EQU   C0h   ;below this only complains about sickness

;Hungry_dec EQU   01    ;subtract X amount for each sensor trigger
;Sick_dec   EQU   01    ;subtract X amount for each sensor trigger
;Max_sick   EQU   see EQU


      LDA   Hungry_counter    ;current

;mod F-rels2 ;
;     CLC
      SEC
;end mod

      SBC   #Hungry_dec ;-X for each trigger
      BCS   frst_life   ;jump if not neg
      LDA   #00         ;reset
frst_life:
      STA   Hungry_counter    ;get count
      CLC
      SBC   #Sick_reff  ;ck if getting sick
      BCS   Sick_inc    ;jump if not sick
      LDA   Sick_counter      ;current

;mod F-rels2 ;
;     CLC
      SEC
;end mod

;mod testr3a

;     SBC   #Sick_dec   ;-X for each trigger
;     BCS   frst_sick   ;jump if not neg
```

```
;       LDA   #00           ;reset

        SBC   #Sick_dec     ;-X for each trigger
        STA   Sick_counter      ;
        BCC   Max_Sref      ;jump if neg
        CLC
        LDA   Sick_counter      ;get again
        SBC   #Max_sick     ;ck if at minimum allowed count
        BCS   frst_sick     ;jump if not at min
Max_Sref:
        LDA   #Max_sick     ;set to min

frst_sick:
        STA   Sick_counter      ;
        JMP   Hunger1           ;
;end mod testr3a

Sick_inc:
        INC   Sick_counter      ;+1 if is
        BNE   No_sick_inc ;jump if didn't roll over
        LDA   #FFh          ;if did the set to max
        STA   Sick_counter      ;
No_sick_inc:
;
Hunger1:
        LDA   Sick_counter      ;ck how sick
        CLC
        SBC   #Really_sick      ;decide if too sick to play
        BCC   Hunger2           ;jump if <

        LDA   Hungry_counter    ;check how hungry he is
        CLC
        SBC   #Need_food  ;ck if getting hungry
        BCC   Decd_Hung_norm    ;jump if is
Life_normal:
        SEC                 ;tell sensor to do normal routine
        RTS                 ;done

Hunger2:
        LDA   Hungry_counter    ;check how hungry he is
        CLC
        SBC   #Sick_reff  ;ck if very hungry and is sick
        BCC   Decd_Hung_sick    ;only speak hungry / sick

        LDA   Hungry_counter    ;check how hungry he is
        CLC
        SBC   #Need_food  ;ck if getting hungry
        BCS   Decd_Sick_norm    ;jump if is
;       JMP   Decd_Hung_sck_norm ;do hungry & sick speech

Decd_Hung_sck_norm:
        JSR   Random            ;need 3-way decision
        CLC
        SBC   #A0h          ;hi split
        BCS   Life_normal ;>A0 = normal sensor
        LDA   Seed_1            ;get again
        BMI   Say_sick      ;>80
        JMP   Say_hunger    ;<80
;
Decd_Hung_norm:
```

```
      JSR   Random            ;go get random 50/50 decision
      BMI   Life_normal ;
      JMP   Say_hunger  ;
;
Decd_Sick_norm:
      JSR   Random            ;go get random 50/50 decision
      BMI   Life_normal ;
      JMP   Say_sick    ;
;
Decd_Hung_sick:
      JSR   Random            ;go get random 50/50 decision
      BMI   Say_hunger  ;
      JMP   Say_sick    ;
;
Say_hunger:
      LDA   #Hunger_split     ;get random/sequential split
      STA   IN_DAT            ;save for random routine

      LDX   #Seq_hunger ;get how many sequential selections
      LDA   #Ran_hunger ;get number of random slections
      JSR   Ran_seq           ;go decide random/sequential
      BCS   Hunger_ran  ;Random mode when carry SET

      LDA   Sensor_timer      ;ck if timed out since last action
      BEQ   Hunger_reset      ;yep
      INC   Hungr_count ;if not then next table entry
      LDA   Hungr_count ;get
      CLC
      SBC   #Seq_hunger-1     ;ck if > assignment
      BCC   Hunger_side ;jump if <
      LDA   #Seq_hunger-1     ;dont inc off end
      STA   Hungr_count ;
      JMP   Hunger_side ;do it
Hunger_reset:
      LDA   #00         ;reset to 1st entry of sequential
      STA   Hungr_count ;
Hunger_side:
      LDA   #Global_time      ;get timer reset value
      STA   Sensor_timer      ;reset it
      LDA   Hungr_count ;get current pointer to tables

Hunger_ran:
      JSR   Decid_age   ;do age calculation for table entry
      LDX   TEMP0       ;age offset
      LDA   Hunger_S1,X ;get lo byte
      STA   Macro_Lo    ;save lo byte of Macro table entry
      INX               ;
      LDA   Hunger_S1,X ;get hi byte
      STA   Macro_Hi    ;save hi byte of Macro table entry
      JSR   Get_macro   ;go start motor/speech
      JSR   Notrdy            ;Do / get  status for speech and motor
      CLC               ;tells sensor to do nothing
      RTS
;
Say_sick:
      LDA   #Sick_split ;get random/sequential split
      STA   IN_DAT            ;save for random routine

      LDX   #Seq_sick   ;get how many sequential selections
      LDA   #Ran_sick   ;get number of random slections
```

```
      JSR   Ran_seq              o decide random/sequential
      BCS   Sick_ran    ;Ran_    mode when carry SET

      LDA   Sensor_timer       ;ck if timed out since last action
      BEQ   Sick_reset  ;yep
      INC   Sickr_count ;if not then next table entry
      LDA   Sickr_count ;get
      CLC
      SBC   #Seq_sick-1 ;ck if > assignment
      BCC   Sick_side   ;jump if <
      LDA   #Seq_sick-1 ;dont inc off end
      STA   Sickr_count ;
      JMP   Sick_side   ;do it
Sick_reset:
      LDA   #00         ;reset to 1st entry of sequential
      STA   Sickr_count ;
Sick_side:
      LDA   #Global_time       ;get timer reset value
      STA   Sensor_timer       ;reset it
      LDA   Sickr_count ;get current pointer to tables

Sick_ran:
      JSR   Decid_age   ;do age calculation for table entry
      LDX   TEMP0       ;age offset
      LDA   Sick_S1,X   ;get lo byte
      STA   Macro_Lo    ;save lo byte of Macro table entry
      INX               ;
      LDA   Sick_S1,X   ;get hi byte
      STA   Macro_Hi    ;save hi byte of Macro table entry
      JSR   Get_macro   ;go start motor/speech
      JSR   Nodrdy            ;Do / get  status for speech and motor
      CLC               ;tells sensor to do nothing
      RTS

;****   **********   ****************************************
;*****    ********   ****************************************
;



GoToSleep:

; save light sensor fail or sleep command in 'Seed_2' into EEPROM

      LDA   Stat_0              ;system
      AND   #Dark_sleep_prev ;
      BEQ   Nodrk_prev  ;jump if none
      LDA   #01         ;set flag that it was done
      STA   Seed_2              ;save in EEPROM
      JMP   Gs2         ;
Nodrk_prev:
      LDA   #00         ;set flag that it was clear
      STA   Seed_2              ;save in EEPROM
Gs2:

;*****************************************************************

; EEPROM WRITE
```

```
; Enter with 'TEMP0' holding adrs of 0-63. Areg holds lo byte and
; Xreg holds hi byte. If carry is clear then it was succesfull, if
; carry is set the write failed.

; MODIFIED eeprom , load lo byte in temp1 and _i byte in temp2
;  and call EEWRIT2.

      LDA    #00          ;use DAC output to put TI in reset
      STA    DAC1         ;
      SEI                 ;turn IRQ off

      LDA    #00          ;EEPROM adrs to write data  o
      STA    Sgroup             ;save adrs
      LDA    #13          ;number of ram adrs to transf   (x/2)
      STA    Which_delay ;save
      LDA    #00          ;Xreg offset
      STA    Which_motor ;save

; Need one read cycle before a write to wake up EEPROM

      LDX    Which_motor ;eeprom address to read from
      JSR    EEREAD             ;get data (wakes up eeprom)


IWrite_loop:

      LDA    Sgroup             ;get next EEPROM adrs
      STA    TEMP0        ;buffer
      LDX    Which_motor ;ram source
      LDA    Age,X        ;lo byte (data byte #1)
      STA    TEMP1        ;save data bytes
      INC    Which_motor ;
      INX
      LDA    Age,X        ;
      STA    TEMP2        ;hi byte (data byte #2)
      JSR    EEWRIT2            ;send em
;     BCS    EEfail             ;jump if bad

      INC    Sgroup             ;0-63 EEPROM adrs next
      INC    Sgroup             ;0-63 EEPROM adrs next (eeprom writes 2
bytes)
      INC    Which_motor ;next adrs
      DEC    Which_delay ;how many to send
      BNE    IWrite_loop ;send some more

;*************************************************************

GoToSleep_2:


      Include            Sleep.asm   ;


;
;Éííííííííííííííííííííííííííííííííííííííííííííííííííííííííííííííííííííí
ííííí
;ºInterrupt  Subroutines
;Èííííííííííííííííííííííííííííííííííííííííííííííííííííííííííííííííííííí
ííííí
```

```
;************  CAUTION  ************
; Any ram location written outside of IRQ can only be read in the IRQ,
; likewise if written in the IRQ, then can only be read outside the IRQ.
; THIS WILL PREVENT DATA CORRRUPTION.



NMI:
      RTI                ;Not used


IRQ:
       PHA               ;push acc on stack
      PHP                ;push cpu status on stack

;****** timer A = 166 uSEC *******

CkTimerA:
;        LDA     Interrupts    ;get who did it
;        AND     #20H             ;test for timerA
;        BNE     Do_ta         ;jump if is
;     JMP   Ck_timerB   ;
;
;Do_ta:

;***** timer B = 700 uSEC  ******

Ck_timerB:
       LDA     Interrupts     ;get status again
       AND     #10H           ;test for timer B
      BNE   Do_timeB     ;jump if request true
       JMP     Intt_false     ;bypass all if not

;       also changed TimerB relaod value from #10h to 00 in EQU

Do_timeB:

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

; RE-CALIBRATE SWITCH for motor position

; This counter must meet a threshold to decide if the
; calposition switch is really engaged.

      LDA   Port_C             ;get I/O
      AND   #Motor_cal  ;lo when limit hit
      BNE   No_cal_sw   ;no position switch found
      INC   Cal_switch_cnt     ;inc each time found low
      BNE   Cal_noroll1 ;jump if didnt roll over (stopped on switch)
      LDA   #31         ;max count
      STA   Cal_switch_cnt     ;
Cal_noroll:
      LDA   Cal_switch_cnt     ;
      CLC
      SBC   #30         ;ck if enough counts
      BCC   No_lim_stp  ;jump if not enough
      LDA   #Cal_pos_fwd       ; force value
      STA   Pot_timeL2  ;reset both
```

```
      JMP   No_lim_stp  ;done

No_cal_sw:
      LDA   #00         ;clear count if hi
      STA   Cal_switch_cnt    ;update


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


No_lim_stp:

      LDA   Wait_time   ;4 times thru loop = 2.9 mSec
      BNE   WTa         ;>0
      LDA   #04         ;counter reset
      STA   Wait_time   ;reload
      JMP   Timer_norm  ;
WTa:  DEC   Wait_time   ;
      JMP   TimerB_dn   ;bypass timers until done

Timer_norm:

;********** Below routines run at 2.9 mSec

      LDA   Mot_speed_cnt     ;ck for active
      BEQ   No_spd_m    ;jump if not
      DEC   Mot_speed_cnt     ;-1
No_spd_m:

      LDA   motorstoped ;motor drift timer
      BEQ   No_mstop    ;jump if done
      DEC   motorstoped ;-1
No_mstop:

      LDA   Motor_led_timer   ; Motor_led timer * 742 mSec
      BEQ   TimeB1            ;jump if done
      DEC   Motor_led_timer   ;-1
TimeB1:
      LDA   Cycle_timer ;2.9mSec timer * cycle reload
      BEQ   TimeB2            ;jump if done
      DEC   Cycle_timer ;-1
TimeB2:
;m    LDA   Motor_pulse ;2.9mSec timer * Motor_pulse
 m    BEQ   TimeB3            ;jump if done
;m    DEC   Motor_pulse ;-1
TimeB3:

      DEC   Mili_sec    ;-1 & allow rollover
      BNE   TimerB_dn   ;wait for rollover  (2.9mS * 256 = 742mSec)
      INC   Milisec_flag      ;tell task rtn to decrement timers

TimerB_dn:

;**********       We could test all interrupts here as needed
;Ck2Khz:
;Ck500hz:
;Ck60hz:

;********* Check motor position - IR slot in wheel sensor
```

```
; This version does two reads to eliminate noise and sets a done flag to
; prevent multiple counts. It also reads twice when no slot is present
to
; clear the done flag.

      LDA   Port_C            ;get I/O
      AND   #Pos_sen    ;ck position sensor
      BNE   Clr_pos           ;jump if no I      ger
      LDA   Port_C            ;get I/O
      AND   #Pos_sen    ; READ 2x to prevent noise trigger
      BNE   Clr_pos           ;jump if no IR trigger
      LDA   Slot_vote   ;get prev cycle
      BEQ   Pc_done           ;bail if prev counted
      LDA   #00         ;
      STA   Slot_vote   ;set ram to 0. (faster than setting a bit)
      JMP   Force_int   ;go count slot

Clr_pos:
      LDA   Port_C            ;get I/O
      AND   #Pos_sen    ; READ 2x to prevent noise trigger
      BEQ   Pc_done           ;not 2 equal reads so bypass this cycle
      STA   Slot_vote   ;set ram to 1. (faster than setting a bit)
      JMP   Pc_done           ;

;
;*******************************

ExtportC:
      JMP   Intt_false  ;this should be turned off
;        LDA     Interrupts    ;get status again
;        AND     #01H          ;test for port C bit 1 rising edge
;        BEQ     Pc_done       ;jump if not

Force_int:
;     LDA   Port_D_Image      ;system
;     AND   #Motor_led  ;ck if position I.R. led is on
;     BEQ   Pc_done           ;jump if not off

      LDA   Stat_2            ;get system
      AND   #Motor_fwd  ;if set then FWD else REV
      BEQ   Cnt_rev           ;jump if clr
      INC   Pot_timeL2  ;sensor counter
      CLC
      LDA   Pot_timeL2  ;current
      SBC   #207        ;ck for > 207
      BCC   Updt_cnt    ;jump if not
      LDA   #00         ;roll over
      STA   Pot_timeL2  ;
      JMP   Updt_cnt    ;
Cnt_rev:
      DEC   Pot_timeL2  ;-1
      CLC
      LDA   #208        ;max count
      C     Pot_timeL2  ;ck for negative ( >207 )
      BS    Updt_cnt    ;jump if not
Cnt_d
      LDA   #207        ;when neg roll over to max count
      STA   Pot_timeL2  ;
Updt_cnt:
      INC   Drift_counter     ;to be used for braking pulse
```

```
      LDA   Pot_timeL2  ;get current count
      STA   Pot_timeL   ;save in motor routine counter

; THis routine used to calculate motor speed based on battery voltage.
      LDA   Mot_speed_cnt     ;ck for active
      BEQ   Pc_done           ;jump if not
      INC   Mot_opto_cnt      ;

Pc_done:
      LDA   Motor_led_timer   ;ck if active (>0)
      BEQ   Mot_led_off ;jump if done
      LDA   Port_D_Image      ;system
      ORA   #Motor_led  ;turn LED on
      JMP   Mot_led_dn  ;
Mot_led_off:
      LDA   Port_D_Image      ;system
      AND   #Nt_Motor_led     ;turn LED off
Mot_led_dn:
      STA   Port_D_Image      ;update motor led

M_drft_F1:
      LDA   Drift_fwd   ;get delay value
      BEQ   M_drft_R1   ;jump if prev done
      LDA   Drift_fwd   ;get delay value
      CMP   #01         ;01=turn motors off
      BEQ   M_drft_F2   ;send it

      DEC   Drift_fwd   ;-1
;m32
      LDA   Port_D_Image      ;get system  (note lo is tranys off)
      AND   #3Fh        ;turn both motors off to prevent transistors
      STA   Port_D            ;on at same time
;m32
      LDA   Port_D_Image      ;get system
      ORA   #Motor_off  ;turn both motors off
      AND   #Motor_fwds ;move motor in fwd dir to stop motion
      JMP   Intt_motor_end
M_drft_F2:
      DEC   Drift_fwd   ;-1
      LDA   Port_D_Image      ;get system
      ORA   #Motor_off  ;turn both motors off
      JMP   Intt_motor_end

M_drft_R1:
      LDA   Drift_rev   ;get delay value
      BEQ   Intt_motor  ;jump if prev done
      LDA   Drift_rev   ;get delay value
      CMP   #01         ;01=turn motors off
      BEQ   M_drft_R2   ;send it
      DEC   Drift_rev   ;-1

;m32
      LDA   Port_D_Image      ;get system  (note lo is tranys off)
      AND   #3Fh        ;turn both motors off to prevent transistors
      STA   Port_D            ;on at same time
;m32

      LDA   Port_D_Image      ;get system
      ORA   #Motor_off  ;turn both motors off
      AND   #Motor_revs ;move motor in rev  dir to stop motion
```

```
      JMP   Intt_motor_end
M_drft_R2:
      DEC   Drift_rev   ;-1
      LDA   Port_D_Image      ;get system
      ORA   #Motor_off  ;turn both motors off
      JMP   Intt_motor_end

Intt_motor:
      LDA   Stat_3
      AND   #C0h        ;get motor command bits
      STA   Intt_Temp   ;save motor direction

;_____ Furby1? .. move motor pulse width to interrupt routine

      LDA   Motor_pulse1      ;get on time
      BEQ   Intmotor1   ;jump if 0
      DEC   Motor_pulse1            ;-1
      JMP   Intmotor_dn ;exit (dont change Intt_temp if on)
Intmotor1:
      LDA   Motor_pulse2      ;get off time
      BEQ   Intmotor2   ;got reset timer
      DEC   Motor_pulse2      ;-1
      LDA   #C0h        ;shut motor off
      STA   Intt_Temp   ;
      JMP   Intmotor_dn ;exit
Intmotor2:
      LDA   Mon_len           ;reset on time
      STA   Motor_pulse1            ;
      LDA   Moff_len    ;reset off time
      STA   Motor_pulse2      ;
Intmotor_dn

;----- end motor pulse width

      LDA   Port_D_Image      ;get system
      AND   #3Fh        ;clear motor direction bits
      CLC
      ADC   Intt_Temp   ;put in motor commands

Intt_motor_end:
      STA   Port_D_Image      ;update system

; st Tracker
        EOR     #%11000000      ;;Tracker add  invert motor drivers
; end Tracker

      STA   Port_D            ;output

Intt_done:                    ;ge ral  turn

      LDA   Stat_3            ;syst
      ORA   #IRQ_dn           ;flag  tem IRQ occured
      STA   Stat_3            ;upd at
Intt_false:
      LDA   #00H        ;clear all intts first
      STA   Interrupts  ;
        LDA #Intt_dflt  ;get default for interrupt reg
        STA Interrupts  ;set reg & clear intt flag

      PLP               ;recover CPU
```

```
        PLA             ;recover ACC
        RTI             ;reset interrupt

;*********************************************************************
;*********************************************************************
;*********************************************************************

; Communication protocal with the TI is:
;
;       FF is a no action command. (used as end of speech command)
;       FE sets the command data mode and the TI expects two
;       additional data bytes to complete the string. (3 TOTAL)
;       ALL OTHERS (0-FD) ARE CONSIDERED START OF A SPEECH WORD !
;       Command data structure is BYTE 1 + BYTE 2 + BYTE 3

; BYTE 1 is always  FE

; Command 1
;       BYTE 2 = FE is pitch table control;
;       BYTE 3 = bit 7 set = subtract value from current course value
;                      clr = add value to current course value
;                bit 6 set = select music pitch table
;                      clr = select normal speech pitch table
;                bit 0-5 value to change course value (no change = 0)
;
; Command 2
;       BYTE 2 = FD is Infrared transmit cmnd
;       BYTE 3 = Is the I.R. code to send  ( 0 - 0Fh only )
;
; Command 3
;       BYTE 2 = FC is the speech speed control
;       BYTE 3 = a value of 0 - 255 where  2Eh is normal speed.


; Enter subroutine with TEMP1 = command byte (1st)
;                       TEMP2 = data byte (2nd)

Xmit_TI:
        LDA     #FEh            ;tells TI command data to follow
        JSR     Spch_more       ;out data
        LDA     TEMP1           ;command code
        JSR     Spch_more       ;out data
        LDA     TEMP2           ;data to send
        JSR     Spch_more       ;out data
        RTS                     ;done


;*********************************************************** ***********
;*************************************************************************
;
; There is an entry for each bank of speech and only the words in that
; bank are in the list. THis is a subroutine call.

; The first time thru, we call SAY_x and as long as WORD_ACTIV or
SAY_ACTIV
; is set we call DO_NEXTSENT until saysent is done.

; There are 4 groups of 128 pointers in each group. This gives 512
```

```
saysents.

; 1. Enter with 'Which_word' holding 0-127 and 'Sgroup' for the 1 of 4
tables
;    which points to two byte adrs of a saysent. These two bytes are
;    loaded into Saysent_lo & Saysent_hi.

; 2. Data is shuffled to the TI according to the BUSY/REQ line
;

; Currently we have 167 speech words or sounds in ROM. Words 1 - 12
; are in bank 0 and 13 - 122 are in bank 1 & 123 - 167 in bank 2.

Say_0:
      LDA    Which_word  ;get offsett
      TAX                ;load offset to Xreg
             Sgroup             ;get current
      CMP    #03         ;is it table group 4
      BEQ    Dec_say4    ;jump if is
      CMP    #02         ;is it table group 3
      BEQ    Dec_say3    ;jump if is
      CMP    #01         ;is it table group 2
      BEQ    Dec_say2    ;jump if is
Dec_say1:                ;default group 1
      LDA    Spch_grp1,X ;get lo pointer
      STA    Saysent_lo  ;save
      INX                ;X+1
      LDA    Spch_grp1,X ;get hi pointer
      STA    Saysent_hi  ;save
      JMP    Dec_say5    ;go calc word
Dec_say2:
      LDA    Spch_grp2,X ;get lo pointer
      STA    Saysent_lo  ;save
      INX                ;X+1
      LDA    Spch_grp2,X ;get hi pointer
      STA    Saysent_hi  ;save
      JMP    Dec_say5    ;go calc word
Dec_say3:
      LDA    Spch_grp3,X ;get lo pointer
      STA    Saysent_lo  ;save
      INX                ;X+1
      LDA    Spch_grp3,X ;get hi pointer
      STA    Saysent_hi  ;save
      JMP    Dec_say5    ;go calc word
Dec_say4:
      LDA    Spch_grp4,X ;get lo pointer
      STA    Saysent_lo  ;save
      INX                ;X+1
      LDA    Spch_grp4,X ;get hi pointer
      STA    Saysent_hi  ;save
Dec_say5:
      LDX    #00         ;no offsett
      LDA    (Saysent_lo,X)    ;get data @ 16 bit adrs
      STA    TEMP2       ;save new speech speed
      LDA    #FCh        ;command for TI to except speed data
      STA    TEMP1       ;
      JSR    Xmit_TI           ;send it to TI
      INC    Saysent_lo  ;next saysent pointer
      BNE    Xney_say    ;jump if no roll over
      INC    Saysent_hi  ;+1
```

```
Xney_say:
      LDX   #00          ;no offsett
      LDA   (Saysent_lo,X)     ;get data @ 16 bit adrs
      CLC
      ADC   Rvoice             ;adjut to voice selected on power up
      STA   TEMP2        ;save new speech pitch
      LDA   #FEh         ;command for TI to except pitch data
      STA   TEMP1        ;


; The math routine  converts the value to 00 for 80 and
; if  80 then subtracts from 80 to get the minus version of 00
; ie, if number is 70 then TI gets sent 10 (-1

      LDA   TEMP2        ;get voice with offsett
      BMI   No_voice_chg       ;if >80 then no change
      LDA   #80h         ;remove offsett if <80
      CLC
      SBC   TEMP2        ;kill offset
      STA   TEMP2        ;update
No_voice_chg:
      JSR   Xmit_TI            ;send it to TI

Do_nextsent:
Frst_say:
      INC   Saysent_lo   ;next saysent pointer
      BNE   Scnd_say     ;jump if no roll over
      INC   Saysent_hi   ;+1
Scnd_say:
      LDX   #00          ;no offsett
      LDA   (Saysent_lo,X)     ;get data @ 16 bit adrs
      CMP   #FFH         ;check for end
      BEQ   Say_end            ;done
      LDA   (Saysent_lo,X)     ;get data @ 16 bit adrs
      STA   Which_word   ;
Wtest:
      CLC
      SBC   #12          ;ck if in bank 1
      BCS   Get_group1   ;jump if is

Get_group0:
      LDA   #00          ;set bank
        STA     Bank_ptr       ;Bank number
      CLC                ;clear carry
      LDA   Which_word   ;get word
      ROL   A            ;2's offsett
      TAX                ;load offset to Xreg
      LDA   Word_group0,X      ;get lo pointer
      STA   Word_lo            ;save
      INX                ;X+1
      LDA   Word_group0,X      ;get hi pointer
      STA   Word_hi            ;save
      JMP   Word_fini    ;go do it

Get_group1:
      LDA   Which_word   ;selection
      CLC
      SBC   #122         ;ck if in bank 2
      BCS   Get_group2   ;jump if is
```

```
      LDA   #01          ;set bank
        STA     Bank_ptr       ;Bank number
      CLC
      LDA   Which_word   ;get word
      SBC   #12          ;1st 12 in word_group0
      CLC
      ROL   A            ;2's offsett
      TAX                ;load offset to Xreg
      LDA   Word_group1,X      ;get lo pointer
      STA   Word_lo            ;save
      INX                ;X+1
      LDA   Word_group1,X      ;get hi pointer
      STA   Word_hi            ;save
      JMP   Word_fini

Get_group2:
      LDA   #02          ;set bank
        STA     Bank_ptr       ;Bank number
      CLC                ;clear carry
      LDA   Which_word   ;get word
      SBC   #122         ;1st 122 in word_group 0 & 1
      CLC
      ROL   A            ;2's offsett
      TAX                ;load offset to Xreg
      LDA   Word_group2,X      ;get lo pointer
      STA   Word_lo            ;save
      INX                ;X+1
      LDA   Word_group2,X      ;get hi pointer
      STA   Word_hi            ;save
Word_fini:
      LDA   Stat_1             ;get system
      ORA   #Say_activ   ;Set spch active after word pointer loaded
      ORA   #Word_activ ;Set status
      STA   Stat_1             ;update system
      JMP   Do_spch            ;go say it

Say_end:
      LDA   Stat_1             ;get system
      AND   #^Clr_spch   ;turn say_activ & Spch_activ off
      STA   Stat_1             ;save system
      RTS                ;done


; This is the re-entry point during speech for all words to be spoken

; ******** start of chg for 3 - #FFh xmits ti TI

Do_spch:
        LDA     Bank_ptr       ;Bank number
        STA     Bank           ;set it

        LDX     #00H
        LDA     (Word_lo,X)    ;Get the speech data
      CMP   #FFH         ;is it end of word
      BNE   Clr_word_end       ;jump if not end

      LDA   Stat_1             ;get system
      AND   #Word_term   ;was it prev set
      BEQ   Set_end            ;nope
```

```
; WAKE2
;     adds deep sleep mode. If 'Deep_sleep'=11h then tilt will not
;     wake us up. only invert.


; Power up reset decision for three types of startup:
; 1. Powerup with feed switch zeros ram & EEPROM, & calls 10-200-10 macro.
; 2. Power up from battery change wont clear EEPROM but calls 10-200-10 macro.
; 3. Wake up from Port_D clears ram and jumps directly to startup. No macro.



        SEI                     ;interrupts off
        LDX     #C0H            ;startup setting
        STX     Interrupts      ;disable Watch Dog
        LDX     #FFH           ;Reset stack pointer address $0FFH
        TXS

        LDX     #0
        LDA     Wake_up         ;Get the information from hardware to check
                                ;whether reset is from power up or wakeup
        STA     TEMP5
        STX     Wake_up         ;disable wakeup immediately, this action can
                                ;stop the reset occupied by another changed on
                                ;portD, so once the program can execute to
                                ;this line then chip will not be reset due to
                                ;port changed again


        AND     #%00000001      ;mask the rest of bit and just check the port
                                ;wake up information
        BEQ     Power_battery   ;jump to power up initial if not port D

; Need to debounce tilt and invert since they are very unstable

Ck_wakeup:
        LDA     #00             ;clear
        STA     TEMP1           ;
        STA     TEMP2           ;
        LDX     #FFh            ;loop counter
Dbnc_lp:
        LDA     Port_D
        AND     #01             ;ck tilt sw
        BEQ     Dbnc_lp2        ;jump if not tilt
        INC     TEMP1           ;switch counter
Dbnc_lp2:
        LDA     Port_D
        AND     #02             ;ck invert sw
        BEQ     Dbnc_lp3        ;jump if not invert
        INC     TEMP2           ;switch counter
Dbnc_lp3:
        DEX                     ;-1 loop count
        BNE     Dbnc_lp         ;loop

        LDA     Deep_sleep      ;decide if normal or deep sleep
        CMP     #11h            ;
        BEQ     Dbnc_lp4        ;if deep sleep then only test invert
        LDA     TEMP1           ;get tilt count
        BEQ     Dbnc_lp4        ;jump if 0
        CLC
        SBC     #               ;min count to insure not noise
        BCS     Power_Port_D    ;jump if > min
```

```
Dbnc_lp4:
        LDA     TEMP2           ;get invert count
        BEQ     Dbnc_lp5        ;jump if 0
        CLC
        SBC     #10             ;min count to insure not noise
        BCS     Power_Port_D    ;jump if > min
Dbnc_lp5:

;Verify that Port_D is no longer changing before going to sleep.
;If not, the CPU will lock up without setting the low power mode.
;Before we exit here when count is less than minimum count, we must
;be sure Port_D is not changing. If we jump to sleep routine when
;it is not stable, the sleep routine will wait forever to be stable
;which causes Furby appear to be locked up.

        LDA     #00             ;
        STA     TEMP1           ;counter
        LDA     Port_D          ;get current status
Test_sleep:
        CMP     Port_D          ;check if changed
        BNE     Ck_wakeup       ;start over if did
        DEC     TEMP1           ;-1 counter
        BNE     Test_sleep      ;loop
        JMP     GoToSleep_2     ;otherwise, just goto sleep again


Power_Port_D:
        LDA     #11h            ;signal port D wakeup
        STA     Warm_cold       ;
        JMP     L_PowerOnInitial

Power_battery:
        LDA     #05h            ;signal battery wakeup
        STA     Warm_cold       ;


L_PowerOnInitial:
        LDA     #00             ;clear deep sleep command
        STA     Deep_sleep      ;
```

```
;*****************************************************************

;; MODS :

; LIGHT3.asm
; Add test to light counter so that if the oscillator
; fails, the system will ignore light sensor and keep running.
;
; Light4
; When goes to complete dark and hits the 'Dark_sleep' level
; and stays there until the reff level updates, at that point
; we send Furby to sleep.
;
; Light5 (used in F-RELS2 )
; Change detection of light threshold to prevent false or continuous trigger.


;*****************************************************************

Bright          EQU     15      ;light sensor trigger > reff level (Hon)
Dim             EQU     15      ;Light sensor trigger < reff level (Hon)

Shift_reff      EQU     10      ;max count to set or clear prev done flag

Dark_sleep      EQU     B0h     ;when timer A hi =0f and timer A low
;                               is = to this EQU then send him to sleep


; The CDS light sensor generates a square wave of 500hz to 24khz based on
; light brightness. We can loop on the sense line and count time for the
; lo period to determine if light has changed and compare it to previous
; samples. This also determines going lighter or darker. We also set a timer
; so that if someone holds their hand over the sensor and we announce it,
; if the change isnt stable for 10 second, we ignore the change back to the
; previous state. If it does exis  for > 10 seconds, then it becomes the
; new sample to compare against on the next cycle.

; In order to announce light change, the system must have a consistent
; count > 'Shift_reff'.

; If a previous reff has been set then the 'Up_light' bit is set to
; look for counts greater than the reff. The system passes through the
; light routine 'Shift_reff' times. If it is consistently greater than
; the reff level, we get a speech trigger. If any single pass is less
; than the reff, the counter is set back to zero. This scenario also
; is obeyed when the trigger goes away, ie remove your hand, and the system
; counts down to zero.('Up_light' bit is cleared ) If during this time any
; trigger greater than reff occurs, the count is set back to max.
; This should prevent false triggers.



Get_light:      ;alt entry for diagnostics

; This uses timer A to get a count from the lo period of the clk

        SEI                     ;interrupts off
        LDA     #0C0H           ;disable timer, clock, ext ints,
        STA     Interrupts      ; & watchdog; select IRQ int.
        LDA     #000H           ;set timer A for timer mode
        STA     TMA_CON         ;
```

```
        LDA     #000H           ;re-start timer A
        STA     TMA_LSB         ;
        LDA     #000H           ;now CPUCLK; was #010H = CPUCLK/4 (Hon)
        STA     TMA_MSB         ;
Ck_lght2:
        LDA     TMA_MSB         ;test for dead light osc
        AND     #0Fh            ;get timer
        CMP     #0Fh            ;ck for > 0E
        BNE     Ck_lt2a         ;jump if not
        LDA     TMA_LSB         ;get lo byte
        CLC
        SBC     #E0h            ;ck for > (msb+lsb =0FE0)
        BCC     Ck_lt2a         ;jump if not
        JMP     Light_fail      ;bail out if >

Ck_lt2a:
        LDA     Port_D          ;get I/O
        AND     #Light_in       ;ck light clk is hi
        BEQ     Ck_lght2        ;wait for it to go hi

        LDA     #000H           ;re-start timer A
        STA     TMA_LSB         ;
        LDA     #000H           ;now CPUCLK; was #010H = CPUCLK/4 (Hon)
        STA     TMA_MSB         ;

Ck_lght3:
        LDA     TMA_MSB         ;test for dead light osc
        AND     #0Fh            ;get timer
        CMP     #0Fh            ;ck for > 0E
        BNE     Ck_lt3a         ;jump if not
        LDA     TMA_LSB         ;get lo byte
        CLC
        SBC     #E0h            ;ck for > (msb+lsb =0FE0)
        BCS     Light_fail      ;bail out if >

Ck_lt3a:
        LDA     Port_D          ;get I/O
        AND     #Light_in       ;ck light clk is lo
        BNE     Ck_lght3        ;wait for it to go lo to insure the clk edge
Ck_lght4:

        LDA     #000H           ;re-start timer A
        STA     TMA_LSB         ;
        LDA     #000H           ;now CPUCLK; was #010H = CPUCLK/4 (Hon)
        STA     TMA_MSB         ;

Ck_lght4a:
        LDA     Port_D          ;get I/O
        AND     #Light_in       ;ck if still lo
        BEQ     Ck_lght4a       ;loop till hi

; Timer A holds count for lo period of clk

Lght4cmp:
        LDA     TMA_MSB         ;get timer high byte
        AND     #00FH           ; mask out high nybble
        STA     TEMP2           ; and save it
        LDA     TMA_LSB         ;get timer low byte
        STA     TEMP1           ; and save it

        LDA     TMA_MSB         ;get timer A high byte again
```

```
        AND     #00FH           ; mask out high nybble
        CMP     TEMP2           ; and compare it with last reading
        BNE     Lght4cmp        ;loop until they're equal

; take 12 bit timer (2 bytes) and move to one byte and trash lo nible
; of low byte. End up with hi 8 bits out of 12.

        LDX     #04             ;loop counter
Light_byte:
        ROR     TEMP2           ;get lo bit into carry
        ROR     TEMP1           ;shuffle down and get carry from TEMP2
        DEX                     ;-1
        BNE     Light_byte      ;loop till done

Ck_lght4b:
        LDA     #Intt_dflt      ;Initialize timers, etc.
        STA     Interrupts      ;re-establish normal system
        CLI                     ;re-enable interrupt
        JSR     Kick_IRQ        ;wait for motor R/C to start working again
        CLC                     ;clear

;--- now have new count in 'TEMP1'

        LDA     Light_reff      ;get previous sample
        SBC     TEMP1           ;ck against current sample
        BCC     Ck_lght5        ;jump if negative
        CLC
        SBC     #Bright         ;ck if difference > reff
        BCS     Lght_brt        ;go do speech
        JMP     Kill_ltrf       ;bail out if not

Ck_lght5:
        CLC
        LDA     TEMP1           ;try the reverse subtraction
        SBC     Light_reff      ;prev
        BCC     Kill_ltrf       ;quit if negative
        CLC
        SBC     #Dim            ;is diff < reff
        BCC     Kill_ltrf       ;bail out if not
Lght_dim:
        LDA     Stat_3          ;system
        AND     #Nt_lght_stat   ;clear bit to indicate dark table
        STA     Stat_3          ;update system
        JMP     Do_lght         ;go fini
Lght_brt:
        LDA     Stat_3          ;system
        ORA     #Lght_stat      ;set bit to indicate light table
        STA     Stat_3          ;update system
        JMP     Do_lght ;

Light_fail:
        LDA     #FFh            ;force lo number so no conflicts
        STA     TEMP1
        LDA     #Intt_dflt      ;Initialize timers, etc.
        STA     Interrupts      ;re-establish normal system
        CLI                     ;re-enable interrupt
        JSR     Kick_IRQ        ;wait for motor R/C to start working again
        JMP     Kill_shift      ;ret with no req

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

Do_lght:
```

```
        LDA     Stat_1          ;system
        AND     #Up_light       ;ck if incrmnt mode
        BNE     Rst_shftup      ;jump if incrmnt mode
        LDA     #Shift_reff     ;set to max
        STA     Light_shift     ;
        JMP     No_lt_todo      ;
Rst_shftup:

        INC     Light_shift     ;+1
        LDA     Light_shift     ;get counter
        CLC
        SBC     #Shift_reff     ;ck if > max reff count
        BCC     No_lt_todo      ;jump if < max count
        LDA     #Shift_reff     ;reset to max
        STA     Light_shift     ;

        LDA     Stat_0          ;system
        AND     #Lt_prev_dn     ;check if previously done
        BNE     New_ltreff      ;jump if was

        LDA     Stat_0          ;system
        ORA     #Lt_prev_dn     ;set previously done
        STA     Stat_0          ;update

;       LDA     Stat_1          ;system
;       AND     #EFh            ;set sytem to shift decrmnt mode
;       STA     Stat_1          ;update

        LDA     #Light_reload   ;reset for next trigger
        STA     Light_timer     ;set it
        JMP     Do_ltchg        ;go announce it

New_ltreff:
        LDA     Light_timer     ;get current
        BNE     No_lt_todo      ;nothing to do
        LDA     TEMP1           ;get new count
        STA     Light_reff      ;update system

        LDA     Stat_1          ;system
        AND     #EFh            ;set sytem to shift decrmnt mode
        STA     Stat_1          ;update

        LDA     TEMP1           ;get current value
        CLC
        SBC     #Dark_sleep     ;ck if > sleep level
        BCS     Ck_drk          ;jump if >
        LDA     Stat_0          ;system
        AND     #7Fh            ;kill prev done
        STA     Stat_0          ;update
        JMP     Kill_ltrf       ;

Ck_drk:
        LDA     Stat_0          ;system
        AND     #Dark_sleep_prev ;ck if this was already done
        BNE     Kill_ltrf       ;jump if was

        LDA     Stat_0          ;system
        ORA     #REQ_dark_sleep ;set it
        ORA     #Dark_sleep_prev ;set also
        STA     Stat_0          ;update

Kill_ltrf:
```

```
;       LDA     Stat_0          ;system
;       AND     #Lt_prev_dn     ;check if previously done
;       BEQ     No_lt_todo      ;jump if clear
        LDA     Light_shift     ;get shift counter
        BEQ     Kill_shift      ;jump if went zero last time
        LDA     Stat_1          ;system
        AND     #Up_light       ;ck if incrmnt mode
        BEQ     Rst_shftdn      ;jump if decrmnt mode
        LDA     #00             ;set to min
        STA     Light_shift     ;
        JMP     No_lt_todo      ;
Rst_shftdn:
        DEC     Light_shift     ;-1
        JMP     No_lt_todo      ;done
Kill_shift:
        LDA     Stat_0          ;system
        AND     #FDh            ;clears Lt_prev_dn
        STA     Stat_0          ;update

        LDA     Stat_1          ;system
        ORA     #Up_light       ;prepare to incrmnt 'Light_shift'
        STA     Stat_1          ;update


No_lt_todo:
        SEC                     ;carry set indicates no light change
        RTS

;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;

;****** alert system to start speech

Do_ltchg:
        LDA     Stat_3          ;system
        AND     #Lght_stat      ;ck if went light or dark
        BNE     LT_ref_brt      ;went brighter if set
        LDA     Stat_4          ;get system
        ORA     #Do_lght_dim    ;set indicating change < reff level
        JMP     Ltref_egg       ;
LT_ref_brt:
        LDA     Stat_4          ;
        ORA     #Do_lght_brt    ;set indicating change > reff level
Ltref_egg:
        STA     Stat_4          ;update egg info
        CLC                     ;carry clear indicates light > reff
        RTS                     ;done
```

```
;ƒííííííííííííííííííííííííííííííííííííííííííííííííííííííííííííííííííííí
;º 'Diagnostics and calibration Routine                                º
;Èííííííííííííííííííííííííííííííííííííííííííííííííííííííííííííííííííííí
;
; Mods to the diagnostic routines :

; DIAG6 :
; Init memory,voice,name and write EEPROM before exiting.

; Diag7:
; EEprom memory test, reads and writes all locations.
; On power up if port D woke us, then bypass diagnostics.

;******************************************

; refer to self test mode documentation


;******************  START
;
; Diagnostic EQU's

Dwait_tilt      EQU     02      ;full test waiting for no tilt (step 1)

Diagnostic:

; All speech / motor calls use standard macro routines, except we
; force the macro directly. Be carefull to load the 'MACRO_LO' and
; 'MACRO_HI' bytes properly. We use a common subroutine to set the macro
; so 'MACRO_HI' is loaded only once in the subroutine. Be sure the macros
; are in the same 128 byte block. Initially chose adrs 400 (190) for these
; diags.

        LDA     Warm_cold       ;get startup condition
        CMP     #11h            ;ck for port D wakeup
        BEQ     No_Diag         ;jump if not

        LDX     #FFh            ;loop counter
DportD_tst:
        LDA     Port_D          ;get I/O
        AND     #03             ;ck for tilt and invert
        BNE     No_Diag         ;if either hi then bail out
        DEX                     ;-1
        BNE     DportD_tst      ;loop till done (ckg for Port D bounce)

        LDA     Port_C          ;get I/O
        AND     #0Ch            ;ck for front and back switches made
        BEQ     Diag1           ;if both not lo then bail out else start diag

No_Diag:
        JMP     Test_byp        ;no diagnostic request

Diag1:                          ;Start test
        ;; force voice to normal condition while diag is active
        LDA     #9              ;;Tracker add for constant diag
        STA     Rvoice          ;;Tracker add
        LDA     #0              ;hi beep for start of test
        JSR     Diag_macro      ;go send motor/speech

;wait for front & back to clear

        LDA     Port_C          ;get I/O
```

```
        AND     #0Ch            ;get keys
        CMP     #0Ch            ;must be both hi
        BNE     Diag1           ;wait till are
New_top:
        LDA     #03             ;set delay for switch bounce
        JSR     Half_delay      ;x *  delay
;
Diag2a:         ;press front key & go to EEPROM test
        LDA     Port_C          ;get I/O
        AND     #Touch_frnt     ;wait for switch
        BNE     Diag2b          ;go ck if next test is requesting

        LDA     #01             ;hi beep for start of test
        JSR     Diag_macro      ;go send motor/speech

Diag2a1:
        LDA     Port_C
        AND     #Touch_frnt
        BEQ     Diag2a1

; EEPROM WRITE

; init ram as 1,2,3,4,5,..... to 26

        LDA     #01H            ; data for fill
        LDX     #Age            ; start at ram location

RAMset:
        STA     00,X            ; base 00, offset x
        CLC
        ADC     #01             ;inc Acc
        INX                     ; next ram location
        CPX     #Age+26         ; check for end
        BNE     RAMset          ; branch, not finished
                                ; fill done

        JSR     Do_EE_write     ;write the EEPROM
        JSR     S_EEPROM_READ   ;read data to ram

        LDA     #00             ;clear
        STA     Task_ptr        ;
        LDX     #Age            ; start at ram location
RAMtest:
        LDA     00,X            ; base 00, offset x
        CLC                     ;
        ADC     Task_ptr        ;running CRC
        STA     Task_ptr        ;running total
        INX                     ; next ram location
        CPX     #Age+26         ; check for end
        BNE     RAMtest         ; branch, not finished
        LDA     Task_ptr        ;get result
        CMP     #5Fh            ;matching CRC (actual total is 15Fh )
        BNE     EEfail          ;jump if bad

EEpass:
        LDA     #02             ;beep to signal good test
        STA     Feed_count      ;use as temp storage
        JMP     EEdone          ;send sounds
EEfail:
        LDA     #03             ;beep indicates failure
        STA     Feed_count      ;temp storage
EEdone:
```

```
        CLI                     ;enable IRQ
        JSR     Kick_IRQ        ;wait for timer      e-sync
        JSR     TI_reset        ;clear TI from

        LDA     Feed_count      ;get lo byte of macro to call
        JSR     Diag_macro      ;go send motor/speech

Diag2b:         ; Speaker tone / I.R. xmit
        LDA     Port_C          ;get I/O
        AND     #Touch_bck      ;wait for switch
        BNE     Diag2c          ;go check if next test is requesting

        LDA     #1              ;hi beep for start of test
        JSR     Diag_macro      ;go send motor/speech
Diag2blp:
        LDA     Port_C
        AND     #Touch_bck
        BEQ     Diag2blp

Diag2b1:
        LDA     #04             ;send long tone (1k sinewave)
        JSR     Diag_macro      ;go send motor/speech

        LDA     Port_C          ;
        AND     #Touch_bck      ;mask for back switch
        BNE     Diag2b1         ;loop until back switch pressed

Xmit_lp:
        LDA     #01             ;beep
        JSR     Diag_macro      ;go send motor/speech

;        LDA     Port_C          ;
;        AND     #Touch_bck      ;mask for back switch
;        BNE     Xmit_lp         ;loop until back switch pressed

        LDA     #05h            ;send '5' to I.R. xmiter
        STA     TEMP2           ;
        LDA     #FDh            ;send command I.R. to TI
        STA     TEMP1           ;
        JSR     Xmit_TI         ;send it

dumb:   LDA     Port_C          ;get I/O
        AND     #Touch_bck      ;wait for switch
        BNE     dumb            ;wait for back to be pressed

dumber: LDA     Port_C          ;get I/O
        AND     #Touch_frnt     ;ck switch
        BEQ     Next_1
        JMP     Xmit_lp

Next_1: LDA     #2              ;hi beep for start of test
        JSR     Diag_macro      ;go send motor/speech
        LDA     Port_C          ;get I/O
        AND     #0Ch            ;ck for front and back switches made
        BEQ     Next_1          ;if both not lo then bail out else start diag
        JMP     New_top


; Full test starts here
Diag2c: LDA     Port_D          ;get I/O
        AND     #Ball_invert    ;wait for switch
        BNE     Diag2d          ;onward if key pressed
```

```
        JMP     Diag2a          ;loop back to top if none

Diag2d:
        LDA     #01             ;hi beep for start of test
        JSR     Diag_macro      ;go send motor/speech

; FULL TEST MODE

DiagF1:         ;wait for no tilt to start full diag
        LDA     #Dwait_tilt     ;set delay to be sure no tilts
        STA     TEMP1           ;
DiagF1a:
        LDA     Port_D
        AND     #3
        BNE     DiagF1
        DEC     TEMP1
        BNE     DiagF1a

        LDA     #2              ;pass beep
        JSR     Diag_macro      ;go send motor/speech
;
DiagF2:         ;test tilt 45 deg

        LDA     Port_C
        AND     #00001100b
        CMP     #0CH
        BEQ     DiagF22
        LDA     #3              ; fail beep
        JSR     Diag_macro      ;

DiagF22:
        LDA     Port_D
        AND     #2
        BEQ     DiagF23

        LDA     #3              ; fail beep
        JSR     Diag_macro      ;

DiagF23:
        LDA     Port_D          ;get I/O
        AND     #Ball_side      ;ck for tilt switch (hi = tilted)
        BEQ     DiagF2          ;wait for tilt

        LDA     Port_D          ;get I/O
        AND     #Ball_invert    ;ck if invert sw made
        BNE     DiagF2a         ;jump to error if so

        LDA     Port_C          ;get I/O
        AND     #0Ch            ;get front & back
        CMP     #0Ch            ;must be hi else error
        BEQ     DiagF2b         ;if hi then pass

DiagF2a:
        LDA     #3              ;fail beep
        JSR     Diag_macro      ;go send motor/speech
        JMP     DiagF2          ;loop till no error

DiagF2b:
        LDA     #2              ;pass beep
        JSR     Diag_macro      ;go send motor/speech

DiagF2c:        ;wait for no tilt before continuing
```

```
        LDA     Port_C
        AND     #Touch_bck
        BEQ     DiagF3b

        LDA     Port_D          ;get I/O
        AND     #Ball_side      ;ck for tilt switch (hi = tilted)
        BNE     DiagF2c         ;wait for no tilt

;DANGER
;       LDA     Port_C          ;get I/O
;       AND     #Touch_frnt     ;ck switch
;       BEQ     DiagF3          ; no other switch can be made here else error
;       JMP     DiagF23         ; allow multiple checks

DiagF3:         ;test back switch
;       LDA     Port_C          ;get I/O
;       AND     #Touch_bck      ;wait for switch
;       BEQ     release         ;loop if hi (touch is not pressed)
        JMP     DiagF23

release:
        LDA     Port_C          ;get I/O
        AND     #Touch_frnt     ;ck switch
        BEQ     DiagF3a         ;no other switch can be made here else error

        LDA     Port_D          ;get I/O
        AND     #03             ;ck for tilt and invert
        BEQ     DiagF3b         ;if either hi then error else continue

DiagF3a:
        LDA     #3              ;fail beep
        JSR     Diag_macro      ;go send motor/speech
        JMP     DiagF3          ;loop till no error

DiagF3b:
        LDA     #2              ;pass beep
        JSR     Diag_macro      ;go send motor/speech
;
DiagF4:
        LDA     Port_C          ;get I/O   wait for front to clear
        AND     #Touch_frnt     ;ck switch
        BEQ     DiagF4          ;if pressed then wait for release

;  Send motor forward until front switch pressed

        LDA     Stat_2          ;get system
        ORA     #Motor_fwd      ;set = motor fwd (inc)
        ORA     #Motor_actv     ;set motor in motion
        STA     Stat_2          ;update system
        LDA     Stat_3          ;get current status
        ORA     #Motor_off      ;turn both motors off
        AND     #Motor_fwds     ;move motor in fwd dir
        STA     Stat_3          ;update

DiagF4a1:
        LDA     Port_C          ;get I/O   wait for front
        AND     #Touch_frnt     ;ck switch
        BEQ     DiagF4a2        ;got it
        JMP     DiagF4a1        ;loop till found

;  Send motor reverse until front switch pressed
```

```
DiagF4a2:
        LDA     Port_C          ;get I/O   wait for front to clear
        AND     #Touch_frnt     ;ck switch
        BEQ     DiagF4a2        ;if pressed then wait for release

        LDA     Stat_2          ;get system
        AND     #Motor_rev      ;clear fwd flag
        ORA     #Motor_actv     ;set motor in motion
        STA     Stat_2          ;update system
        LDA     Stat_3          ;get current status
        ORA     #Motor_off      ;turn both motors off
        AND     #Motor_revs     ;move motor in rev dir
        STA     Stat_3

DiagF4a3:
        LDA     Port_C          ;get I/O   wait for front
        AND     #Touch_frnt     ;ck switch
        BEQ     DiagF4a4        ;got it
        JMP     DiagF4a3        ;loop till found

;  Send motor end to end and stop on cal sw, else error

DiagF4a4:
        LDA     Stat_3          ;get current status
        ORA     #Motor_off      ;turn both motors off
        STA     Stat_3          ;update
        LDA     Stat_2          ;get system
        AND     #Motor_inactv   ;clear activ flag
        STA     Stat_2          ;update system

        LDA     #5             ;start motor test
        JSR     Diag_macro      ;go
        LDA     #33             ;set delay for motor to stop
        JSR     Half_delay      ;x * half sec delay
;       LDA     Port_C          ;get I/O
;       AND     #Motor_cal      ;lo when hit
        BNE     DiagF4b         ;no position switch found
        LDA     #2              ;pass beep
        JSR     Diag_macro      ;go send it
        JMP     DiagF5          ;done
DiagF4b:
        LDA     #3              ;fail beep
        JSR     Diag_macro      ;go send it
;
DiagF5:         ;send motor to mouth open for feed sw test
        LDA     Port_C          ;get I/O
        AND     #Touch_frnt     ;wait for switch
        BNE     DiagF5          ;loop

        LDA     #6              ;feed position
        JSR     Diag_macro      ;send it
;
DiagF6:
; ck for feed sw, all other sw = error
; Remember to test invert before setting feed sw test, else conflict.

        LDA     #00
        STA     DAC2            ;clear feed sw enable
        LDA     Port_C          ;get I/O
        AND     #0Ch            ;ck for front and back switches made
        CMP     #0Ch            ;ck both are clear
        BNE     DiagF6a         ;wait till are
```

```
        LDA     Port_D          ;get I/O
        AND     #03             ;ck for tilt and invert
        BNE     DiagF6a         ;if either hi then wait till clear
        JMP     DiagF6b         ;jump when all clear
DiagF6a:
        LDA     #3              ;fail beep when any other switch made
        JSR     Diag_macro      ;send it
        JMP     DiagF6          ;loop
DiagF6b:

;mod diag6 ; inc random number seeds until feed switch down

        INC     Seed_1          ;create random based on switches
        LDA     TMA_LSB         ;get timer A also (should be unknown)
        STA     Seed_2          ;save it

;end mod

        LDA     #FFh            ;turn DAC2 on to enable feed switch
        STA     DAC2            ;out
        LDA     Port_D          ;get I/O
        AND     #Ball_invert    ;ck if feed switch closed
        BEQ     DiagF6          ;loop until switch closed
        LDA     #00
        STA     DAC2            ;clear feed sw enable
        LDA     #7              ;pass beep
        JSR     Diag_macro      ;go send motor/speech
;
DiagF7:         ;Light sensor test

;mod to compensate for new light sense routine

;       LDA     #00             ;clear light timer to force new reff cycle
;       STA     Light_timer     ;set it
;       LDA     Stat_3          ;get system
;       ORA     #Lt_reff        ;make this pass a new light reff
;       STA     Stat_3          ;update
;       JSR     Get_light       ;go get light level, establish 1st level

        LDA     Stat_4          ;
        AND     #Nt_do_lt_dim   ;clear indicating change > reff level
        STA     Stat_4          ;update system

        JSR     Get_light       ;go get light level sample
        LDA     TEMP1           ;get new count
        STA     Light_reff      ;update system



DiagF7a:
        JSR     Get_light       ;go get again and test for lower level
        LDA     Stat_4          ;get system
        AND     #Do_lght_dim    ;check if went dimmer
        BEQ     DiagF7a         ;loop if no change
        LDA     #8              ;pass beep and motor motion
        JSR     Diag_macro      ;send it
;
DiagF8:         ;Sound sensor test
        LDA     #00             ;clear sound timer to force new reff cycle
        STA     Sound_timer     ;set
        LDA     Stat_1          ;get system again
        ORA     #Snd_reff       ;make this pass a new sound reff
```

```
        STA     Stat_1          ;update
        JSR     Get_sound       ;go get light level, establish 1st level
        LDA     Stat_4          ;
        AND     #Nt_do_snd      ;clear indicating change > reff level
        STA     Stat_4          ;update system
DiagF8a:
        JSR     Get_sound       ;go get again and test for lower level
        LDA     Stat_4          ;get system
        AND     #Do_snd         ;check if went louder
        BEQ     DiagF8a         ;loop if no change
        LDA     #9              ;pass beep and motor motion
        JSR     Diag_macro      ;send it
;
DiagF9:         ;wait for I.R. data received

        LDX     #10             ;;Tracker change, orginal is 100

DiagF9a1:
        LDA     #1
        JSR     Half_delay
        DEX
        BNE     DiagF9a1

        JSR     D_IR_test       ;go ck for data
        BCC     DiagF9          ;;loop until data receive
        CMP     #A5H            ;is it the expected data
        BNE     DiagF9a         ;jump if wrong data
        LDA     #1              ;pass beep and motor motion
        JSR     Diag_macro      ;send it
        JMP     DiagF10         ;done

DiagF9a:
        LDA     #3              ;fail beep and motor motion
        JSR     Diag_macro      ;send it

DiagF10:        ;all tests complete, send to sleep mode
        LDA     #10             ;
        JSR     Half_delay      ;

        LDA     #10             ;put furby in sleep postion
        JSR     Diag_macro      ;send it

; Clear RAM to 00H

; we dont clear Seed_1 or Seed_2 since they are randomized at startup.

; -------------------------------------------------------------------------
        LDA     #00H            ; data for fill
        LDX     #D7h            ; start at ram location

Clear:
        STA     00,X            ; base 00, offset x
        DEX                     ; next ram location
        CPX     #7FH            ; check for end
        BNE     Clear           ; branch, not finished

;*************************************************************************


; Random voice selection here

        LDA     #80h            ;get random/sequential split
```

```
        STA     IN_DAT          ;save for random routine

        LDX     #00             ;make sure only gives random
        LDA     #10h            ;get number of random selections
        JSR     Ran_seq         ;go get random selection

        TAX
        LDA     Voice_table,X   ;get new voice
        STA     Rvoice          ;set new voice pitch

;*****************************************************************************

; On power up or reset, Furby must go select a new name ,,, ahw how cute.

        JSR     Random          ;
        AND     #1Fh            ;get 32 possible
        STA     Name            ;set new name pointer

;*****************************************************************************

        LDA     #FFh            ;insure not hungry or sick
        STA     Hungry_counter  ;max not hungry
        STA     Sick_counter    ;Max not sick

; Clear training on all sensors

        LDA     #00

        STA     Temp_ID
        STA     Temp_ID2

        STA     Tilt_learned
        STA     Tilt_lrn_cnt

        STA     Feed_learned
        STA     Feed_lrn_cnt

        STA     Light_learned
        STA     Light_lrn_cnt

        STA     Dark_learned
        STA     Dark_lrn_cnt

        STA     Front_learned
        STA     Front_lrn_cnt

        STA     Sound_learned
        STA     Sound_lrn_cnt

        STA     Wake_learned
        STA     Wake_lrn_cnt

        STA     Invert_learned
        STA     Invert_lrn_cnt

        JMP     GoToSleep       ;write ee memory YO !
```

```
; Furby27.inc ;; change twinkle egg song to one pass in macro


; Lowered voice+10,voice+9 to voice+8
; Wayne's mods:
; Furby5b.inc = add voice selection table
;
; Dave's
; added feed (mouth open)
; 170,171,173,174,175,182,183,190,191,194
; mod for ir
; NOW 24 NAMES
;
;TABLES          MACRO    |SAY
;-------------------------------------
;FRONT           2-64     |1-61
;FORTUNE         65-83    |62-78
;o-too-mah       84
;HANGOUT         85-101   |79-106
;delay           102      |107
;FEED            103-145  |108-123
;WAKE            146-169  |124-156
;HUNGER          170-201  |157-168
;INVERT          202-238  |169-192
;BACK            239-275  |193-236
;SICK            276-292  |237-250
;LIGHT           293-307  |251-265
;DARK            308-331  |266-289
;SOUND           332-351  |290-309
;TILT            352-392  |310-350
;IR              393-429  |351-390
;FURBY SAYS      430-434  |50 TICKLE,196 PET,71 SOUND,391 LIGHT,198 PURR
;                435,436  |392 NO LIGHT, 393 LOUD SOUND
;                437,438  |115,116          ; hide and seek sounds
;                95,96,97 |98,99,100        ; hide and seek reuse
;                439      |                 ; furby says win sound
;Diagnostic      440-450  |400-410
;                451,452  | 117,118         ; hide and seek sounds
;Names           453      | 399,395,110     ; me koko      (more)
;                454      | 399,395,396     ; me meme      (very)
;                455      | 399,395,112     ; me e-day     (good)
;                456      | 399,395,397     ; me do-moh    (please)
;                457      | 399,395,114     ; me toh-dye   (done)
;                458      | 399,395,117     ; me boo       (no)
;                459      | 399,395,398     ; me toh-loo   (like)
;                460      | 399,395,120     ; me ay-tay    (hungry)
;                         | 399             ; delay 1.3 seconds
;                461      | 399,395,131     ; me way-loh   (sleep)
;                462      | 399,395,143     ; me u-tye     (up)
;                463      | 399,395,145     ; me ay-loh    (light)
;                464      | 399,395,152     ; me kah       (me)
;                465      | 399,395,166     ; me dah       (big)
;                466      | 399,395,175     ; me boh-bay   (worry)
;                467      | 399,395,177     ; me nah-bah   (down)
; NEW EASTER EGGS
;                468      ; DODLE DO, ME LOVE YOU
;                469      ; SING A SONG
;                470      ; BURB ATTACK
;                471                      ; furby says win sound
;                472      |46              ; furby says lose sound
```

```
;                473      |53,123              ; me done (leaving any game)
;                474      |394                 ; LISTEN ME
;                475      |411                 ; HIDE ME (hide and seek)
;                         |412                 ; aaah,aaah,aaah feed dmh
;
; MORE NAMES
;                476      | 399,395,186        ; me loo-loo  (joke)
;                477      | 399,395,194        ; me ah-may   (pet)
;                478      | 399,395,201        ; me noo-loo  (happy)
;                479      | 399,395,208        ; me may-may  (love)
;                480      | 399,395,224        ; me may-lah  (hug)
;                481      | 399,395,228        ; me dah-noh-lah (big dance)
;                482      | 399,395,398,152    ; me toh-loo-ka  (like me)
;                483      | 399,395,152,166    ; me ka-da       (me big)
;                484      | 399,395,224,152    ; me may-lah-ka  (hug me)
;
;not used        476-511  |413-510

; TRAP FOLLOW MACROS FOR NAME
;
; SENSOR
; HANGOUT 97
; WAKE-UP 149
; BACK 248
; LIGHT BRIGHT 305
; IR 393,404,414,421
;
; GAMES
; FORTUNE 69,77
; HIDE AND SEEK 475
; FURBY SAYS 474
;
; end trap macros for name
;
; reused ; reused ; reused ; reused ; reused
;                 72,380  |                  ; furby says win sounds
;                15       |15                ; LAUGH
;                         |395               ; me (for use with names)
;DANCE           407,416  |367,376           ; reused for dance easter egg
;
;not used                 |396-399
;*****************************************************************************

; Sensor tables
; Each sensor has 4 speech/motor tables based on age 1-4, of 16 entries
each.
; These tables are 16 bit entries, the user enters as a decimal 1-511
; **** '00' is illegal ****
; This number calls the MACRO tables to get specific speech and motor
; tables. MACRO tables chain together multiple motor and speech tables.
; The first 8 entries of speech is random selections and
; the second 8 entries is sequential.


;
;
; one of three voice pitch selections, randomly load table and
; table is randomly called on power up to select a new voice.
; THis gives a number added to voice 3 to create which voice will be
```

```
used.

Voice_table:
        DB      S_voice1
        DB      S_voice2
        DB      S_voice3
        DB      S_voice1
        DB      S_voice2
        DB      S_voice3
        DB      S_voice1
        DB      S_voice2
        DB      S_voice3
        DB      S_voice1
        DB      S_voice2
        DB      S_voice3
        DB      S_voice1
        DB      S_voice2
        DB      S_voice3
        DB      S_voice1

;Ball tilt sensor table
;DO TILT

Tilt_S1:  DW    352             ; #1  AGE 1
          DW    353             ; #2  AGE 1
          DW    354             ; #3  AGE 1
          DW    352             ; #4  AGE 1
          DW    355             ; #5  AGE 1
          DW    356             ; #6  AGE 1
          DW    357             ; #7  AGE 1
          DW    358             ; #8  AGE 1
          DW    359             ; #9  AGE 1
          DW    360             ; #10 AGE 1
          DW    361             ; #11 AGE 1
          DW    362             ; #12 AGE 1
          DW    363             ; #13 AGE 1
          DW    352             ; #14 AGE 1
          DW    364             ; #15 AGE 1
          DW    365             ; #16 AGE 1

Tilt_S2:  DW    366             ; #1  AGE 2
          DW    367             ; #2  AGE 2
          DW    366             ; #3  AGE 2
          DW    355             ; #4  AGE 2
          DW    368             ; #5  AGE 2
          DW    357             ; #6  AGE 2
          DW    369             ; #7  AGE 2
          DW    370             ; #8  AGE 2
          DW    359             ; #9  AGE 2
          DW    360             ; #10 AGE 2
          DW    371             ; #11 AGE 2
          DW    372             ; #12 AGE 2
          DW    373             ; #13 AGE 2
          DW    374             ; #14 AGE 2
          DW    365             ; #15 AGE 2
          DW    375             ; #16 AGE 2

Tilt_S3:  DW    366             ; #1  AGE 3
          DW    355             ; #2  AGE 3
```

```
            DW      376             ; #3  AGE 3
            DW      377             ; #4  AGE 3
            DW      378             ; #5  AGE 3
            DW      379             ; #6  AGE 3
            DW      380             ; #7  AGE 3
            DW      381             ; #8  AGE 3
            DW      382             ; #9  AGE 3
            DW      383             ; #10 AGE 3
            DW      384             ; #11 AGE 3
            DW      385             ; #12 AGE 3
            DW      365             ; #13 AGE 3
            DW      375             ; #14 AGE 3
            DW      363             ; #15 AGE 3
            DW      386             ; #16 AGE 3

Tilt_S4:    DW      366             ; #1  AGE 4
            DW      355             ; #2  AGE 4
            DW      387             ; #3  AGE 4
            DW      377             ; #4  AGE 4
            DW      388             ; #5  AGE 4
            DW      389             ; #6  AGE 4
            DW      380             ; #7  AGE 4
            DW      381             ; #8  AGE 4
            DW      382             ; #9  AGE 4
            DW      383             ; #10 AGE 4
            DW      390             ; #11 AGE 4
            DW      385             ; #12 AGE 4
            DW      391             ; #13 AGE 4
            DW      375             ; #14 AGE 4
            DW      384             ; #15 AGE 4
            DW      392             ; #16 AGE 4
;
;
Sick_S1:
        DW      276             ; #1  AGE 1
        DW      280             ; #2  AGE 1
        DW      283             ; #3  AGE 1
        DW      286             ; #4  AGE 1
        DW      288             ; #5  AGE 1
        DW      288             ; #6  AGE 1
        DW      289             ; #7  AGE 1
        DW      290             ; #8  AGE 1
        DW      291             ; #9  AGE 1
        DW      292             ; #10 AGE 1
        DW      288             ; #11 AGE 1
        DW      288             ; #12 AGE 1
        DW      289             ; #13 AGE 1
        DW      290             ; #14 AGE 1
        DW      291             ; #15 AGE 1
        DW      292             ; #16 AGE 1

Sick_S2:
        DW      277             ; #1  AGE 2
        DW      280             ; #2  AGE 2
        DW      284             ; #3  AGE 2
        DW      286             ; #4  AGE 2
        DW      288             ; #5  AGE 2
        DW      288             ; #6  AGE 2
        DW      289             ; #7  AGE 2
        DW      290             ; #8  AGE 2
```

```
        DW      291             ; #9  AGE 2
        DW      292             ; #10 AGE 2
        DW      288             ; #11 AGE 2
        DW      288             ; #12 AGE 2
        DW      289             ; #13 AGE 2
        DW      290             ; #14 AGE 2
        DW      291             ; #15 AGE 2
        DW      292             ; #16 AGE 2

Sick_S3:
        DW      276             ; #1  AGE 3
        DW      281             ; #2  AGE 3
        DW      285             ; #3  AGE 3
        DW      287             ; #4  AGE 3
        DW      288             ; #5  AGE 3
        DW      288             ; #6  AGE 3
        DW      289             ; #7  AGE 3
        DW      290             ; #8  AGE 3
        DW      291             ; #9  AGE 3
        DW      292             ; #10 AGE 3
        DW      288             ; #11 AGE 3
        DW      288             ; #12 AGE 3
        DW      289             ; #13 AGE 3
        DW      290             ; #14 AGE 3
        DW      291             ; #15 AGE 3
        DW      292             ; #16 AGE 3

Sick_S4:
        DW      279             ; #1  AGE 4
        DW      282             ; #2  AGE 4
        DW      285             ; #3  AGE 4
        DW      287             ; #4  AGE 4
        DW      288             ; #5  AGE 4
        DW      288             ; #6  AGE 4
        DW      289             ; #7  AGE 4
        DW      290             ; #8  AGE 4
        DW      291             ; #9  AGE 4
        DW      292             ; #10 AGE 4
        DW      288             ; #11 AGE 4
        DW      288             ; #12 AGE 4
        DW      289             ; #13 AGE 4
        DW      290             ; #14 AGE 4
        DW      291             ; #15 AGE 4
        DW      292             ; #16 AGE 4

; SWITCH FOR DO SOUND) js
Sound_S1: DW    332             ; #1  AGE 1
          DW    333             ; #2  AGE 1
          DW    334             ; #3  AGE 1
          DW    335             ; #4  AGE 1
          DW    336             ; #5  AGE 1
          DW    337             ; #6  AGE 1
          DW    338             ; #7  AGE 1
          DW    339             ; #8  AGE 1
          DW    332             ; #9  AGE 1
          DW    333             ; #10 AGE 1
          DW    334             ; #11 AGE 1
```

```
            DW      335             ; #12 AGE 1
            DW      336             ; #13 AGE 1
            DW      337             ; #14 AGE 1
            DW      338             ; #15 AGE 1
            DW      339             ; #16 AGE 1

Sound_S2:   DW      332             ; #1  AGE 2
            DW      333             ; #2  AGE 2
            DW      340             ; #3  AGE 2
            DW      341             ; #4  AGE 2
            DW      342             ; #5  AGE 2
            DW      337             ; #6  AGE 2
            DW      343             ; #7  AGE 2
            DW      344             ; #8  AGE 2
            DW      332             ; #9  AGE 2
            DW      333             ; #10 AGE 2
            DW      340             ; #11 AGE 2
            DW      341             ; #12 AGE 2
            DW      342             ; #13 AGE 2
            DW      337             ; #14 AGE 2
            DW      343             ; #15 AGE 2
            DW      344             ; #16 AGE 2

Sound_S3:   DW      332             ; #1  AGE 3
            DW      333             ; #2  AGE 3
            DW      345             ; #3  AGE 3
            DW      346             ; #4  AGE 3
            DW      342             ; #5  AGE 3
            DW      337             ; #6  AGE 3
            DW      347             ; #7  AGE 3
            DW      339             ; #8  AGE 3
            DW      332             ; #9  AGE 3
            DW      333             ; #10 AGE 3
            DW      345             ; #11 AGE 3
            DW      346             ; #12 AGE 3
            DW      342             ; #13 AGE 3
            DW      337             ; #14 AGE 3
            DW      347             ; #15 AGE 3
            DW      339             ; #16 AGE 3

Sound_S4:   DW      348             ; #1  AGE 4
            DW      333             ; #2  AGE 4
            DW      349             ; #3  AGE 4
            DW      346             ; #4  AGE 4
            DW      342             ; #5  AGE 4
            DW      350             ; #6  AGE 4
            DW      347             ; #7  AGE 4
            DW      351             ; #8  AGE 4
            DW      348             ; #9  AGE 4
            DW      333             ; #10 AGE 4
            DW      349             ; #11 AGE 4
            DW      346             ; #12 AGE 4
            DW      342             ; #13 AGE 4
            DW      350             ; #14 AGE 4
            DW      347             ; #15 AGE 4
            DW      351             ; #16 AGE 4

;*************************************************************
```

```
; DO HUNGER
;
;
Hunger_S1:
          DW    170             ; #1  AGE 1
          DW    173             ; #2  AGE 1
          DW    176             ; #3  AGE 1
          DW    180             ; #4  AGE 1
          DW    182             ; #5  AGE 1
          DW    173             ; #6  AGE 1
          DW    185             ; #7  AGE 1
          DW    189             ; #8  AGE 1
          DW    193             ; #9  AGE 1
          DW    194             ; #10 AGE 1
          DW    173             ; #11 AGE 1
          DW    195             ; #12 AGE 1
          DW    189             ; #13 AGE 1
          DW    193             ; #14 AGE 1
          DW    194             ; #15 AGE 1
          DW    199             ; #16 AGE 1

Hunger_S2:
          DW    171             ; #1  AGE 2
          DW    174             ; #2  AGE 2
          DW    177             ; #3  AGE 2
          DW    181             ; #4  AGE 2
          DW    183             ; #5  AGE 2
          DW    174             ; #6  AGE 2
          DW    186             ; #7  AGE 2
          DW    190             ; #8  AGE 2
          DW    193             ; #9  AGE 2
          DW    194             ; #10 AGE 2
          DW    174             ; #11 AGE 2
          DW    196             ; #12 AGE 2
          DW    190             ; #13 AGE 2
          DW    193             ; #14 AGE 2
          DW    194             ; #15 AGE 2
          DW    200             ; #16 AGE 2

Hunger_S3:
          DW    172             ; #1  AGE 3
          DW    174             ; #2  AGE 3
          DW    178             ; #3  AGE 3
          DW    181             ; #4  AGE 3
          DW    184             ; #5  AGE 3
          DW    175             ; #6  AGE 3
          DW    187             ; #7  AGE 3
          DW    191             ; #8  AGE 3
          DW    193             ; #9  AGE 3
          DW    173             ; #10 AGE 3
          DW    175             ; #11 AGE 3
          DW    197             ; #12 AGE 3
          DW    191             ; #13 AGE 3
          DW    193             ; #14 AGE 3
          DW    173             ; #15 AGE 3
          DW    200             ; #16 AGE 3

Hunger_S4:
          DW    171             ; #1  AGE 4
          DW    175             ; #2  AGE 4
```

```
            DW      179             ; #3  AGE 4
            DW      181             ; #4  AGE 4
            DW      184             ; #5  AGE 4
            DW      175             ; #6  AGE 4
            DW      188             ; #7  AGE 4
            DW      192             ; #8  AGE 4
            DW      194             ; #9  AGE 4
            DW      193             ; #10 AGE 4
            DW      174             ; #11 AGE 4
            DW      198             ; #12 AGE 4
            DW      192             ; #13 AGE 4
            DW      193             ; #14 AGE 4
            DW      194             ; #15 AGE 4
            DW      201             ; #16 AGE 4

; Fortune teller game
;GEORGE 07/04/98          MACRO 65-83,SAY 62-78
Fortyes_S1:

            DW      065             ; #1  AGE 1
            DW      066             ; #2  AGE 1
            DW      067             ; #3  AGE 1
            DW      068             ; #4  AGE 1
            DW      069             ; #5  AGE 1
            DW      070             ; #6  AGE 1
            DW      071             ; #7  AGE 1
            DW      072             ; #8  AGE 1
            DW      073             ; #9  AGE 1
            DW      074             ; #10 AGE 1
            DW      075             ; #11 AGE 1
            DW      076             ; #12 AGE 1
            DW      077             ; #13 AGE 1
            DW      078             ; #14 AGE 1
            DW      079             ; #15 AGE 1
            DW      080             ; #16 AGE 1

Fortyes_S2:

            DW      081             ; #1  AGE 2
            DW      082             ; #2  AGE 2
            DW      083             ; #3  AGE 2
            DW      065             ; #4  AGE 2
            DW      066             ; #5  AGE 2
            DW      067             ; #6  AGE 2
            DW      068             ; #7  AGE 2
            DW      069             ; #8  AGE 2
            DW      070             ; #9  AGE 2
            DW      071             ; #10 AGE 2
            DW      072             ; #11 AGE 2
            DW      073             ; #12 AGE 2
            DW      074             ; #13 AGE 2
            DW      075             ; #14 AGE 2
            DW      076             ; #15 AGE 2
            DW      077             ; #16 AGE 2

;END FORTUNE
;END GEORGE 07/04/98
;
;
;
```

```
;touch front sensor table
;GEORGE 07/03/98 MACRO 2-64,SAY 1-61
Tfrnt_S1: DW    002             ; #1  AGE 1
          DW    003             ; #2  AGE 1
          DW    004             ; #3  AGE 1
          DW    005             ; #4  AGE 1
          DW    006             ; #5  AGE 1
          DW    007             ; #6  AGE 1
          DW    008             ; #7  AGE 1
          DW    009             ; #8  AGE 1
          DW    010             ; #9  AGE 1
          DW    011             ; #10 AGE 1
          DW    012             ; #11 AGE 1
          DW    013             ; #12 AGE 1
          DW    014             ; #13 AGE 1
          DW    015             ; #14 AGE 1
          DW    016             ; #15 AGE 1
          DW    017             ; #16 AGE 1

Tfrnt_S2: DW    018             ; #1  AGE 2
          DW    019             ; #2  AGE 2
          DW    020             ; #3  AGE 2
          DW    021             ; #4  AGE 2
          DW    022             ; #5  AGE 2
          DW    023             ; #6  AGE 2
          DW    024             ; #7  AGE 2
          DW    025             ; #8  AGE 2
          DW    026             ; #9  AGE 2
          DW    027             ; #10 AGE 2
          DW    028             ; #11 AGE 2
          DW    029             ; #12 AGE 2
          DW    030             ; #13 AGE 2
          DW    031             ; #14 AGE 2
          DW    032             ; #15 AGE 2
          DW    033             ; #16 AGE 2

Tfrnt_S3: DW    034             ; #1  AGE 3
          DW    035             ; #2  AGE 3
          DW    036             ; #3  AGE 3
          DW    037             ; #4  AGE 3
          DW    038             ; #5  AGE 3
          DW    039             ; #6  AGE 3
          DW    040             ; #7  AGE 3
          DW    041             ;025 ; #8  AGE 3
          DW    002             ; #9  AGE 3
          DW    042             ; #10 AGE 3
          DW    043             ; #11 AGE 3
          DW    044             ; #12 AGE 3
          DW    045             ; #13 AGE 3
          DW    046             ; #14 AGE 3
          DW    047             ; #15 AGE 3
          DW    048             ; #16 AGE 3

Tfrnt_S4: DW    049             ; #1  AGE 4
          DW    050             ; #2  AGE 4
          DW    051             ; #3  AGE 4
          DW    052             ; #4  AGE 4
          DW    053             ; #5  AGE 4
          DW    054             ; #6  AGE 4
          DW    055             ; #7  AGE 4
```

```
          DW     056              ; #8  AGE 4
          DW     057              ; #9  AGE 4
          DW     058              ; #10 AGE 4
          DW     059              ; #11 AGE 4
          DW     060              ; #12 AGE 4
          DW     061              ; #13 AGE 4
          DW     062              ; #14 AGE 4
          DW     063              ; #15 AGE 4
          DW     064              ; #16 AGE 4
;END GEORGE 07/03/98

;
;
;feed sense table
; DO FEED (Do 1INVERT)
;GEORGE 07/05/98
Feed_S1:

          DW     117              ; #1  AGE 1
          DW     103              ; #2  AGE 1
          DW     104              ; #3  AGE 1
          DW     105              ; #4  AGE 1
          DW     106              ; #5  AGE 1
          DW     107              ; #6  AGE 1
          DW     108              ; #7  AGE 1
          DW     109              ; #8  AGE 1
          DW     110              ; #9  AGE 1
          DW     111              ; #10 AGE 1
          DW     112              ; #11 AGE 1
          DW     113              ; #12 AGE 1
          DW     114              ; #13 AGE 1
          DW     111              ; #14 AGE 1
          DW     115              ; #15 AGE 1
          DW     116              ; #16 AGE 1

Feed_S2:

          DW     118              ; #1  AGE 2
          DW     119              ; #2  AGE 2
          DW     120              ; #3  AGE 2
          DW     121              ; #4  AGE 2
          DW     122              ; #5  AGE 2
          DW     123              ; #6  AGE 2
          DW     124              ; #7  AGE 2
          DW     125              ; #8  AGE 2
          DW     126              ; #9  AGE 2
          DW     127              ; #10 AGE 2
          DW     128              ; #11 AGE 2
          DW     113              ; #12 AGE 2
          DW     114              ; #13 AGE 2
          DW     111              ; #14 AGE 2
          DW     129              ; #15 AGE 2
          DW     116              ; #16 AGE 2
Feed_S3:

         DW    118               ; #1  AGE 3
          DW     130              ; #2  AGE 3
          DW     131              ; #3  AGE 3
          DW     132              ; #4  AGE 3
          DW     122              ; #5  AGE 3
```

```
            DW      107             ; #6  AGE 3
            DW      133             ; #7  AGE 3
            DW      134             ; #8  AGE 3
            DW      110             ; #9  AGE 3
            DW      111             ; #10 AGE 3
            DW      135             ; #11 AGE 3
            DW      113             ; #12 AGE 3
            DW      114             ; #13 AGE 3
            DW      111             ; #14 AGE 3
            DW      135             ; #15 AGE 3
            DW      116             ; #16 AGE 3
Feed_S4:

            DW      145             ; #1  AGE 4
            DW      136             ; #2  AGE 4
            DW      137             ; #3  AGE 4
            DW      138             ; #4  AGE 4
            DW      139             ; #5  AGE 4
            DW      140             ; #6  AGE 4
            DW      141             ; #7  AGE 4
            DW      142             ; #8  AGE 4
            DW      110             ; #9  AGE 4
            DW      111             ; #10 AGE 4
            DW      143             ; #11 AGE 4
            DW      113             ; #12 AGE 4
            DW      114             ; #13 AGE
            DW      111             ; #14 AGE 4
            DW      144             ; #15 AGE 4
            DW      116             ; #16 AGE 4
;END GEORGE 07/05/98
;touch front sensor table
; DO WAKE  ;DONE SG
Wakeup_S1:
            DW      146             ; #1  AGE 1
            DW      149             ; #2  AGE 1
            DW      150             ; #3  AGE 1
            DW      154             ; #4  AGE 1
            DW      158             ; #5  AGE 1
            DW      159             ; #6  AGE 1
            DW      163             ; #7  AGE 1
            DW      166             ; #8  AGE 1
            DW      146             ; #9  AGE 1
            DW      149             ; #10 AGE 1
            DW      150             ; #11 AGE 1
            DW      154             ; #12 AGE 1
            DW      158             ; #13 AGE 1
            DW      159             ; #14 AGE 1
            DW      163             ; #15 AGE 1
            DW      166             ; #16 AGE 1

Wakeup_S2: DW    147                ; #1  AGE 2
            DW      149             ; #2  AGE 2
            DW      151             ; #3  AGE 2
            DW      155             ; #4  AGE 2
            DW      158             ; #5  AGE 2
            DW      160             ; #6  AGE 2
            DW      163             ; #7  AGE 2
            DW      167             ; #8  AGE 2
            DW      147             ; #9  AGE 2
            DW      149             ; #10 AGE 2
```

```
            DW     151                ; #11 AGE 2
            DW     155                ; #12 AGE 2
            DW     158                ; #13 AGE 2
            DW     160                ; #14 AGE 2
            DW     163                ; #15 AGE 2
            DW     167                ; #16 AGE 2

Wakeup_S3: DW      148                ; #1  AGE 3
            DW     149                ; #2  AGE 3
            DW     152                ; #3  AGE 3
            DW     156                ; #4  AGE 3
            DW     158                ; #5  AGE 3
            DW     161                ; #6  AGE 3
            DW     164                ; #7  AGE 3
            DW     168                ; #8  AGE 3
            DW     148                ; #9  AGE 3
            DW     149                ; #10 AGE 3
            DW     152                ; #11 AGE 3
            DW     156                ; #12 AGE 3
            DW     158                ; #13 AGE 3
            DW     161                ; #14 AGE 3
            DW     164                ; #15 AGE 3
            DW     168                ; #16 AGE 3

Wakeup_S4: DW      148                ; #1  AGE 4
            DW     149                ; #2  AGE 4
            DW     153                ; #3  AGE 4
            DW     157                ; #4  AGE 4
            DW     158                ; #5  AGE 4
            DW     162                ; #6  AGE 4
            DW     165                ; #7  AGE 4
            DW     169                ; #8  AGE 4
            DW     148                ; #9  AGE 4
            DW     149                ; #10 AGE 4
            DW     153                ; #11 AGE 4
            DW     157                ; #12 AGE 4
            DW     158                ; #13 AGE 4
            DW     162                ; #14 AGE 4
            DW     165                ; #15 AGE 4
            DW     169                ; #16 AGE 4

;Ball tilt sensor table
; DO TILT (HANGING OUT)
;START HANGOUT MACRO 85-101,SAY 79-106
;GEORGE 07/04/98
;
;
; DO HANGOUT
; DO BORED
Bored_S1:          ;bored time out
        DW     085                ; #1  AGE 1
          DW     086              ; #2  AGE 1
          DW     087              ; #3  AGE 1
          DW     088              ; #4  AGE 1
          DW     089              ; #5  AGE 1
          DW     090              ; #6  AGE 1
          DW     091              ; #7  AGE 1 ;sleep
          DW     092              ; #8  AGE 1
          DW     093              ; #9  AGE 1 ;dobedo
          DW     094              ; #10 AGE 1 ;yawn
```

```
            DW      095             ; #11 AGE 1 ;sigh
            DW      095             ; #12 AGE 1 ;sigh
            DW      096             ; #13 AGE 1 ;haa
            DW      091             ; #14 AGE 1 ;sleep was 96 dmh
            DW      097             ; #15 AGE 1 ;heey
            DW      098             ; #16 AGE 1 ;phone

Bored_S2:   DW      085             ; #1  AGE 2
            DW      086             ; #2  AGE 2
            DW      087             ; #3  AGE 2
            DW      088             ; #4  AGE 2
            DW      089             ; #5  AGE 2
            DW      099             ; #6  AGE 2
            DW      091             ; #7  AGE 2
            DW      092             ; #8  AGE 2
            DW      093             ; #9  AGE 2
            DW      094             ; #10 AGE 2
            DW      095             ; #11 AGE 2
            DW      095             ; #12 AGE 2
            DW      096             ; #13 AGE 2
            DW      091             ; #14 AGE 1 ;sleep was 96 dmh
            DW      097             ; #15 AGE 2
            DW      098             ; #16 AGE 2

Bored_S3:   DW      085             ; #1  AGE 3
            DW      086             ; #2  AGE 3
            DW      087             ; #3  AGE 3
            DW      088             ; #4  AGE 3
            DW      101             ; #5  AGE 3
            DW      100             ; #6  AGE 3
            DW      091             ; #7  AGE 3
            DW      092             ; #8  AGE 3
            DW      093             ; #9  AGE 3
            DW      094             ; #10 AGE 3
            DW      095             ; #11 AGE 3
            DW      095             ; #12 AGE 3
            DW      096             ; #13 AGE 3
            DW      091             ; #14 AGE 1 ;sleep was 96 dmh
            DW      097             ; #15 AGE 3
            DW      098             ; #16 AGE 3

Bored_S4:   DW      085             ; #1  AGE 4
            DW      086             ; #2  AGE 4
            DW      087             ; #3  AGE 4
            DW      088             ; #4  AGE 4
            DW      101             ; #5  AGE 4
            DW      100             ; #6  AGE 4
            DW      091             ; #7  AGE 4
            DW      092             ; #8  AGE 4
            DW      093             ; #9  AGE 4
            DW      094             ; #10 AGE 4
            DW      095             ; #11 AGE 4
            DW      095             ; #12 AGE 4
            DW      096             ; #13 AGE 4
            DW      091             ; #14 AGE 1 ;sleep was 96 dmh
            DW      097             ; #15 AGE 4 FIXED DMH WAS 96
            DW      098             ; #16 AGE 4

;END HANGOUT
;END GEORGE 07/04/98
```

```
;GEORGE 07/07/98
;INVERT
;Ball invert sensor table
;
Invrt_S1: DW     202              ; #1  AGE 1
          DW     203              ; #2  AGE 1
          DW     206              ; #3  AGE 1
          DW     208              ; #4  AGE 1
          DW     212              ; #5  AGE 1
          DW     213              ; #6  AGE 1
          DW     217              ; #7  AGE 1
          DW     219              ; #8  AGE 1
          DW     220              ; #9  AGE 1
          DW     224              ; #10 AGE 1
          DW     228              ; #11 AGE 1
          DW     232              ; #12 AGE 1
          DW     234              ; #13 AGE 1
          DW     232              ; #14 AGE 1
          DW     234              ; #15 AGE 1
          DW     235              ; #16 AGE 1

Invrt_S2: DW     202              ; #1  AGE 2
          DW     203              ; #2  AGE 2
          DW     207              ; #3  AGE 2
          DW     209              ; #4  AGE 2
          DW     212              ; #5  AGE 2
          DW     214              ; #6  AGE 2
          DW     217              ; #7  AGE 2
          DW     219              ; #8  AGE 2
          DW     221              ; #9  AGE 2
          DW     225              ; #10 AGE 2
          DW     229              ; #11 AGE 2
          DW     232              ; #12 AGE 2
          DW     234              ; #13 AGE 2
          DW     232              ; #14 AGE 2
          DW     234              ; #15 AGE 2
          DW     236              ; #16 AGE 2

Invrt_S3: DW     202              ; #1  AGE 3
          DW     204              ; #2  AGE 3
          DW     207              ; #3  AGE 3
          DW     210              ; #4  AGE 3
          DW     212              ; #5  AGE 3
          DW     215              ; #6  AGE 3
          DW     218              ; #7  AGE 3
          DW     219              ; #8  AGE 3
          DW     222              ; #9  AGE 3
          DW     226              ; #10 AGE 3
          DW     230              ; #11 AGE 3
          DW     232              ; #12 AGE 3
          DW     234              ; #13 AGE 3
          DW     232              ; #14 AGE 3
          DW     234              ; #15 AGE 3
          DW     237              ; #16 AGE 3

Invrt_S4: DW     202              ; #1  AGE 4
          DW     205              ; #2  AGE 4
          DW     207              ; #3  AGE 4
          DW     211              ; #4  AGE 4
```

```
            DW      212             ; #5  AGE 4
            DW      216             ; #6  AGE 4
            DW      218             ; #7  AGE 4
            DW      219             ; #8  AGE 4
            DW      223             ; #9  AGE 4
            DW      227             ; #10 AGE 4
            DW      231             ; #11 AGE 4
            DW      233             ; #12 AGE 4
            DW      231             ; #13 AGE 4
            DW      233             ; #14 AGE 4
            DW      234             ; #15 AGE 4
            DW      238             ; #16 AGE 4

;GEORGE 07/07/98
;BACK
;touch back sensor table
;
Tback_S1:   DW      239             ; #1  AGE 1
            DW      240             ; #2  AGE 1
            DW      244             ; #3  AGE 1
            DW      248             ; #4  AGE 1
            DW      249             ; #5  AGE 1
            DW      248             ; #6  AGE 1
            DW      253             ; #7  AGE 1
            DW      256             ; #8  AGE 1
            DW      258             ; #9  AGE 1
            DW      239             ; #10 AGE 1
            DW      248             ; #11 AGE 1
            DW      261             ; #12 AGE 1
            DW      263             ; #13 AGE 1
            DW      266             ; #14 AGE 1
            DW      269             ; #15 AGE 1
            DW      272             ; #16 AGE 1

Tback_S2:   DW      239             ; #1  AGE 2
            DW      241             ; #2  AGE 2
            DW      245             ; #3  AGE 2
            DW      248             ; #4  AGE 2
            DW      250             ; #5  AGE 2
            DW      248             ; #6  AGE 2
            DW      253             ; #7  AGE 2
            DW      257             ; #8  AGE 2
            DW      259             ; #9  AGE 2
            DW      239             ; #10 AGE 2
            DW      248             ; #11 AGE 2
            DW      262             ; #12 AGE 2
            DW      264             ; #13 AGE 2
            DW      267             ; #14 AGE 2
            DW      270             ; #15 AGE 2
            DW      273             ; #16 AGE 2

Tback_S3:   DW      239             ; #1  AGE 3
            DW      242             ; #2  AGE 3
            DW      246             ; #3  AGE 3
            DW      248             ; #4  AGE 3
            DW      251             ; #5  AGE 3
            DW      248             ; #6  AGE 3
            DW      254             ; #7  AGE 3
            DW      257             ; #8  AGE 3
            DW      260             ; #9  AGE 3
```

```
            DW    239               ; #10 AGE 3
            DW    248               ; #11 AGE 3
            DW    261               ; #12 AGE 3
            DW    265               ; #13 AGE 3
            DW    268               ; #14 AGE 3
            DW    271               ; #15 AGE 3
            DW    274               ; #16 AGE 3

Tback_S4:   DW    239               ; #1  AGE 4
            DW    243               ; #2  AGE 4
            DW    247               ; #3  AGE 4
            DW    248               ; #4  AGE 4
            DW    252               ; #5  AGE 4
            DW    248               ; #6  AGE 4
            DW    255               ; #7  AGE 4
            DW    257               ; #8  AGE 4
            DW    260               ; #9  AGE 4
            DW    239               ; #10 AGE 4
            DW    248               ; #11 AGE 4
            DW    262               ; #12 AGE 4
            DW    265               ; #13 AGE 4
            DW    268               ; #14 AGE 4
            DW    271               ; #15 AGE 4
            DW    275               ; #16 AGE 4
;
;END GEORGE 07/07/98


;
;I.R. receive table
;DO IR

IR_S1:      DW    393               ; #1  AGE 1
            DW    393               ; #2  AGE 1
            DW    393               ; #3  AGE 1
            DW    393               ; #4  AGE 1
            DW    394               ; #5  AGE 1
            DW    395               ; #6  AGE 1
            DW    396               ; #7  AGE 1
            DW    396               ; #8  AGE 1
            DW    291               ; #9  AGE 1
            DW    399               ; #10 AGE 1
            DW    399               ; #11 AGE 1
            DW    400               ; #12 AGE 1
            DW    401               ; #13 AGE 1
            DW    401               ; #14 AGE 1
            DW    402               ; #15 AGE 1
            DW    403               ; #16 AGE 1

IR_S2:      DW    404               ; #1  AGE 2
            DW    404               ; #2  AGE 2
            DW    404               ; #3  AGE 2
            DW    405               ; #4  AGE 2
            DW    405               ; #5  AGE 2
            DW    406               ; #6  AGE 2
            DW    407               ; #7  AGE 2
            DW    407               ; #8  AGE 2
            DW    291               ; #9  AGE 2
            DW    409               ; #10 AGE 2
            DW    409               ; #11 AGE 2
```

```
            DW    400               ; #12 AGE 2
            DW    411               ; #13 AGE 2
            DW    411               ; #14 AGE 2
            DW    412               ; #15 AGE 2
            DW    413               ; #16 AGE 2

IR_S3:      DW    414               ; #1  AGE 3
            DW    414               ; #2  AGE 3
            DW    414               ; #3  AGE 3
            DW    414               ; #4  AGE 3
            DW    414               ; #5  AGE 3
            DW    415               ; #6  AGE 3
            DW    416               ; #7  AGE 3
            DW    416               ; #8  AGE 3
            DW    291               ; #9  AGE 3
            DW    408               ; #10 AGE 3
            DW    418               ; #11 AGE 3
            DW    428               ; #12 AGE 3
            DW    419               ; #13 AGE 3
            DW    419               ; #14 AGE 3
            DW    420               ; #15 AGE 3
            DW    403               ; #16 AGE 3

IR_S4:      DW    421               ; #1  AGE 4
            DW    421               ; #2  AGE 4
            DW    421               ; #3  AGE 4
            DW    421               ; #4  AGE 4
            DW    421               ; #5  AGE 4
            DW    422               ; #6  AGE 4
            DW    423               ; #7  AGE 4
            DW    423               ; #8  AGE 4
            DW    291               ; #9  AGE 4
            DW    425               ; #10 AGE 4
            DW    426               ; #11 AGE 4
            DW    427               ; #12 AGE 4
            DW    428               ; #13 AGE 4
            DW    428               ; #14 AGE 4
            DW    429               ; #15 AGE 4
            DW    413               ; #16 AGE 3
;
;
;
;light sense table (bright sense)
;DO LIGHT
Light_S1:

            DW    293             ; #1  AGE 1
            DW    305     ;003        ; #2  AGE 1
            DW    294               ; #3  AGE 1
            DW    295               ; #4  AGE 1
            DW    296               ; #5  AGE 1
            DW    297               ; #6  AGE 1
            DW    298               ; #7  AGE 1
            DW    299               ; #8  AGE 1
            DW    293             ; #9  AGE 1
            DW    305     ;003              ; #10 AGE 1
            DW    294               ; #11 AGE 1
            DW    295               ; #12 AGE 1
            DW    296               ; #13 AGE 1
            DW    297               ; #14 AGE 1
```

```
            DW    298             ; #15 AGE 1
            DW    299             ; #16 AGE 1
Light_S2:
            DW    293            ; #1  AGE 2
            DW    305     ;003            ; #2  AGE 2
            DW    294             ; #3  AGE 2
            DW    300             ; #4  AGE 2
            DW    296             ; #5  AGE 2
            DW    301             ; #6  AGE 2
            DW    298             ; #7  AGE 2
            DW    299             ; #8  AGE 2
            DW    293            ; #9  AGE 2
            DW    305     ;003            ; #10 AGE 2
            DW    294             ; #11 AGE 2
            DW    295             ; #12 AGE 2
            DW    296             ; #13 AGE 2
            DW    301             ; #14 AGE 2
            DW    298             ; #15 AGE 2
            DW    299             ; #16 AGE 2

Light_S3:
            DW    302             ; #1  AGE 3
            DW    305     ;003            ; #2  AGE 3
            DW    294             ; #3  AGE 3
            DW    303             ; #4  AGE 3
            DW    296             ; #5  AGE 3
            DW    304             ; #6  AGE 3
            DW    298             ; #7  AGE 3
            DW    299             ; #8  AGE 3
            DW    302             ; #9  AGE 3
            DW    305     ;003            ; #10 AGE 3
            DW    294             ; #11 AGE 3
            DW    303             ; #12 AGE 3
            DW    296             ; #13 AGE 3
            DW    304             ; #14 AGE 3
            DW    298             ; #15 AGE 3
            DW    299             ; #16 AGE 3

Light_S4:
            DW    302             ; #1  AGE 4
            DW    305     ;003            ; #2  AGE 4
            DW    294             ; #3  AGE 4
            DW    306             ; #4  AGE 4
            DW    296             ; #5  AGE 4
            DW    307             ; #6  AGE 4
            DW    298             ; #7  AGE 4
            DW    299             ; #8  AGE 4
            DW    302             ; #9  AGE 4
            DW    305     ;003            ; #10 AGE 4
            DW    294             ; #11 AGE 4
            DW    306             ; #12 AGE 4
            DW    296             ; #13 AGE 4
            DW    307             ; #14 AGE 4
            DW    298             ; #15 AGE 4
            DW    299             ; #16 AGE 4
;
;
;light sense table (DARK SENSE)
; DO DARK
```

```
; DO LIGHT DARKER
Dark_S1:  DW    308             ; #1  AGE 1
          DW    309             ; #2  AGE 1
          DW    310             ; #3  AGE 1
          DW    311             ; #4  AGE 1
          DW    312             ; #5  AGE 1
          DW    313             ; #6  AGE 1
          DW    314             ; #7  AGE 1
          DW    315             ; #8  AGE 1
          DW    308             ; #9  AGE 1
          DW    309             ; #10 AGE 1
          DW    310             ; #11 AGE 1
          DW    311             ; #12 AGE 1
          DW    312             ; #13 AGE 1
          DW    313             ; #14 AGE 1
          DW    314             ; #15 AGE 1
          DW    315             ; #16 AGE 1

Dark_S2:
          DW    316             ; #1  AGE 2
          DW    317             ; #2  AGE 2
          DW    318             ; #3  AGE 2
          DW    311             ; #4  AGE 2
          DW    319             ; #5  AGE 2
          DW    313             ; #6  AGE 2
          DW    320             ; #7  AGE 2
          DW    315             ; #8  AGE 2
          DW    316             ; #9  AGE 2
          DW    317             ; #10 AGE 2
          DW    318             ; #11 AGE 2
          DW    311             ; #12 AGE 2
          DW    319             ; #13 AGE 2
          DW    313             ; #14 AGE 2
          DW    320             ; #15 AGE 2
          DW    315             ; #16 AGE 2

Dark_S3:  DW    321             ; #1  AGE 3
          DW    322             ; #2  AGE 3
          DW    323             ; #3  AGE 3
          DW    311             ; #4  AGE 3
          DW    319             ; #5  AGE 3
          DW    313             ; #6  AGE 3
          DW    324             ; #7  AGE 3
          DW    325             ; #8  AGE 3
          DW    321             ; #9  AGE 3
          DW    322             ; #10 AGE 3
          DW    323             ; #11 AGE 3
          DW    311             ; #12 AGE 3
          DW    319             ; #13 AGE 3
          DW    313             ; #14 AGE 3
          DW    324             ; #15 AGE 3
          DW    325             ; #16 AGE 3

Dark_S4:  DW    326             ; #1  AGE 4
          DW    327             ; #2  AGE 4
          DW    328             ; #3  AGE 4
          DW    311             ; #4  AGE 4
          DW    329             ; #5  AGE 4
          DW    313             ; #6  AGE 4
          DW    330             ; #7  AGE 4
```

```
            DW      331             ; #8  AGE 4
            DW      326             ; #9  AGE 4
            DW      327             ; #10 AGE 4
            DW      328             ; #11 AGE 4
            DW      311             ; #12 AGE 4
            DW      329             ; #13 AGE 4
            DW      313             ; #14 AGE 4
            DW      330             ; #15 AGE 4
            DW      331             ; #16 AGE 4
;
;
;
;
; Hide and Seek game table

Peek_S1:  DW    000             ; #0  AGE 1
          DW  000           ; #1  AGE 1
          DW  000           ; #2  AGE 1
          DW  000           ; #3  AGE 1
          DW  000           ; #4  AGE 1
          DW  000           ; #5  AGE 1
          DW  000           ; #6  AGE 1
          DW  000           ; #7  AGE 1
          DW  000           ; #8  AGE 1
          DW  000           ; #9  AGE 1
          DW  000           ; #10 AGE 1
          DW  000           ; #11 AGE 1
          DW  000           ; #12 AGE 1
          DW  000           ; #13 AGE 1
          DW  000           ; #14 AGE 1
          DW  000           ; #15 AGE 1

Peek_S2:  DW    000             ; #0  AGE 2
          DW  000           ; #1  AGE 2
          DW  000           ; #2  AGE 2
          DW  000           ; #3  AGE 2
          DW  000           ; #4  AGE 2
          DW  000           ; #5  AGE 2
          DW  000           ; #6  AGE 2
          DW  000           ; #7  AGE 2
          DW  000           ; #8  AGE 2
          DW  000           ; #9  AGE 2
          DW  000           ; #10 AGE 2
          DW  000           ; #11 AGE 2
          DW  000           ; #12 AGE 2
          DW  000           ; #13 AGE 2
          DW  000           ; #14 AGE 2
          DW  000           ; #15 AGE 2

Peek_S3:  DW    000             ; #0  AGE 3
          DW  000           ; #1  AGE 3
          DW  000           ; #2  AGE 3
          DW  000           ; #3  AGE 3
          DW  000           ; #4  AGE 3
          DW  000           ; #5  AGE 3
          DW  000           ; #6  AGE 3
          DW  000           ; #7  AGE 3
          DW  000           ; #8  AGE 3
          DW  000           ; #9  AGE 3
          DW  000           ; #10 AGE 3
```

```
        DW  000          ; #11 AGE 3
        DW  000          ; #12 AGE 3
        DW  000          ; #13 AGE 3
        DW  000          ; #14 AGE 3
        DW  000          ; #15 AGE 3

Peek_S4:  DW    000              ; #0  AGE 4
        DW  000          ; #1  AGE 4
        DW  000          ; #2  AGE 4
        DW  000          ; #3  AGE 4
        DW  000          ; #4  AGE 4
        DW  000          ; #5  AGE 4
        DW  000          ; #6  AGE 4
        DW  000          ; #7  AGE 4
        DW  000          ; #8  AGE 4
        DW  000          ; #9  AGE 4
        DW  000          ; #10 AGE 4
        DW  000          ; #11 AGE 4
        DW  000          ; #12 AGE 4
        DW  000          ; #13 AGE 4
        DW  000          ; #14 AGE 4
          DW    000              ; #15 AGE 4
;
;
;***********************************************
;***********************************************
;***********************************************

Macro_grp1:  ;points into macro tables

 DW    Tbl1_Macro0
 DW    Tbl1_Macro1,Tbl1_Macro2,Tbl1_Macro3,Tbl1_Macro4,Tbl1_Macro5
 DW    Tbl1_Macro6,Tbl1_Macro7,Tbl1_Macro8,Tbl1_Macro9,Tbl1_Macro10
 DW    Tbl1_Macro11,Tbl1_Macro12,Tbl1_Macro13,Tbl1_Macro14,Tbl1_Macro15
 DW    Tbl1_Macro16,Tbl1_Macro17,Tbl1_Macro18,Tbl1_Macro19,Tbl1_Macro20
 DW    Tbl1_Macro21,Tbl1_Macro22,Tbl1_Macro23,Tbl1_Macro24,Tbl1_Macro25
 DW    Tbl1_Macro26,Tbl1_Macro27,Tbl1_Macro28,Tbl1_Macro29,Tbl1_Macro30
 DW    Tbl1_Macro31,Tbl1_Macro32,Tbl1_Macro33,Tbl1_Macro34,Tbl1_Macro35
 DW    Tbl1_Macro36,Tbl1_Macro37,Tbl1_Macro38,Tbl1_Macro39,Tbl1_Macro40
 DW    Tbl1_Macro41,Tbl1_Macro42,Tbl1_Macro43,Tbl1_Macro44,Tbl1_Macro45
 DW    Tbl1_Macro46,Tbl1_Macro47,Tbl1_Macro48,Tbl1_Macro49,Tbl1_Macro50
 DW    Tbl1_Macro51,Tbl1_Macro52,Tbl1_Macro53,Tbl1_Macro54,Tbl1_Macro55
 DW    Tbl1_Macro56,Tbl1_Macro57,Tbl1_Macro58,Tbl1_Macro59,Tbl1_Macro60
 DW    Tbl1_Macro61,Tbl1_Macro62,Tbl1_Macro63,Tbl1_Macro64,Tbl1_Macro65
 DW    Tbl1_Macro66,Tbl1_Macro67,Tbl1_Macro68,Tbl1_Macro69,Tbl1_Macro70
 DW    Tbl1_Macro71,Tbl1_Macro72,Tbl1_Macro73,Tbl1_Macro74,Tbl1_Macro75
 DW    Tbl1_Macro76,Tbl1_Macro77,Tbl1_Macro78,Tbl1_Macro79,Tbl1_Macro80
 DW    Tbl1_Macro81,Tbl1_Macro82,Tbl1_Macro83,Tbl1_Macro84,Tbl1_Macro85
 DW    Tbl1_Macro86,Tbl1_Macro87,Tbl1_Macro88,Tbl1_Macro89,Tbl1_Macro90
 DW    Tbl1_Macro91,Tbl1_Macro92,Tbl1_Macro93,Tbl1_Macro94,Tbl1_Macro95
 DW    Tbl1_Macro96,Tbl1_Macro97,Tbl1_Macro98,Tbl1_Macro99
 DW    Tbl1_Macro100,Tbl1_Macro101,Tbl1_Macro102,Tbl1_Macro103,Tbl1_Macro
104
 DW    Tbl1_Macro105,Tbl1_Macro106,Tbl1_Macro107,Tbl1_Macro108,Tbl1_Macro
109
 DW    Tbl1_Macro110,Tbl1_Macro111,Tbl1_Macro112,Tbl1_Macro113,Tbl1_Macro
114
 DW    Tbl1_Macro115,Tbl1_Macro116,Tbl1_Macro117,Tbl1_Macro118,Tbl1_Macro
119
 DW    Tbl1_Macro120,Tbl1_Macro121,Tbl1_Macro122,Tbl1_Macro123,Tbl1_Macro
```

```
124
 DW     Tbl1_Macro125,Tbl1_Macro126,Tbl1_Macro127
;
Macro_grp2:  ;points into macro tables

 DW    Tbl2_Macro128
 DW    Tbl2_Macro129,Tbl2_Macro130,Tbl2_Macro131,Tbl2_Macro132,Tbl2_Macro
133
 DW    Tbl2_Macro134,Tbl2_Macro135,Tbl2_Macro136,Tbl2_Macro137,Tbl2_Macro
138
 DW    Tbl2_Macro139,Tbl2_Macro140,Tbl2_Macro141,Tbl2_Macro142,Tbl2_Macro
143
 DW    Tbl2_Macro144,Tbl2_Macro145,Tbl2_Macro146,Tbl2_Macro147,Tbl2_Macro
148
 DW    Tbl2_Macro149,Tbl2_Macro150,Tbl2_Macro151,Tbl2_Macro152,Tbl2_Macro
153
 DW    Tbl2_Macro154,Tbl2_Macro155,Tbl2_Macro156,Tbl2_Macro157,Tbl2_Macro
158
 DW    Tbl2_Macro159,Tbl2_Macro160,Tbl2_Macro161,Tbl2_Macro162,Tbl2_Macro
163
 DW    Tbl2_Macro164,Tbl2_Macro165,Tbl2_Macro166,Tbl2_Macro167,Tbl2_Macro
168
 DW    Tbl2_Macro169,Tbl2_Macro170,Tbl2_Macro171,Tbl2_Macro172,Tbl2_Macro
173
 DW    Tbl2_Macro174,Tbl2_Macro175,Tbl2_Macro176,Tbl2_Macro177,Tbl2_Macro
178
 DW    Tbl2_Macro179,Tbl2_Macro180,Tbl2_Macro181,Tbl2_Macro182,Tbl2_Macro
183
 DW    Tbl2_Macro184,Tbl2_Macro185,Tbl2_Macro186,Tbl2_Macro187,Tbl2_Macro
188
 DW    Tbl2_Macro189,Tbl2_Macro190,Tbl2_Macro191,Tbl2_Macro192,Tbl2_Macro
193
 DW    Tbl2_Macro194,Tbl2_Macro195,Tbl2_Macro196,Tbl2_Macro197,Tbl2_Macro
198
 DW    Tbl2_Macro199,Tbl2_Macro200,Tbl2_Macro201,Tbl2_Macro202,Tbl2_Macro
203
 DW    Tbl2_Macro204,Tbl2_Macro205,Tbl2_Macro206,Tbl2_Macro207,Tbl2_Macro
208
 DW    Tbl2_Macro209,Tbl2_Macro210,Tbl2_Macro211,Tbl2_Macro212,Tbl2_Macro
213
 DW    Tbl2_Macro214,Tbl2_Macro215,Tbl2_Macro216,Tbl2_Macro217,Tbl2_Macro
218
 DW    Tbl2_Macro219,Tbl2_Macro220,Tbl2_Macro221,Tbl2_Macro222,Tbl2_Macro
223
 DW    Tbl2_Macro224,Tbl2_Macro225,Tbl2_Macro226,Tbl2_Macro227,Tbl2_Macro
228
 DW    Tbl2_Macro229,Tbl2_Macro230,Tbl2_Macro231,Tbl2_Macro232,Tbl2_Macro
233
 DW    Tbl2_Macro234,Tbl2_Macro235,Tbl2_Macro236,Tbl2_Macro237,Tbl2_Macro
238
 DW    Tbl2_Macro239,Tbl2_Macro240,Tbl2_Macro241,Tbl2_Macro242,Tbl2_Macro
243
 DW    Tbl2_Macro244,Tbl2_Macro245,Tbl2_Macro246,Tbl2_Macro247,Tbl2_Macro
248
 DW    Tbl2_Macro249,Tbl2_Macro250,Tbl2_Macro251,Tbl2_Macro252,Tbl2_Macro
253
 DW    Tbl2_Macro254,Tbl2_Macro255
;
Macro_grp3: ; points into macro tables
```

```
 DW    Tbl3_Macro256
 DW    Tbl3_Macro257,Tbl3_Macro258,Tbl3_Macro259,Tbl3_Macro260,Tbl3_Macro
261
 DW    Tbl3_Macro262,Tbl3_Macro263,Tbl3_Macro264,Tbl3_Macro265,Tbl3_Macro
266
 DW    Tbl3_Macro267,Tbl3_Macro268,Tbl3_Macro269,Tbl3_Macro270,Tbl3_Macro
271
 DW    Tbl3_Macro272,Tbl3_Macro273,Tbl3_Macro274,Tbl3_Macro275,Tbl3_Macro
276
 DW    Tbl3_Macro277,Tbl3_Macro278,Tbl3_Macro279,Tbl3_Macro280,Tbl3_Macro
281
 DW    Tbl3_Macro282,Tbl3_Macro283,Tbl3_Macro284,Tbl3_Macro285,Tbl3_Macro
286
 DW    Tbl3_Macro287,Tbl3_Macro288,Tbl3_Macro289,Tbl3_Macro290,Tbl3_Macro
291
 DW    Tbl3_Macro292,Tbl3_Macro293,Tbl3_Macro294,Tbl3_Macro295,Tbl3_Macro
296
 DW    Tbl3_Macro297,Tbl3_Macro298,Tbl3_Macro299,Tbl3_Macro300,Tbl3_Macro
301
 DW    Tbl3_Macro302,Tbl3_Macro303,Tbl3_Macro304,Tbl3_Macro305,Tbl3_Macro
306
 DW    Tbl3_Macro307,Tbl3_Macro308,Tbl3_Macro309,Tbl3_Macro310,Tbl3_Macro
311
 DW    Tbl3_Macro312,Tbl3_Macro313,Tbl3_Macro314,Tbl3_Macro315,Tbl3_Macro
316
 DW    Tbl3_Macro317,Tbl3_Macro318,Tbl3_Macro319,Tbl3_Macro320,Tbl3_Macro
321
 DW    Tbl3_Macro322,Tbl3_Macro323,Tbl3_Macro324,Tbl3_Macro325,Tbl3_Macro
326
 DW    Tbl3_Macro327,Tbl3_Macro328,Tbl3_Macro329,Tbl3_Macro330,Tbl3_Macro
331
 DW    Tbl3_Macro332,Tbl3_Macro333,Tbl3_Macro334,Tbl3_Macro335,Tbl3_Macro
336
 DW    Tbl3_Macro337,Tbl3_Macro338,Tbl3_Macro339,Tbl3_Macro340,Tbl3_Macro
341
 DW    Tbl3_Macro342,Tbl3_Macro343,Tbl3_Macro344,Tbl3_Macro345,Tbl3_Macro
346
 DW    Tbl3_Macro347,Tbl3_Macro348,Tbl3_Macro349,Tbl3_Macro350,Tbl3_Macro
351
 DW    Tbl3_Macro352,Tbl3_Macro353,Tbl3_Macro354,Tbl3_Macro355,Tbl3_Macro
356
 DW    Tbl3_Macro357,Tbl3_Macro358,Tbl3_Macro359,Tbl3_Macro360,Tbl3_Macro
361
 DW    Tbl3_Macro362,Tbl3_Macro363,Tbl3_Macro364,Tbl3_Macro365,Tbl3_Macro
366
 DW    Tbl3_Macro367,Tbl3_Macro368,Tbl3_Macro369,Tbl3_Macro370,Tbl3_Macro
371
 DW    Tbl3_Macro372,Tbl3_Macro373,Tbl3_Macro374,Tbl3_Macro375,Tbl3_Macro
376
 DW    Tbl3_Macro377,Tbl3_Macro378,Tbl3_Macro379,Tbl3_Macro380,Tbl3_Macro
381
 DW    Tbl3_Macro382,Tbl3_Macro383
;
Macro_grp4:  ;points into macro tables

 DW    Tbl4_Macro384
 DW    Tbl4_Macro385,Tbl4_Macro386,Tbl4_Macro387,Tbl4_Macro388,Tbl4_Macro
389
 DW    Tbl4_Macro390,Tbl4_Macro391,Tbl4_Macro392,Tbl4_Macro393,Tbl4_Macro
394
```

```
 DW    Tbl4_Macro395,Tbl4_Macro396,Tbl4_Macro397,Tbl4_Macro398,Tbl4_Macro
399
 DW    Tbl4_Macro400,Tbl4_Macro401,Tbl4_Macro402,Tbl4_Macro403,Tbl4_Macro
404
 DW    Tbl4_Macro405,Tbl4_Macro406,Tbl4_Macro407,Tbl4_Macro408,Tbl4_Macro
409
 DW    Tbl4_Macro410,Tbl4_Macro411,Tbl4_Macro412,Tbl4_Macro413,Tbl4_Macro
414
 DW    Tbl4_Macro415,Tbl4_Macro416,Tbl4_Macro417,Tbl4_Macro418,Tbl4_Macro
419
 DW    Tbl4_Macro420,Tbl4_Macro421,Tbl4_Macro422,Tbl4_Macro423,Tbl4_Macro
424
 DW    Tbl4_Macro425,Tbl4_Macro426,Tbl4_Macro427,Tbl4_Macro428,Tbl4_Macro
429
 DW    Tbl4_Macro430,Tbl4_Macro431,Tbl4_Macro432,Tbl4_Macro433,Tbl4_Macro
434
 DW    Tbl4_Macro435,Tbl4_Macro436,Tbl4_Macro437,Tbl4_Macro438,Tbl4_Macro
439
 DW    Tbl4_Macro440,Tbl4_Macro441,Tbl4_Macro442,Tbl4_Macro443,Tbl4_Macro
444
 DW    Tbl4_Macro445,Tbl4_Macro446,Tbl4_Macro447,Tbl4_Macro448,Tbl4_Macro
449
 DW    Tbl4_Macro450,Tbl4_Macro451,Tbl4_Macro452,Tbl4_Macro453,Tbl4_Macro
454
 DW    Tbl4_Macro455,Tbl4_Macro456,Tbl4_Macro457,Tbl4_Macro458,Tbl4_Macro
459
 DW    Tbl4_Macro460,Tbl4_Macro461,Tbl4_Macro462,Tbl4_Macro463,Tbl4_Macro
464
 DW    Tbl4_Macro465,Tbl4_Macro466,Tbl4_Macro467,Tbl4_Macro468,Tbl4_Macro
469
 DW    Tbl4_Macro470,Tbl4_Macro471,Tbl4_Macro472,Tbl4_Macro473,Tbl4_Macro
474
 DW    Tbl4_Macro475,Tbl4_Macro476,Tbl4_Macro477,Tbl4_Macro478,Tbl4_Macro
479
 DW    Tbl4_Macro480,Tbl4_Macro481,Tbl4_Macro482,Tbl4_Macro483,Tbl4_Macro
484
 DW    Tbl4_Macro485,Tbl4_Macro486,Tbl4_Macro487,Tbl4_Macro488,Tbl4_Macro
489
 DW    Tbl4_Macro490,Tbl4_Macro491,Tbl4_Macro492,Tbl4_Macro493,Tbl4_Macro
494
 DW    Tbl4_Macro495,Tbl4_Macro496,Tbl4_Macro497,Tbl4_Macro498,Tbl4_Macro
499
 DW    Tbl4_Macro500,Tbl4_Macro501,Tbl4_Macro502,Tbl4_Macro503,Tbl4_Macro
504
 DW    Tbl4_Macro505,Tbl4_Macro506,Tbl4_Macro507,Tbl4_Macro508,Tbl4_Macro
509
 DW    Tbl4_Macro510,Tbl4_Macro511

;***********************************************
;***********************************************
;***********************************************


; MACRO TABLES
;
; The sensor tables point into the Macro table. This table in turn
; gets speech and motor table data.
; This can be an entry of 1-511 and effectively chains motor and
; speech tables together to reuse previous speech motor segments.
```

```
; The first group of numbers is the speech/motor table value.
; The last line is the terminator of 00. (00 so 'DB' takes 1 less byte)
;
; ex: 1 = will call the saysent 1 and the motor table 1.

Tbl1_Macro0:
        DW      511
      DW    00    ;end

; FOR NAME TESTING DMH
; WAKE
;         DW      124     ;02
;         DW      125
;         DW      126
;
;
;         DW      399     ; delay
;         DW      395     ; ME
;         DW      224     ; MAY-LAH-KA
;         DW      152
;       DW    00    ;end

;
; (MIDDLE)
;
;
; put sounds and motions together
;  DW 5          (first sound and motion, in this case "5")
;  DW 3          (next sound and motion, in this case "3")
;  DW 00         ( end of sequence)
;

Tbl1_Macro1:
        DW      01
        DW      00      ;end

;GEORGE 07/03/98
Tbl1_Macro2:
        DW      001      ;FRONT SEQ1AGE1
        DW      00      ;end
;
Tbl1_Macro3:
        DW      002      ;FRONT SEQ2AGE1
        DW      00      ;end
;
Tbl1_Macro4:
        DW      003     ;FRONT SEQ3AGE1
        DW      004
        DW      00      ;end
;
Tbl1_Macro5:
        DW      003     ;FRONT SEQ4AGE1
        DW      005
        DW      00      ;end
;
Tbl1_Macro6:
        DW      006      ;FRONT SEQ5AGE1
        DW      00      ;end
;
```

```
Tbl1_Macro7:
        DW      006         ;FRONTSEQ6AGE1
        DW      007
        DW      00        ;end
;
Tbl1_Macro8:
        DW      008         ;FRONT SEQ7AGE1
        DW      003
        DW      00        ;end
;
Tbl1_Macro9:
        DW      009         ;FRONTSEQ8AGE1
        DW      003
        DW      00        ;end
;
Tbl1_Macro10:
        DW      010               ;FRONT SEQ9age1
        DW      00        ;end
;
Tbl1_Macro11:
        DW      011
        DW      001               ;frontseq10age1
        DW      00        ;end
;
Tbl1_Macro12:
        DW      012
        DW      001          ;seq11 FRONT AGE1 ADD SAY001
        DW      00        ;end
;
Tbl1_Macro13:
        DW      001
        DW      013               ;seq12 FRONT AGE1 ADD SAY001
        DW      00        ;end
;
Tbl1_Macro14:
        DW      014                ;seq13 FRONT AGE1 ADD SAY003
        DW      003
        DW      00        ;end
;
Tbl1_Macro15:
        DW      015               ;seq14 FRONT AGE1
        DW      00        ;end
;
Tbl1_Macro16:
        DW      016                ;seq15 FRONT AGE1
        DW      00        ;end
;
Tbl1_Macro17:
        DW      001
        DW      017
        DW      018
        DW      001                    ;seq16 FRONT AGE1 BETWEEN 2(20)
        DW      00        ;end
;
Tbl1_Macro18:
        DW      019               ;FRONT SEQ1AGE2
        DW      00        ;end
;
Tbl1_Macro19:
        DW      001
```

```
        DW      020             ;FRONT SEQ2 AGE2
        DW      00      ;end
;
Tbl1_Macro20:
        DW      010
        DW      021      ;SEQ3AGE2 FRONT ADD SEQ9AGE1
        DW      00      ;end
;
Tbl1_Macro21:
        DW      022       ;SEQ4 AGE2 FRONT
        DW      023
        DW      00       ;end
;
Tbl1_Macro22:
        DW      024       ;SEQ5 AGE2 FRONT
        DW      00       ;end
;
Tbl1_Macro23:
        DW      025       ;SEQ6 AGE2 FRONT
        DW      00       ;end
;
Tbl1_Macro24:
        DW      026      ;SEQ 7 AGE2 FRONT PART1
        DW      027
        DW      00      ;end
;
Tbl1_Macro25:

        DW      026
        DW      026      ;SEQ 8 AGE2 FRONT
        DW      028
        DW      003
        DW      00      ;end
;
Tbl1_Macro26:
        DW      029                     ;SEQ 9 FRONT
        DW      00      ;end
;
Tbl1_Macro27:
        DW      030
        DW      029      ;SEQ 10 FRONT AGE2
        DW      00      ;end
;
Tbl1_Macro28:
        DW      022
        DW      031      ;SEQ 11 FRONT AGE2
        DW      00      ;end
;
Tbl1_Macro29:
        DW      001
        DW      032      ;SEQ 12 FRONT AGE 2
        DW      00      ;end
;
Tbl1_Macro30:
        DW      014             ;seq13 FRONT AGE1&2 ADD SAY003
        DW      003
        DW      00      ;end
;
Tbl1_Macro31:
        DW      033             ;SEQ14 FRONT AGE2
```

```
        DW      00      ;end
;
Tbl1_Macro32:
        DW      034             ;SEQ15 FRONT AGE2
        DW      001
        DW      00      ;end
;
Tbl1_Mac o33:
        DW      001
        DW      035             ;SEQ16 FRONT AGE2
        DW      00      ;end
;
Tbl1_Macro34:
        DW      001
        DW      036             ;SEQ1 FRONT AGE3
        DW      00      ;end
;
Tbl1_Macro35:
        DW      003
        DW      037             ;SEQ2 FRONT AGE3
        DW      00      ;end
;
Tbl1_Macro36:
        DW      010
        DW      038             ;SEQ3 FRONT AGE3
        DW      00      ;end
;
Tbl1_Macro37:
        DW      015
        DW      039              ;SEQ4 FRONT AGE3
        DW      00      ;end
;
Tbl1_Macro38:
        DW      015
        DW      023              ;SEQ5 FRONT AGE3
        DW      00      ;end
;
Tbl1_Macro39:
        DW      040             ;SEQ6 FRONT AGE3
        DW      00      ;end
;
Tbl1_Macro40:
        DW      041              ;SEQ7 FRONT AGE3
        DW      003
        DW      00      ;end
;
Tbl1_Macro41:
        DW      042
        DW      003               ;SEQ8 FRONT AGE3
        DW      00      ;end
;
Tbl1_Macro42:
        DW      043             ;SEQ10 FRONT AGE3
        DW      001
        DW      00      ;end
;
Tbl1_Macro43:
        DW      044             ;SEQ11 FRONT AGE3
        DW      00      ;end
;
```

```
Tbl1_Macro44:
        DW      045
        DW      001         ;SEQ12 FRONT AGE3 (HEEY,TICKLE ME) ADD20
        DW      00        ;end
;
Tbl1_Macro45:
        DW      001
        DW      046             ;SEQ13 fRONT AGE3 (NANNY,NANNY) ADD20
        DW      047       ;RASBERRY HE HE HE
        DW      00        ;end
;
Tbl1_Macro46:
        DW      003
        DW      028       ;SEQ14 FRONT AGE3
        DW      003
        DW      00        ;end
;
Tbl1_Macro47:
        DW      034         ;SEQ15 FRONT AGE3
        DW      001
        DW      00        ;end
;
Tbl1_Macro48:
        DW      001
        DW      048
        DW      049           ;SEQ16 FRONT AGE3
        DW      00        ;end
;
Tbl1_Macro49:
        DW      044       ;SEQ1  FRONT AGE4
        DW      00        ;end
;
Tbl1_Macro50:
        DW      001
        DW      050         ; SEQ2  FRONT AGE4
        DW      051
        DW      00        ;end
;
Tbl1_Macro51:
        DW      003
        DW      052         ;SEQ3 (YOU) FRONT AGE4
        DW      050
        DW      053         ;SEQ3 (ME) FRONT AGE4
        DW      00        ;end
;
Tbl1_Macro52:
        DW      026
        DW      053
        DW      054
        DW      050 ;SEQ4  FRONT AGE4
        DW      001
        DW      00        ;end
;
Tbl1_Macro53:
        DW      007
        DW      055
        DW      056               ; SEQ5 FRONT AGE4
        DW      00        ;end
;
Tbl1_Macro54:
```

```
        DW      026
        DW      053
        DW      054
        DW      052
        DW      018                 ;SEQ6 FRONT AGE4
        DW      00          ;end
;
Tbl1_Macro55:
        DW      001
        DW      046
        DW      055           ;SEQ7 FRONT AGE4
        DW      00          ;end
;
Tbl1_Macro56:
        DW      026
        DW      057
        DW      050
        DW      051
        DW      058
        DW      003           ;SEQ8 FRONT AGE4
        DW      00          ;end
;
Tbl1_Macro57:
        DW      042,001       ;SEQ9 FRONT AGE4
        DW      00          ;end
;
Tbl1_Macro58:
        DW      059           ;SEQ10 FRONT AGE4
        DW      050
        DW      00          ;end
;
Tbl1_Macro59:
        DW      044
        DW      003           ;SEQ11 FRONT AGE4
        DW      00          ;end
;
Tbl1_Macro60:
        DW      001             ;SEQ12
        DW      00          ;end
;
Tbl1_Macro61:
        DW      001
        DW      046
        DW      047           SEQ13 FRONT AGE4
        DW      00          ;end
;
Tbl1_Macro62:
        DW      026
        DW      060           ;SEQ14 FRONT AGE4
        DW      00          ;end
;
Tbl1_Macro63:
        DW      061
        DW      003           SEQ15 FRONT AGE4
        DW      00          ;end
;
Tbl1_Macro64:
        DW      007
        DW      051           ;SEQ16 FRONT AGE4
        DW      00          ;end
```

```
;END GEORGE 07/03/98


;GEORGE 07/04/98
;START FORTUNE
;

Tbl1_Macro65:
        DW      062
        DW      051 ;72         ;FORTUNE 1
      DW    00     ;end
;
Tbl1_Macro66:
        DW      003
        DW      063       ;FORTUNE 2
        DW      003
      DW    00     ;end
;
Tbl1_Macro67:
        DW      090       ;94
        DW      064
        DW      063       ;FORTUNE 3
      DW    00     ;end
;
Tbl1_Macro68:
        DW      065       ;FORTUNE 4
        DW      063
      DW    00     ;end
;
Tbl1_Macro69:             ; MODIFIED FOR NAME DMH
;         DW      067      ;FORTUNE
;         DW      068
        DW      053
        DW      066       ;FORTUNE 5
        DW      063
      DW    00     ;end
;
Tbl1_Macro70:
        DW      069       ;FORTUNE 6
        DW      070
        DW      00        ;end
;
Tbl1_Macro71:
        DW      067
        DW      068       ;FORTUNE 7
        DW      071
        DW      073
        DW      072
      DW    00     ;end
;
Tbl1_Macro72:
        DW      074         ;FORTUNE 8
      DW    00     ;end
;
Tbl1_Macro73:
        DW      074         ;FORTUNE 9
        DW      063
      DW    00     ;end
;
Tbl1_Macro74:
```

```
        DW      069     ;FORTUNE 10
      DW    00    ;end
;
Tbl1_Macro75:
        DW      064     ;FORTUNE 11
        DW      069
      DW    00    ;end
;
Tbl1_Macro76:
        DW      073
        DW      064     ;FORTUNE 12
        DW      069
      DW    00    ;end
;
Tbl1_Macro77:           ; MODIFIED TO WORK WITH NAME DMH
;         DW      067
;         DW      068
        DW      053      ;FORTUNE 13
        DW      066
        DW      069
      DW    00    ;end
;
Tbl1_Macro78:
        DW      071
        DW      073
        DW      069
        DW      075     ;FORTUNE 14
      DW    00    ;end
;
Tbl1_Macro79:
        DW      076
        DW      077     ;FORTUNE 15
        DW      00      ;end
;
Tbl1_Macro80:
        DW      076
        DW      069     ;FORTUNE 16
      DW    00    ;end
;
Tbl1_Macro81:
        DW      078     ;FORTUNE 17 SEQ1 AGE2
      DW    00    ;end
;
Tbl1_Macro82:
        DW      078     ;FORTUNE 18  SEQ2 AGE2
        DW      063
      DW    00    ;end
;
Tbl1_Macro83:
        DW      078     ;FORTUNE 19  SEQ2 AGE2
        DW      069
      DW    00    ;end


Tbl1_Macro84:           ;SPECIAL "O TWO MA"
        DW      067     ;
        DW      068     ;
        DW      00
;END GEORGE 07/04/98
;END FORTUNE
```

```
;START HANGOUT
;GEORGE 07/04/98
Tbl1_Macro85:
        DW      079
        DW      080
        DW      079       ;SEQ1 HANGING
        DW      080
      DW    00    ;end
;
Tbl1_Macro86:
        DW      081       ;SEQ2 HANGING
        DW      081
      DW    00    ;end
;
Tbl1_Macro87:
        DW      082
        DW      083
;        DW     083
        DW      084 ;SEQ3 HANGING (YA DA DA OMPAH  bRUMM BABABUM)
      DW    00    ;end
;
Tbl1_Macro88:
        DW      085
        DW      085
        DW      086
        DW      087   ;SEQ4 HANGING (LA LA)
      DW    00    ;end
;
Tbl1_Macro89:
        DW      087
        DW      088    ;SEQ5 HANGING
      DW    00    ;end
;
Tbl1_Macro90:
        DW      089
        DW      089
        DW      090       ;SEQ6 HANGING
        DW      091
        DW      092
      DW    00    ;end
;
Tbl1_Macro91:
        DW      093        ;SEQ7 HANGING (SOFTER)
        DW      093
        DW      093
        DW      094
      DW    00    ;end
;
Tbl1_Macro92:
        DW      095
        DW      095
        DW      055     ;WAS 76      ;SEQ8 HANGING
      DW    00    ;end
;
Tbl1_Macro93:
        DW      096       ;SEQ9 HANGING
      DW    00    ;end
;
```

```
Tbl1_Macro94:
        DW      097     ;SEQ10 HANGING
      DW    00    ;end
;
Tbl1_Macro95:
        DW      098      ;SEQ11 AND SEQ12 HANGING (SIGH)
      DW    00    ;end
;
Tbl1_Macro96:
        DW      099      ;SEQ13 HANGING (HAA)
      DW    00    ;end
;
Tbl1_Macro97:
        DW      100     ;SEQ14 SEQ15 HANGING (hEEY)
      DW    00    ;end
;
Tbl1_Macro98:
        DW      101     ;SEQ16  hANGING (PHONE)
        DW      102
        DW      101
        DW      101
        DW      001     ;20
        DW      00      ;end
;
Tbl1_Macro99:
        DW      089    ;SEQ6 HANGING AGE2
        DW      089
        DW      090
        DW      091
        DW      103
      DW    00    ;end
;
Tbl1_Macro100:
        DW      089    ;SEQ6 HANGING AGE2
        DW      089
        DW      090
        DW      105
        DW      104
        DW      103
      DW    00    ;end
;
Tbl1_Macro101:
        DW      087
        DW      106     ;SEQ5 AGE3 4
      DW    00    ;end
;END HANGOUT
;
Tbl1_Macro102:
        DW      107     ;Fortune pause
      DW    00    ;end
;
;END GEORGE 07/04/98
;GEORGE 07/05/98

;FEED TABLE
Tbl1_Macro103:
        DW      108
        DW      110     ;SEQ2 FEED AGE1
        DW      109
      DW    00    ;end
```

```
;
Tbl1_Macro104:
        DW      108     ;SEQ3 FEED AGE1
        DW      111
        DW      112
        DW      109
      DW    00    ;end
;
Tbl1_Macro105:
        DW      108     ;SEQ4 FEED AGE1
        DW      110
        DW      113
        DW      109
      DW    00    ;end
;
Tbl1_Macro106:
        DW      108     ;SEQ5 FEED AGE1
        DW      108
        DW      078     ;127
        DW      110
        DW      109
      DW    00    ;end
;
Tbl1_Macro107:
        DW      108     ;SEQ6 FEED AGE1
        DW      105     ;109
        DW      114
      DW    00    ;end
;
Tbl1_Macro108:
        DW      108     ;SEQ7 FEED AGE1
        DW      115
        DW      116
        DW      117
        DW      110
      DW    00    ;end
;
Tbl1_Macro109:
        DW      076     ;125     ;SEQ8 FEED AGE1
        DW      117
        DW      120
        DW      118
      DW    00    ;end
;
Tbl1_Macro110:
        DW      108
        DW      115
      DW    20          ;SEQ9 FEED AGE1
      DW    00    ;end
;
Tbl1_Macro111:
        DW      108     ;SEQ10 FEED AGE1
        DW      109
     DW    00    ;end
;
Tbl1_Macro112:
        DW      108     ;SEQ11 FEED AGE1
        DW      076     ;125
        DW      117
        DW      119
```

```
      DW    00     ;end
;
Tbl1_Macro113:
        DW      108     ;SEQ12 FEED AGE1
        DW      108
        DW      109
      DW    00     ;end
;
Tbl1_Macro114:
        DW      108     ;SEQ13 REUSE 10 FOR14  FEED AGE1
        DW      115
        DW      001     ;20
      DW    00     ;end
;
Tbl1_Macro115:
        DW      108     ;SEQ15 FEED AGE1
        DW      076     ;125
        DW      117
        DW      119
        DW      00
;
Tbl1_Macro116:
        DW      108
        DW      108
        DW      109     ;SEQ1 FEED AGE1 ()
      DW    00     ;end
;
Tbl1_Macro117:                  ;WIERD SHIT SEE 101
        DW      108
        DW      120
        DW      109
      DW    00     ;end
;                   ;end-----------------AGE1
Tbl1_Macro118:
        DW      108
        DW      121
        DW      109     ;SEQ1 FEED AGE2
      DW    00     ;end
;
Tbl1_Macro119:
        DW      108
        DW      051     ;72
        DW      109     ;SEQ2 FEED AGE2
      DW    00     ;end
;
Tbl1_Macro120:
        DW      108
        DW      073     ;122
        DW      112
        DW      109     ;SEQ3 FEED AGE2
      DW    00     ;end
;
Tbl1_Macro121:
        DW      108
        DW      051     ;72
        DW      113
        DW      109     ;SEQ4 FEED AGE2
      DW    00     ;end
;
Tbl1_Macro122:
```

```
        DW      108
        DW      108
        DW      078     ;127     ;SEQ5 FEED AGE2
        DW      051     ;72
        DW      109
      DW    00    ;end
;
Tbl1_Macro123:
        DW      108
        DW      105     ;109
        DW      114     ;SEQ6 FEED AGE2
      DW    00    ;end
;
Tbl1_Macro124:
        DW      108
        DW      115
        DW      116
        DW      069     ;118   ;SEQ7 FEED AGE2
        DW      110
      DW    00    ;end
;
Tbl1_Macro125:
        DW      076     ;125
        DW      057     ;78
        DW      120
        DW      118     ;SEQ8 FEED AGE2
      DW    00    ;end
;
Tbl1_Macro126:
        DW      108
        DW      115     ;SEQ9 FEED AGE2
        DW      001     ;20
      DW    00    ;end
;
Tbl1_Macro127:
        DW      108
        DW      109     ;SEQ10 FEED AGE2
      DW    00    ;end

;
; Macro_grp2 was here


;
Tbl2_Macro128:
        DW      108
        DW      076     ;125
        DW      069     ;118
        DW      119     ;SEQ11 FEED AGE2
      DW    00    ;end
;
; Macro_grp2 was here

Tbl2_Macro129:
        DW      108
        DW      076     ;125
        DW      069     ;118
        DW      119     ;SEQ15 FEED AGE2
      DW    00    ;end
;               ------------------------END AGE2------------|
```

```
Tbl2_Macro130:
        DW      108
        DW      110
        DW      109     ;SEQ2 FEED AGE3
      DW    00    ;end
;
Tbl2_Macro131:
        DW      108
        DW      111
        DW      072 ;143
        DW      109     ;SEQ3 FEED AGE3
      DW    00    ;end
;
Tbl2_Macro132:
        DW      108
        DW      110
        DW      058 ;144
        DW      109     ;SEQ4 FEED AGE3
      DW    00    ;end
;
Tbl2_Macro133:
        DW      108
        DW      115
        DW      116
        DW      117
        DW      051     ;72     ;SEQ7 FEED AGE3
      DW    00    ;end
;
Tbl2_Macro134:
        DW      076     ;125
        DW      117
        DW      121
        DW      118     ;SEQ8 FEED AGE3
      DW    00    ;end
;
Tbl2_Macro135:
        DW      108
        DW      076     ;125
        DW      117     ;SEQ11 FEED AGE3
        DW      122
      DW    00    ;end
;       ---------------------------------
Tbl2_Macro136:
        DW      108
        DW      051     ;72
        DW      109
      DW    00    ;end
;
Tbl2_Macro137:
        DW      108
        DW      073     ;122
        DW      072     ;121
        DW      109
      DW    00    ;end
;
Tbl2_Macro138:
        DW      108
        DW      051     ;72
        DW      058     ;144
        DW      109
```

```
        DW      00      ;end
;
Tbl2_Macro139:
        DW      108
        DW      108
        DW      078     ;127
        DW      051     ;72
        DW      109
      DW      00      ;end
;
Tbl2_Macro140:
        DW      108     ;SEQ 6
        DW      105     ;109
        DW      123
      DW      00      ;end
;
Tbl2_Macro141:
        DW      108
        DW      115
        DW      116
        DW      057     ;78
        DW      051     ;72
      DW      00      ;end
;
Tbl2_Macro142:
        DW      076     ;125
        DW      069     ;118
        DW      121
        DW      118
      DW      00      ;end
;
Tbl2_Macro143:
        DW      108
        DW      125
        DW      057     ;78
        DW      122
      DW      00      ;end
;
Tbl2_Macro144:
        DW      108
        DW      125
        DW      057     ;78
        DW      122
      DW      00      ;end
;
Tbl2_Macro145:
        DW      108
        DW      121
        DW      109
      DW      00      ;end
;END FEED
;END GEORGE 07/05/98

;WAKE
;GEORGE 07/06/98
Tbl2_Macro146:    ;SG DONE
        DW      124     ;02
        DW      125
        DW      126
        DW      00      ;end
```

```
;
Tbl2_Macro147:    ;SG DONE
        DW      124
        DW      125
        DW      127
      DW    00    ;end
;
Tbl2_Macro148:    ;SG DONE
        DW      124
        DW      128
        DW      127
      DW    00    ;end
;
Tbl2_Macro149:    ;SG DONE
        DW      124
        DW      129
        DW      055     ;*08
      DW    00    ;end
;
Tbl2_Macro150:    ;SG DONE
        DW      124
        DW      130
        DW      131
        DW      132
      DW    00    ;end
;
Tbl2_Macro151:    ;SG DONE
        DW      124
        DW      130
        DW      131
        DW      123     ;*12
      DW    00    ;end
;
Tbl2_Macro152:    ;SG DONE
        DW      124
        DW      130
        DW      133
        DW      132
      DW    00    ;end
;
Tbl2_Macro153:    ;SG DONE
        DW      124
        DW      130
        DW      133
        DW      123     ;*12
      DW    00    ;end
;
Tbl2_Macro154:    ;SG DONE
        DW      124
        DW      134
        DW      135
        DW      131
      DW    00    ;end
;
Tbl2_Macro155:   ;SG DONE
        DW      124
        DW      134
        DW      136
        DW      131
      DW    00    ;end
```

```
;
Tbl2_Macro156:    ;SG DONE
        DW        124
        DW        134
        DW        135
        DW        133
      DW    00      ;end
;
Tbl2_Macro157:    ;SG DONE
        DW        124
        DW        134
        DW        136
        DW        137
        DW        133
      DW    00      ;end
;
Tbl2_Macro158:    ;SG DONE
        DW        124
        DW        138
        DW        139
      DW    00      ;end
;
Tbl2_Macro159:    ;SG DONE
        DW        124
        DW        140
     ;   DW        141
      DW    00      ;end
;
Tbl2_Macro160:    ;SG DONE
        DW        124
        DW        142
        DW        143
     ;   DW        141
      DW    00      ;end
;
Tbl2_Macro161:    ;SG DONE
        DW        124
        DW        144
        DW        145
        DW        146
        ,DW       141
      DW    00      ;end
;
Tbl2_Macro162:    ;SG DONE
        DW        124
        DW        147
        DW        141
      DW    00      ;end
;
Tbl2_Macro163:    ;SG DONE
        DW        124
        DW        148
      DW    00      ;end
;
Tbl2_Macro164:    ;SG DONE
        DW        124
        DW        053       ;29
        DW        149
        DW        150
        DW        00        ;end
```

```
;
Tbl2_Macro165:   ;SG DONE
        DW      124
        DW      151
      DW    00    ;end
;
Tbl2_Macro166:   ;SG DONE
        DW      124
        DW      152
        DW      131
        DW      153
        DW      154
      DW    00    ;end
;
Tbl2_Macro167:   ;SG DONE
        DW      124
        DW      152
        DW      155
        DW      153
        DW      154
      DW    00    ;end
;
Tbl2_Macro168:   ;SG DONE
        DW      124
        DW      152
        ;DW     153
        DW      131
        DW      156
        DW      154
      DW    00    ;end
;
Tbl2_Macro169:   ;SG DONE
        DW      124
        DW      053     ;*38
        DW      155
        DW      156
        DW      154
      DW    00    ;end
;END WAKE 07/06/98
;END GEORGE
;
;GEORGE 07/06/98
;HUNGER
Tbl2_Macro170:   ;SG DONE  ;HUNGER
        DW      159
        DW      165
        DW      412     ;DMH
      DW    00    ;end

Tbl2_Macro171:     ;SG DONE
        DW      160
        DW      165
        DW      412     ;DMH
      DW    00    ;end
;
Tbl2_Macro172:     ;SG DONE
        DW      160
      DW    00    ;end
;
Tbl2_Macro173:     ;SG DONE
```

```
        DW      168
        DW      159
        DW      165
        DW      412     ;DMH
      DW    00    ;end
;
Tbl2_Macro174:      ;SG DONE
        DW      168
        DW      160
        DW      165
        DW      412     ;DMH
      DW    00    ;end
;
 12_Macro175:       ;SG DONE
        DW      168
        DW      160
        DW      412     ;DMH
      DW    00    ;end
;
Tbl2_Macro176:      ;SG DONE
        DW      163
        DW      158
        DW      159
      DW    00    ;end
;
Tbl2_Macro177:      ;SG DONE
        DW      163
        DW      158
        DW      160
      DW    00    ;end
;
Tbl2_Macro178:      ;SG DONE
        DW      163
        DW      157
        DW      159
      DW    00    ;end
;
Tbl2_Macro179:      ;SG DONE
        DW      163
        DW      157
        DW      160
      DW    00    ;end
;
Tbl2_Macro180:      ;SG DONE
        DW      163
        DW      168

        DW      159
        DW      163
      DW    00    ;end
;
Tbl2_Macro181:      ;SG DONE
        DW      163
        DW      168
        DW      160
        DW      163
      DW    00    ;end
;
Tbl2_Macro182:      ;SG DONE
        DW      163
```

```
        DW      163
        DW      168
        DW      161
        DW      159
        DW      165
        DW      412     ;DMH
      DW    00    ;end
;
Tbl2_Macro183:      ;SG DONE
        DW      163
        DW      163
        DW      168
        DW      161
        DW      160
        DW      165
        DW      412     ;DMH
      DW    00    ;end
;
Tbl2_Macro184:      ;SG DONE
        DW      163
        DW      163
        DW      168
        DW      162
        DW      160
      DW    00    ;end
;
Tbl2_Macro185:      ;SG DONE
        DW      168
        DW      161
        DW      159
      DW    00    ;end
;
Tbl2_Macro186:      ;SG DONE
        DW      168
        DW      161
        DW      160
      DW    00    ;end
;
Tbl2_Macro187:      ;SG DONE
        DW      168
        DW      162
        DW      159
      DW    00    ;end
;
Tbl2_Macro188:      ;SG DONE
        DW      168
        DW      162
        DW      160
      DW    00    ;end
;
Tbl2_Macro189:      ;SG DONE
        DW      168
        DW      166
        DW      159
      DW    00    ;end
;
Tbl2_Macro190:      ;SG DONE
        DW      168
        DW      167
        DW      159
```

```
        DW      165
        DW      412     ;DMH
      DW    00    ;end
;
Tbl2_Macro191:        ;SG DONE
        DW      168
        DW      167
        DW      160
        DW      165
        DW      412     ;DMH
      DW    00    ;end
;
Tbl2_Macro192:        ;SG DONE
        DW      168
        DW      167
        DW      160
      DW    00    ;end
;
Tbl2_Macro193:        ;SG DONE
        DW      163
        DW      163
      DW    00    ;end
;
Tbl2_Macro194:        ;SC DONE
        DW      163
        DW      163
        DW      165
        DW      412     ; DMH
      DW    00    ;end
;
Tbl2_Macro195:        ;SG DONE
        DW      168
        DW      161
        DW      159
      DW    00    ;end
;
Tbl2_Macro196:        ;SG DONE
        DW      168
        DW      161
        DW      160
      DW    00    ;end
;
Tbl2_Macro197:        ;SG DONE
        DW      168
        DW      162
        DW      159
      DW    00    ;end
;
Tbl2_Macro198:        ;SG DONE
        DW      168
        DW      162
        DW      160
      DW    00    ;end
;
Tbl2_Macro199:        ;SG DONE
        DW      164
        DW      168
        DW      161
        DW      159
        DW      165
```

```
      DW    00    ;end
;
Tbl2_Macro200:          ;SG DONE
        DW      164
        DW      168     ;fs40
        DW      162
        DW      159
        DW      165
      DW    00    ;end
;
Tbl2_Macro201:          ;SG DONE
        DW      164
        DW      168     ;40
        DW      162
        DW      160
        DW      165
      DW    00    ;end
;
;END HUNGER
;END GEORGE 07/06/98
;
;
;INVERT
;GEORGE 07/07/98
Tbl2_Macro202:     ;SG DONE  ;INVERT
        DW      164     ;64
      DW    00    ;end
;
Tbl2_Macro203:     ;SG DONE
        DW      164     ;64
        DW      169
      DW    00    ;end
;
Tbl2_Macro204:      ;SG DONE
        DW      164     ;64
        DW      168     ;40
        DW      174
        DW      166
        DW      175
      DW    00    ;end
;
Tbl2_Macro205:     ;SG DONE
        DW      164     ;64
        DW      176
      DW    00    ;end
;
Tbl2_Macro206:     ;SG DONE
        DW      188
        DW      177
      DW    00    ;end
;
Tbl2_Macro207:    ;SG DONE
        DW      188
        DW      178
      DW    00    ;end
;
Tbl2_Macro208:    ;SG DONE
        DW      170
        DW      177
        DW      177
```

```
        DW      00      ;end
;
Tbl2_Macro209:      ;SG DONE
        DW      170
        DW      178
        DW      177
        DW      00      ;end
;
Tbl2_Macro210:      ;SG DONE
        DW      170
        DW      177
        DW      178
        DW      00      ;end
;
Tbl2_Macro211:      ;SG DONE
        DW      170
        DW      178
        DW      178
        DW      00      ;end
;
Tbl2_Macro212:      ;SG DONE
        DW      171
        DW      163     ;63
        DW      00      ;end
;
Tbl2_Macro213:      ;SG DONE
        DW      171
        DW      168     ;40
        DW      179
        DW      180
        DW      165     ;65
        DW      00      ;end
;
Tbl2_Macro214:      ;SG DONE
        DW      171
        DW      168     ;40
        DW      181
        DW      180
        DW      165     ;65
        DW      00      ;end
;
Tbl2_Macro215:      ;SG DONE
        DW      171
        DW      168
        DW      179
        DW      182
        DW      165     ;65
        DW      00      ;end
;
Tbl2_Macro216:      ;SG DONE
        DW      171
        DW      168     ;40
        DW      181
        DW      182
        DW      00      ;end
;
Tbl2_Macro217:      ;SG DONE
        DW      164     ;64
        DW      175
        DW      164     ;64
```

```
        DW      00      ;end
;
Tbl2_Macro218:   ;SG DONE
        DW      164     ;64
        DW      183
        DW      164     ;64
        DW      00      ;end
;
Tbl2_Macro219:   ;SG DONE
        DW      164     ;64
        DW      170
        DW      170
        DW      00      ;end
;
Tbl2_Macro220:   ;SG DONE
        DW      171
        DW      179
        DW      180
        DW      00      ;end
;
Tbl2_Macro221:   ;SG DONE
        DW      171
        DW      181
        DW      180
        DW      00      ;end
;
Tbl2_Macro222:   ;SG DONE
        DW      171
        DW      179
        DW      184
        DW      163     ;63
        DW      00      ;end
;
Tbl2_Macro223:   ;SG DONE
        DW      171
        DW      181
        DW      185
        DW      00      ;end
;
Tbl2_Macro224:   ;SG DONE
        DW      164     ;64
        DW      179
        DW      186
        DW      00      ;end
;
Tbl2_Macro225:   ;SG DONE
        DW      164     ;64
        DW      181
        DW      186
        DW      00      ;end
;
Tbl2_Macro226:   ;SG DONE
        DW      164     ;64
        DW      181
        DW      185
        DW      00      ;end
;
Tbl2_Macro227:   ;SG DONE
        DW      164     ;64
        DW      181
```

```
        DW      184
        DW      163     ;63
      DW    00    ;end
;
Tbl2_Macro228:  ;SG DONE
        DW      164     ;64
        DW      179
        DW      187
      DW    00    ;end
;
Tbl2_Macro229:  ;SG DONE
        DW      164     ;64
        DW      181
        DW      187
      DW    00    ;end
;
Tbl2_Macro230:   ;SG DONE
        DW      172
        DW      158
        DW      178
      DW    00    ;end
;
Tbl2_Macro231:   ;SG DONE
        DW      164     ;64
        DW      181
        DW      189
      DW    00    ;end
;
Tbl2_Macro232:  ;SG DONE
        DW      172
        DW      175
      DW    00    ;end
;
Tbl2_Macro233:  ;SG DONE
        DW      172
        DW      183
      DW    00    ;end
;
Tbl2_Macro234:  ;SG DONE
        DW      172
        DW      172
        DW      164     ;64
      DW    00    ;end
;
Tbl2_Macro235:  ;SG DONE
        DW      173
      DW    00    ;end
;
Tbl2_Macro236:   ;SG DONE
        DW      190
      DW    00    ;end
;
Tbl2_Macro237:  ;SG DONE
        DW      191
      DW    00    ;end
;
Tbl2_Macro238:  ;SG DONE
        DW      192
      DW    00    ;end
;END GEORGE 07/07/98
```

```
;END INVERT

;GEORGE 07/07/98
;BACK
Tbl2_Macro239:  ;BACKSG  ;SGDONE
        DW      193
        DW      193
      DW    00    ;end
;
Tbl2_Macro240:          ;SGDONE
        DW      193
        DW      194
        DW      195
      DW    00    ;end
;
Tbl2_Macro241:          ;SGDONE
        DW      193
        DW      196
        DW      195
      DW    00    ;end
;
Tbl2_Macro242:          ;SGDONE
        DW      193
        DW      194
        DW      197
      DW    00    ;end
;
Tbl2_Macro243:          ;SGDONE
        DW      193
        DW      196
        DW      197
      DW    00    ;end
;
Tbl2_Macro244:          ;SGDONE
        DW      198
        DW      199
        DW      200
        DW      201
      DW    00    ;end
;
Tbl2_Macro245:          ;SGDONE
        DW      198
        DW      199
        DW      202
        DW      201
      DW    00    ;end
;
Tbl2_Macro246:          ;SGDONE
        DW      198
        DW      199
        DW      200
        DW      184     ;148     ;212
        DW      00      ;end
;
Tbl2_Macro247:          ;SGDONE
        DW      198
        DW      199
        DW      202
        DW      184     ;148     ;212
      DW    00    ;end
```

```
;
Tbl2_Macro248:          ;SGDONE
        DW      198
        DW      198
      DW    00    ;end
;
Tbl2_Macro249:          ;SGDONE
        DW      198
        DW      203
        DW      204
      DW    00    ;end
;
Tbl2_Macro250:          ;SGDONE
        DW      198
        DW      205
        DW      206
        DW      207
        DW      204
      DW    00    ;end
;
Tbl2_Macro251:          ;SGDONE
        DW      198
        DW      205
        DW      208
        DW      233
        DW      204
      DW    00    ;end
;
Tbl2_Macro252:          ;SGDONE
        DW      198
        DW      205
        DW      206
        DW      233
        DW      204
      DW    00    ;end
;
Tbl2_Macro253:          ;SGDONE
        DW      198
        DW      209
        DW      210
        DW      00      ;end
;
Tbl2_Macro254:          ;SGDONE
        DW      198
        DW      209
        DW      211
        DW      212
        DW      213
      DW    00    ;end
;
Tbl2_Macro255:          ;SGDONE
        DW      198
        DW      209
        DW      214
      DW    00    ;end
;
Tbl3_Macro256:          ;SGDONE
        DW      198
        DW      215
        DW      216
```

```
        DW      217
        DW      00      ;end
;
Tbl3_Macro257:          ;SGDONE
        DW      198
        DW      215
        DW      216
        DW      218
      DW    00    ;end
;
Tbl3_Macro258:          ;SGDONE
        DW      219
        DW      220
        DW      209
        DW      217
        DW      199
        DW      234
      DW    00    ;end
;
Tbl3_Macro259:          ;SGDONE
        DW      219
        DW      220
        DW      209
        DW      205
        DW      217
        DW      234
        DW      00      ;end
;
Tbl3_Macro260:          ;SGDONE
        DW      219
        DW      220
        DW      209
        DW      205
        DW      218
        DW      234
      DW    00    ;end
;
Tbl3_Macro261:          ;SGDONE
        DW      221
        DW      222
      DW    00    ;end
;
Tbl3_Macro262:          ;SGDONE
        DW      221
        DW      223
        DW      222
      DW    00    ;end
;
Tbl3_Macro263:          ;SGDONE
        DW      198
        DW      224
        DW      199
      DW    00    ;end
;
Tbl3_Macro264:          ;SGDONE
        DW      198
        DW      224
        DW      205
      DW    00    ;end
;
```

```
Tbl3_Macro265:          ;SGDONE
        DW      198
        DW      225
        DW      205
      DW    00    ;end
;
Tbl3_Macro266:          ;SGDONE
        DW      226
        DW      201
      DW    00    ;end
;
Tbl3_Macro267:          ;SGDONE
        DW      198
        DW      227
        DW      227
        DW      228
        DW      229
      DW    00    ;end
;
Tbl3_Macro268:          ;SGDONE
        DW      198
        DW      227
        DW      227
        DW      230
        DW      229
      DW    00    ;end
;
Tbl3_Macro269:          ;SGDONE
        DW      198
        DW      194
        DW      199
      DW    00    ;end
;
Tbl3_Macro270:          ;SGDONE
        DW      198
        DW      194
        DW      205
      DW    00    ;end
;
Tbl3_Macro271:          ;SGDONE
        DW      198
        DW      196
        DW      205
      DW    00    ;end
;
Tbl3_Macro272:          ;SGDONE
        DW      198
        DW      235
        DW      231
        DW      199
      DW    00    ;end
;
Tbl3_Macro273:          ;SGDONE
        DW      198
        DW      235
        DW      231
        DW      205
      DW    00    ;end
;
Tbl3_Macro274:          ;SGDONE
```

```
        DW      198
        DW      235
        DW      232
        DW      205
      DW    00    ;end
;
Tbl3_Macro275:          ;SGDONE
        DW      198
        DW      236
        DW      232
        DW      205
      DW    00    ;end
;END GEORGE 07/07/98
;END BACK
;
;GEORGE 07/08/98
;SICK


Tbl3_Macro276:   ;SG DONE  ;SICK3
        DW      237
        DW      168     ;135     ;40
        DW      117     ;41
        DW      238
      DW    00    ;end
;
Tbl3_Macro277:   ;SG DONE
        DW      237
        DW      168     ;135     ;40
        DW      239
        DW      238
      DW    00    ;end
;
Tbl3_Macro278:   ;SG DONE
        DW      237
        DW      168     ;135     ;40
        DW      117     ;41
        DW      240
      DW    00    ;end
;
Tbl3_Macro279:   ;SG DONE
        DW      237
        DW      53      ;45
        DW      239
        DW      240
      DW    00    ;end
;
Tbl3_Macro280:   ;SG DONE
        DW      237
        DW      241
      DW    00    ;end
;
Tbl3_Macro281:   ;SG DONE
        DW      237
        DW      242
      DW    00    ;end
;
Tbl3_Macro282:   ;SG DONE
        DW      237
        DW      243
```

```
        DW      244
      DW    00    ;end
;
Tbl3_Macro283:   ;SG DONE
        DW      250
        DW      117     ;41
        DW      245
      DW    00    ;end
;
Tbl3_Macro284:   ;SG DONE
        DW      250
        DW      239
        DW      245
      DW    00    ;end
;
Tbl3_Macro285:   ;SG DONE
        DW      250
        DW      239
        DW      182     ;51
        DW      00      ;end
;
Tbl3_Macro286:   ;SG DONE
        DW      237
        DW      246
        DW      250
      DW    00    ;end
;
Tbl3_Macro287:   ;SG DONE
        DW      237
        DW      247
        DW      250
      DW    00    ;end
;
Tbl3_Macro288:   ;SG DONE
        DW      237
      DW    00    ;end
;
Tbl3_Macro289:   ;SG DONE
        DW      237
        DW      248
        DW      250
      DW    00    ;end
;
Tbl3_Macro290:   ;SG DONE
        DW      237
        DW      249
        DW      00      ;end
;
Tbl3_Macro291:   ;SG DONE
        DW      250
        DW      250
      DW    00    ;end
;
Tbl3_Macro292:   ;SG DONE
        DW      250
        DW      248
      DW    00    ;end
;END SICK
;END GEORGE 07/08/98
```

```
;GEORGE 07/08/98
;LIGHT
Tbl3_Macro293:
        DW      251
        DW      00      ;end  RB
;
;Tbl1_Macro294:
;        DW      263
;        DW      00      ;end    RB
;
Tbl3_Macro294:
        DW      252
        DW      00      ;end      RB
;
Tbl3_Macro295:
        DW      253
        DW      00      ;end        RB
;
Tbl3_Macro296:
        DW      254
        DW      00      ;end          RB
;
Tbl3_Macro297:
        DW      255
        DW      00      ;end            RB
;
Tbl3_Macro298:
        DW      256
      DW    00    ;end
;
Tbl3_Macro299:
        DW      257
        DW      00      ;end
;
Tbl3_Macro300:
        DW      258
      DW    00    ;end
;
Tbl3_Macro301:
        DW      259
      DW    00    ;end
;
Tbl3_Macro302:
        DW      260
      DW    00    ;end
;
Tbl3_Macro303:
        DW      261
      DW    00    ;end
;
Tbl3_Macro304:
        DW      262
      DW    00    ;end
;
Tbl3_Macro305:
        DW      263
      DW    00    ;end
;
Tbl3_Macro306:
        DW      264
```

```
        DW      00      ;end
;
Tbl3_Macro307:
        DW      265
        DW      00      ;end
;END GEORGE 07/08/98
;END LIGHT
;GEORGE 07/08/98
;DARK

Tbl3_Macro308:
        DW      266
        DW      00      ;end
;
Tbl3_Macro309:
        DW      267
        DW      00      ;end
;
Tbl3_Macro310:
        DW      268
        DW      00      ;end
;
Tbl3_Macro311:
        DW      269
        DW      00      ;end
;
Tbl3_Macro312:
        DW      270
        DW      00      ;end
;
Tbl3_Macro313:
        DW      271
        DW      00      ;end
;
Tbl3_Macro314:
        DW      272
        DW      00      ;end
;
Tbl3_Macro315:
        DW      273
        DW      00      ;end
;
Tbl3_Macro316:
        DW      274
        DW      00      ;end
;
Tbl3_Macro317:
        DW      275
        DW      00      ;end
;
Tbl3_Macro318:
        DW      276
        DW      00      ;end
;
Tbl3_Macro319:
        DW      277
        DW      00      ;end
;
Tbl3_Macro320:
        DW      278
```

```
      DW    00    ;end
;
Tbl3_Macro321:
        DW      279
      DW    00    ;end
;
Tbl3_Macro322:
        DW      280
      DW    00    ;end
;
Tbl3_Macro323:
        DW      281
      DW    00    ;end
;
Tbl3_Macro324:
        DW      282
      DW    00    ;
;
Tbl3_Macro325:
        DW      283
      DW    00    ;end
;
Tbl3_Macro326:
        DW      284
      DW    00    ;end
;
Tbl3_Macro327:
        DW      285
      DW    00    ;end
;
Tbl3_Macro328:
        DW      286
      DW    00    ;end
;
Tbl3_Macro329:
        DW      287
      DW    00    ;end
;
Tbl3_Macro330:
        DW      288
      DW    00    ;end
;
Tb  _Macro331:
        DW      289
      DW    00    ;end
;END DARK
;END GEORGE 07/08/98

;GEORGE 07/08/98
;SOUND
;
Tbl3_Macro332:
        DW      290       ;S1-A1/S9-A1/S1-A2 SOUND js
      DW    00    ;end
;
Tbl3_Macro333:
        DW      291       ;S2-A1/S10-A1/S2-A2 SOUND js
      DW    00    ;end
;
Tbl3_Macro334:
```

```
        DW      292     ;S3-A1/S11-A1 SOUND js
      DW    00    ;end
;
Tbl3_Macro335:
        DW      293     ;S4-A1/S12-A1 SOUND js
      DW    00    ;end
;
Tbl3_Macro336:
        DW      310
        DW      294     ;S5-A1/S13-A1 SOUND (with say/m2) js
      DW    00    ;end
;
Tbl3_Macro337:
        DW      295     ;S6-A1/S14-A1 SOUND js
      DW    00    ;end
;
Tbl3_Macro338:
        DW      310
        DW      296     ;S7-A1/S15-A1 SOUND (with say/m2) js
      DW    00    ;end
;
Tbl3_Macro339:
        DW      297     ;S8-A1/S16-A1 SOUND js
      DW    00    ;end
;
Tbl3_Macro340:
        DW      298     ;S3-A2 SOUND js
      DW    00    ;end
;
Tbl3_Macro341:
        DW      299     ;S4-A2 SOUND js
      DW    00    ;end
;
Tbl3_Macro342:
        DW      310
        DW      300     ;S5-A2 SOUND (with say/m2) js
      DW    00    ;end
;
Tbl3_Macro343:
        DW      310
        DW      301     ;S7-A2 SOUND (with say/m2) js
      DW    00    ;end
;
Tbl3_Macro344:
        DW      302     ;S8-A2 SOUND js
      DW    00    ;end
;
Tbl3_Macro345:
        DW      303     ;S3-A3 SOUND js
      DW    00    ;end
;
Tbl3_Macro346:
        DW      304     ;S4-A3 SOUND js
        DW      00     ;end
;
Tbl3_Macro347:
        DW      310
        DW      305     ;S7-A3 SOUND (with say/m2) js
     DW    00    ;end
;
```

```
Tbl3_Macro348:
        DW      306         ;S1-A4 SOUND js
      DW    00      ;end
;
Tbl3_Macro349:
        DW      307         ;S3-A4 SOUND js
      DW    00      ;end
;
Tbl3_Macro350:
        DW      308         ;S6-A4 SOUND js
      DW    00      ;end
;
Tbl3_Macro351:
        DW      309         ;S8-A4 SOUND js
      DW    00      ;end
;
;END GEORGE 07/08/98
;END SOUND
;
;
;TILT
;GEORGE 07/09/98
Tbl3_Macro352:
        DW      310         ;S1 A1 TILT/S4 A1 TILT js
      DW    00      ;end
;
Tbl3_Macro353:
        DW      311         :S2 A1 TILT js
      DW    00      ;end
;
Tbl3_Macro354:
        DW      312         ;S3 A1 TILT js
      DW    00      ;end
;
Tbl3_Macro355:
        DW      313         ;S5 A1 TILT js
      DW    00      ;end
;
Tbl3_Macro356:
        DW      314         ;S6 A1 TILT js
      DW    00      ;end
;
Tbl3_Macro357:
        DW      315         ;S7 A1 TILT js
      DW    00      ;end
;
Tbl3_Macro358:
        DW      313         ;S8 A1 TILT js
        DW      316
      DW    00      ;end
;
Tbl3_Macro359:
        DW      317         ;S9 A1 TILT js
      DW    00      ;end
;
Tbl3_Macro360:
        DW      318         ;S10 A1 TILT js
      DW    00      ;end
;
Tbl3_Macro361:
```

```
        DW      310     ;S11 A1 TILT js
        DW      319
      DW    00    ;end
;
Tbl3_Macro362:
        DW      320     ;S12 A1 TILT js
      DW    00    ;end
;
Tbl3_Macro363:
        DW      321     ;S13 A1 TILT js
      DW    00    ;end
;
Tbl3_Macro364:
        DW      322     ;S15 A1 TILT js
      DW    00    ;end
;
Tbl3_Macro365:
        DW      323     ;S16 A1 TILT js
      DW    00    ;end
;
Tbl3_Macro366:
        DW      324     ;S1 A1 TILT js
      DW    00    ;end
;
Tbl3_Macro367:
        DW      324
        DW      325     ;S2 A1 TILT js
      DW    00    ;end
;
Tbl3_Macro368:
        DW      326     ;S5 A2 TILT js
      DW    00    ;end
;
Tbl3_Macro369:
        DW      313
        DW      327     ;S7 A2 TILT js
      DW    00    ;end
;
Tbl3_Macro370:
        DW      313
        DW      328     ;S8 A2 TILT js
      DW    00    ;end
;
Tbl3_Macro371:
        DW      310
        DW      329     ;S11 A2 TILT js
      DW    00    ;end
;
Tbl3_Macro372:
        DW      330     ;S12 A2 TILT js
      DW    00    ;end
;
Tbl3_Macro373:
        DW      313
        DW      331     ;S13 A2 TILT js
      DW    00    ;end
;
Tbl3_Macro374:
        DW      332     ;S12 A2 TILT js
      DW    00    ;end
```

```
;
Tbl3_Macro375:
        DW      333
      DW    00    ;end
;
Tbl3_Macro376:
        DW      334
      DW    00    ;end
;
Tbl3_Macro377:
        DW      334
        DW      335
      DW    00    ;end
;
Tbl3_Macro378:
        DW      336
      DW    00    ;end
;
Tbl3_Macro379:
        DW      313
        DW      337
      DW    00    ;end
;
Tbl3_Macro380:
        DW      313
        DW      338
      DW    00    ;end
;
Tbl3_Macro381:
        DW      339
      DW    00    ;end
;
Tbl3_Macro382:
        DW      317
        DW      340
      DW    00    ;end
;
Tbl3_Macro383:
        DW      341
      DW    00    ;end
;
Tbl4_Macro384:
        DW      310
        DW      329
        DW      342
      DW    00    ;end
;
Tbl4_Macro385:
        DW      313
        DW      343
      DW    00    ;end
;
Tbl4_Macro386:
        DW      313
        DW      344
      DW    00    ;end
;
Tbl4_Macro387:
        DW      334
        DW      345
```

```
      DW    00    ;end
;
Tbl4_Macro388:
        DW      346
      DW    00    ;end
;
Tbl4_Macro389:
        DW      313
        DW      347
      DW    00    ;end
;
Tbl4_Macro390:
        DW      310
        DW      348
      DW    00    ;end
;
Tbl4_Macro391:
        DW      313
        DW      349
      DW    00    ;end
;
Tbl4_Macro392:
        DW      313
        DW      350
      DW    00    ;end
;END TILT
;END GEORGE 07/09/98
;
;IR
;GEORGE 07/09/98
Tbl4_Macro393:
        DW      351
      DW    00    ;end
;
;
Tbl4_Macro394:
        DW      352       seq5, IR age1
      DW    00    ;end
;
Tbl4_Macro395:
        DW      353         seq6, IR age1
        DW      354
      DW    00    ;end
;
Tbl4_Macro396:
        DW      356         ;seq7 ir age1
        DW      355
      DW    00    ;end
;
Tbl4_Macro397:
        DW      357       ;seq8 ir age1
      DW    00    ;end
;
Tbl4_Macro398:
        DW      358       ;seq9 ir age1
      DW    00    ;end
;
Tbl4_Macro399:
        DW      359       ;seq        10,360 ir age1
      DW    00    ;end
```

```
;
Tbl4_Macro400:
        DW      360       ;seq12 ir age1,age2,age,3
      DW    00    ;end
;
Tbl4_Macro401:
        DW      361       ;seq13,14 ir age1
      DW    00    ;end
;
Tbl4_Macro402:
        DW      362        ;seq15 ir age1
      DW    00    ;end
;
Tbl4_Macro403:
        DW      363       ;seq16 ir age1
      DW    00    ;end
;
Tbl4_Macro404:
        DW      364       ;seq1,2,3 ir age2
      DW    00    ;end
;
Tbl4_Macro405:
        DW      365       ;seq4,5 ir age2
      DW    00    ;end
;
Tbl4_Macro406:
        DW      366       ;seq6 ir age2
      DW    00    ;end
;
Tbl4_Macro407:
        DW      367        ;seq7,8 ir age 2
      DW    00    ;end
;
Tbl4_Macro408:
        DW      368        ;seq9 ir age2
      DW    00    ;end
;
Tbl4_Macro409:
        DW      369        ;seq10 ir age2
      DW    00    ;end
;
Tbl4_Macro410:
        DW      370       ;seq11 ir age2
      DW    00    ;end
;
Tbl4_Macro411:
        DW      371       ;seq13,14 ir age2
      DW    00    ;end
;
Tbl4_Macro412:
        DW      372     ;seq15 ir age2
      DW    00    ;end
;
Tbl4_Macro413:
        DW      373       ;seq16 ir age2
      DW    00    ;end
;
Tbl4_Macro414:
        DW      374       ;seq1,2,3,4,5 ir age3
      DW    00    ;end
```

```
;
Tbl4_Macro415:
        DW      375         ;seq6 ir age3
      DW    00    ;end
;
Tbl4_Macro416:
        DW      376         ;seq7,8 ir age3
      DW    00    ;end
;
Tbl4_Macro417:
        DW      377         ;seq9 ir age3
      DW    00    ;end
;
Tbl4_Macro418:
        DW      378         ;seq11 ir age3
      DW    00    ;end
;
Tbl4_Macro419:
        DW      379         ;seq13,14 ir age3
      DW    00    ;end
;
Tbl4_Macro420:
        DW      380         ;seq15 ir age3
      DW    00    ;end
;
Tbl4_Macro421:
        DW      381         ;seq1,2,3,4,5 ir age4
      DW    00    ;end
;
Tbl4_Macro422:
        DW      382         ;seq6 ir age4
      DW    00    ;end
;
Tbl4_Macro423:
        DW      383         ;seq7,8 ir age4
      DW    00    ;end
;
Tbl4_Macro424:
        DW      384         ;seq9 ir age4
      DW    00    ;end
;
Tbl4_Macro425:
        DW      385         ;seq10 ir age4
      DW    00    ;end
;
Tbl4_Macro426:
        DW      386         ;seq11 ir age4
      DW    00    ;end
;
Tbl4_Macro427:
        DW      387         ;seq12 ir age4
      DW    00    ;end
;
Tbl4_Macro428:
        DW      389
        DW      388         ;seq14 ir age4
        DW      389
        DW  00    ;end
;
Tbl4_Macro429:
```

```
        DW      389       ;seq15 ir age4
        DW      390
      DW    00     ;end
;END GEORGE
;END IR
;

; START FURBY SAYS DMH
Tbl4_Macro430:
      DW    50       ; TICKLE
      DW    00     ;end
;
Tbl4_Macro431:
      DW    196      ; PET
      DW    00     ;end
;
Tbl4_Macro432:
      DW    71       ; SOUND
      DW    00     ;end
;
Tbl4_Macro433:
      DW    391      ; LIGHT
      DW    00     ;end
;
Tbl4_Macro434:
      DW    198      ; soft purr
      DW    00     ;end
;
Tbl4_Macro435:
      DW    392      ; no light
      DW    00     ;end
;
Tbl4_Macro436:
      DW    393      ; loud sound
      DW    00     ;end
;
Tbl4_Macro437:
      DW    115      ; burp (hide and seek)
      DW    00     ;end
;
Tbl4_Macro438:
      DW    116      ; sigh (hide and seek)
      DW    00     ;end
;
Tbl4_Macro439:          ; win sound (dmh)
        dw      376
        dw      376
        dw      367
      DW    00     ;end
; END FURBY SAYS DMH
;
;
; start diagnostic tables
Tbl4_Macro440:                  ; start diagnostic beeps
      DW    400
      DW    00     ;end
;
Tbl4_Macro441:                  ; press key beep
      DW    401
      DW    00     ;end
```

```
;
Tbl4_Macro442:                          ; pass beep
      DW    402
      DW    00      ;end
;
Tbl4_Macro443:
      DW    403               ; fail beep
      DW    00      ;end
;
Tbl4_Macro444:                          ; speaker test tone
      DW    404
      DW    00      ;end
;
Tbl4_Macro445:                          ; motor cal
      DW    405
      DW    00      ;end
;
Tbl4_Macro446:                          ; feed1
      DW    406
      DW    00      ;end
;
Tbl4_Macro447:                          ; feed2
      DW    407
      DW    00      ;end
;
Tbl4_Macro448:                          ; light
      DW    408
      DW    00      ;end
;
Tbl4_Macro449:                          ; sound
      DW    409
      DW    00      ;end
;
Tbl4_Macro450:                          ; go to sleep
      DW    410
      DW    00      ;end
;
;end of diagnostic tables dmh
;
Tbl4_Macro451:
        DW      117     ; HIDE AND SEEK SOUND DHM
      DW    00      ;end
;
Tbl4_Macro452:
        DW      118     ; HIDE AND SEEK SOUND DHM
      DW    00      ;end
;
Tbl4_Macro453:
        DW      399     ; delay
        DW      395     ; ME DHM
        DW      110     ; NAME "KOKO" DMH
      DW    00      ;end
;
Tbl4_Macro454:
        DW      399     ; delay
        DW      395     ; ME DHM
        DW      396     ; NAME "MEME" DMH
      DW    00      ;end
;
Tbl4_Macro455:
```

```
        DW      399     ; delay
        DW      395     ; ME
        DW      112     ; NAME "E-DAY" DMH
      DW    00    ;end
;
Tbl4_Macro456:
        DW      399     ; delay
        DW      395     ; ME
        DW      397     ; NAME "DO-MOH" DMH
        DW      00      ;end
Tbl4_Macro457:
        DW      399     ; delay
        DW      395     ; ME
        DW      114     ; NAME "TO-TYE" DMH
      DW    00    ;end
;
Tbl4_Macro458:
        DW      399     ; delay
        DW      395     ; ME
        DW      117     ; NAME "BOO" DHM
      DW    00    ;end
;
Tbl4_Macro459:
        DW      399     ; delay
        DW      395     ; ME
        DW      398     ; NAME "TOH-LOO" DHM
      DW    00    ;end
;
Tbl4_Macro460:
        DW      399     ; delay
        DW      395     ; ME
        DW      120     ; NAME "A-TAY" DHM
      DW    00    ;end
;
Tbl4_Macro461:
        DW      399     ; delay
        DW      395     ; ME
        DW      131     ; NAME "WAY-LOH" DMH
      DW    00    ;end
;
Tbl4_Macro462:
        DW      399     ; delay
        DW      395     ; ME
        DW      143     ; NAME "U-TYE"
      DW      00
;
Tbl4_Macro463:
        DW      399     ; delay
        DW      395     ; ME
        DW      145     ; NAME "A-LOH" DMH
      DW    00    ;end
;
Tbl4_Macro464:
        DW      399     ; delay
        DW      395     ; ME
        DW      152     ; NAME "KA" DMH
      DW    00    ;end
;
Tbl4_Macro465:
        DW      399     ; delay
```

```
        DW      395     ; ME
        DW      166     ; NAME "DAH" DMH
      DW    00    ;end
;
Tbl4_Macro466:
        DW      399     ; delay
        DW      3..     ; ME
        DW      175     ; NAME "BOH-BAY" DMH
      DW    00    ;end
;
Tbl4_Macro467:
        DW      399     ; delay
        DW      395     ; ME
        DW      177     ; NAME "NAH-BAH" DMH
      DW    00    ;end
;
Tbl4_Macro468:
        DW      129     ; dodle do, me love you DMH
        DW      129
        DW      151
      DW    00    ;end
;
Tbl4_Macro469:          ; SING A SONG DMH

        DW      219
        DW      220
;       DW      219
;       DW      220
;       DW      219
;       DW      220
      DW    00    ;end
;
Tbl4_Macro470:          ; BURB ATTACK DMH
        DW      115
        DW      115
        DW      115
        DW      115
        DW      115
        DW      115
        DW      115
        DW      115
      DW    00    ;end
;
Tbl4_Macro471:          ; WIN SOUND DMH
        DW      313
        DW      338
        DW      376
      DW    00    ;end
;
Tbl4_Macro472:
      DW    46
      DW    00    ;end
;
Tbl4_Macro473:          ; ME DONE (DMH)
        DW      53
        DW      123
      DW    00    ;end
;
Tbl4_Macro474:          ; LISTEN ME  (DMH)
      DW    394
```

```
        DW      53
      DW    00      ;end
;
Tbl4_Macro475:
      DW    411
      DW    00      ;end
;
Tbl4_Macro476:
        DW      399     ; delay
        DW      395     ; ME
        DW      186     ; NAME "LOO-LOO" DMH
      DW    00      ;end
;
Tbl4_Macro477:
        DW      399     ; delay
        DW      395     ; ME
        DW      194     ; NAME "AH-MAY" DMH
      DW    00      ;end
;
Tbl4_Macro478:
        DW      399     ; delay
        DW      395     ; ME
        DW      201     ; NAME "NOO-LOO" DMH
      DW    00      ;end
;
Tbl4_Macro479:
        DW      399     ; delay
        DW      395     ; ME
        DW      208     ;   ME "MAY-MAY    H
      DW    00      ;end
;
Tbl4_Macro480:
        DW      399     ; delay
        DW      395     ; ME
        DW      224     ; NAME "MAY-LAH" DMH
      DW    00      ;end
;
Tbl4_Macro481:
        DW      399     ; delay
        DW      395     ; ME
        DW      228     ; DAH-NOH-LAH
      DW    00      ;end
;
Tbl4_Macro482:
        DW      399     ; delay
        DW      395     ; ME
        DW      398     ; NAME "TOH-LOO-KAH" DMH
        DW      152     ;
      DW    00      ;end
;
Tbl4_Macro483:
        DW      399     ; delay
        DW      395     ; ME
        DW      152     ; KA-DA
        DW      166
      DW    00      ;end
;
Tbl4_Macro484:
        DW      399     ; delay
        DW      395     ; ME
```

```
        DW      224     ; MAY-LAH-KA
        DW      152
      DW    00    ;end
;
Tbl4_Macro485:
      DW    4
      DW    00    ;end
;
Tbl4_Macro486:
      DW    4
      DW    00    ;end
;
Tbl4_Macro487:
      DW
      DW    00    ;end
;
Tbl4_Macro488:
      DW    4
      DW    00    ;end
;
Tbl4_Macro489:
      DW    4
      DW    00    ;end
;
Tbl4_Macro490:
      DW    4
      DW    00    ;end
;
Tbl4_Macro491:
      DW    4
      DW    00    ;end
;
Tbl4_Macro492:
      DW    4
      DW    00    ;end
;
Tbl4_Macro493:
      DW    4
      DW    00    ;end
;
Tbl4_Macro494:
      DW    4
      DW    00    ;end
;
Tbl4_Macro495:
      DW    4
      DW    00    ;end
;
Tbl4_Macro496:
      DW    4
      DW    00    ;end
;
Tbl4_Macro497:
      DW    4
      DW    00    ;end
;
Tbl4_Macro498:
      DW    4
      DW    00    ;end
;
```

```
Tbl4_Macro499:
      DW    4
      DW    00    ;end
;
Tbl4_Macro500:
      DW    4
      DW    00    ;end
;
Tbl4_Macro501:
      DW    4
      DW    00    ;end
;
Tbl4_Macro502:
      DW    4
      DW    00    ;end
;
Tbl4_Macro503:
      DW    4
      DW    00    ;end
;
Tbl4_Macro504:
      DW    4
      DW    00    ;end
;
Tbl4_Macro505:
      DW    4
      DW    00    ;end
;
Tbl4_Macro506:
      DW    4
      DW    00    ;end
;
Tbl4_Macro507:
      DW    4
      DW    00    ;end
;
Tbl4_Macro508:
      DW    4
      DW    00    ;end
;
Tbl4_Macro509:
      DW    4
      DW    00    ;end
;
Tbl4_Macro510:
      DW    4
      DW    00    ;end
;
Tbl4_Macro511:
      DW    4
      DW    00    ;end
;



;***********************************************
;***********************************************
;***********************************************
```

```
;
;
;***********************************************
;***********************************************
;***********************************************
;***********************************************
;***********************************************

;;;;;;;  SAYSENT pointer tables (128 max per table ---- 255 tables max)

Spch_grp1:

        DW      Tbl1_say000
        DW
Tbl1_say001,Tbl1_say002,Tbl1_say003,Tbl1_say004,Tbl1_say005
        DW
Tbl1_say006,Tbl1_say007,Tbl1_say008,Tbl1_say009,Tbl1_say010
        DW
Tbl1_say011,Tbl1_say012,Tbl1_say013,Tbl1_say014,Tbl1_say015
        DW
Tbl1_say016,Tbl1_say017,Tbl1_say018,Tbl1_say019,Tbl1_say020
        DW
Tbl1_say021,Tbl1_say022,Tbl1_say023,Tbl1_say024,Tbl1_say025
        DW
Tbl1_say026,Tbl1_say027,Tbl1_say028,Tbl1_say029,Tbl1_say030
        DW
Tbl1_say031,Tbl1_say032,Tbl1_say033,Tbl1_say034,Tbl1_say035
        DW
Tbl1_say036,Tbl1_say037,Tbl1_say038,Tbl1_say039,Tbl1_say040
        DW
Tbl1_say041,Tbl1_say042,Tbl1_say043,Tbl1_say044,Tbl1_say045
        DW
Tbl1_say046,Tbl1_say047,Tbl1_say048,Tbl1_say049,Tbl1_say050
        DW
Tbl1_say051,Tbl1_say052,Tbl1_say053,Tbl1_say054,Tbl1_say055
        DW
Tbl1_say056,Tbl1_say057,Tbl1_say058,Tbl1_say059,Tbl1_say060
        DW
Tbl1_say061,Tbl1_say062,Tbl1_say063,Tbl1_say064,Tbl1_say065
        DW
Tbl1_say066,Tbl1_say067,Tbl1_say068,Tbl1_say069,Tbl1_say070
        DW
Tbl1_say071,Tbl1_say072,Tbl1_say073,Tbl1_say074,Tbl1_say075
        DW
Tbl1_say076,Tbl1_say077,Tbl1_say078,Tbl1_say079,Tbl1_say080
        DW
Tbl1_say081,Tbl1_say082,Tbl1_say083,Tbl1_say084,Tbl1_say085
        DW
Tbl1_say086,Tbl1_say087,Tbl1_say088,Tbl1_say089,Tbl1_say090
        DW
Tbl1_say091,Tbl1_say092,Tbl1_say093,Tbl1_say094,Tbl1_say095
        DW      Tbl1_say096,Tbl1_say097,Tbl1_say098,Tbl1_say099
        DW
Tbl1_say100,Tbl1_say101,Tbl1_say102,Tbl1_say103,Tbl1_say104
      DW    Tbl1_say105,Tbl1_say106,Tbl1_say107,Tbl1_say108,Tbl1_say109
      DW    Tbl1_say110,Tbl1_say111,Tbl1_say112,Tbl1_say113,Tbl1_say114
      DW    Tbl1_say115,Tbl1_say116,Tbl1_say117,Tbl1_say118,Tbl1_say119
      DW    Tbl1_say120,Tbl1_say121,Tbl1_say122,Tbl1_say123,Tbl1_say124
      DW    Tbl1_say125,Tbl1_say126,Tbl1_say127
```

```
;
Spch_grp2:

        DW      Tbl2_say128
        DW      Tbl2_say129,Tbl2_say130,Tbl2_say131,Tbl2_say132,Tbl2_say133
        DW      Tbl2_say134,Tbl2_say135,Tbl2_say136,Tbl2_say137,Tbl2_say138
        DW      Tbl2_say139,Tbl2_say140,Tbl2_say141,Tbl2_say142,Tbl2_say143
        DW      Tbl2_say144,Tbl2_say145,Tbl2_say146,Tbl2_say147,Tbl2_say148
        DW      Tbl2_say149,Tbl2_say150,Tbl2_say151,Tbl2_say152,Tbl2_say153
        DW      Tbl2_say154,Tbl2_say155,Tbl2_say156,Tbl2_say157,Tbl2_say158
        DW      Tbl2_say159,Tbl2_say160,Tbl2_say161,Tbl2_say162,Tbl2_say163
        DW      Tbl2_say164,Tbl2_say165,Tbl2_say166,Tbl2_say167,Tbl2_say168
        DW      Tbl2_say169,Tbl2_say170,Tbl2_say171,Tbl2_say172,Tbl2_say173
        DW      Tbl2_say174,Tbl2_say175,Tbl2_say176,Tbl2_say177,Tbl2_say178
        DW      Tbl2_say179,Tbl2_say180,Tbl2_say181,Tbl2_say182,Tbl2_say183
        DW      Tbl2_say184,Tbl2_say185,Tbl2_say186,Tbl2_say187,Tbl2_say188
        DW      Tbl2_say189,Tbl2_say190,Tbl2_say191,Tbl2_say192,Tbl2_say193
        DW      Tbl2_say194,Tbl2_say195,Tbl2_say196,Tbl2_say197,Tbl2_say198
        DW      Tbl2_say199,Tbl2_say200,Tbl2_say201,Tbl2_say202,Tbl2_say203
        DW      Tbl2_say204,Tbl2_say205,Tbl2_say206,Tbl2_say207,Tbl2_say208
        DW      Tbl2_say209,Tbl2_say210,Tbl2_say211,Tbl2_say212,Tbl2_say213
        DW      Tbl2_say214,Tbl2_say215,Tbl2_say216,Tbl2_say217,Tbl2_say218
        DW      Tbl2_say219,Tbl2_say220,Tbl2_say221,Tbl2_say222,Tbl2_say223
        DW      Tbl2_say224,Tbl2_say225,Tbl2_say226,Tbl2_say227,Tbl2_say228
        DW      Tbl2_say229,Tbl2_say230,Tbl2_say231,Tbl2_say232,Tbl2_say233
        DW      Tbl2_say234,Tbl2_say235,Tbl2_say236,Tbl2_say237,Tbl2_say238
        DW      Tbl2_say239,Tbl2_say240,Tbl2_say241,Tbl2_say242,Tbl2_say243
        DW      Tbl2_say244,Tbl2_say245,Tbl2_say246,Tbl2_say247,Tbl2_say248
        DW      Tbl2_say249,Tbl2_say250,Tbl2_say251,Tbl2_say252,Tbl2_say253
        DW      Tbl2_say254,Tbl2_say255
;
;
Spch_grp3:

          DW        Tbl3_say256
        DW      Tbl3_say257,Tbl3_say258,Tbl3_say259,Tbl3_say260,Tbl3_say261
        DW      Tbl3_say262,Tbl3_say263,Tbl3_say264,Tbl3_say265,Tbl3_say266
        DW      Tbl3_say267,Tbl3_say268,Tbl3_say269,Tbl3_say270,Tbl3_say271
        DW      Tbl3_say272,Tbl3_say273,Tbl3_say274,Tbl3_say275,Tbl3_say276
        DW      Tbl3_say277,Tbl3_say278,Tbl3_say279,Tbl3_say280,Tbl3_say281
        DW      Tbl3_say282,Tbl3_say283,Tbl3_say284,Tbl3_say285,Tbl3_say286
        DW      Tbl3_say287,Tbl3_say288,Tbl3_say289,Tbl3_say290,Tbl3_say291
        DW      Tbl3_say292,Tbl3_say293,Tbl3_say294,Tbl3_say295,Tbl3_say296
        DW      Tbl3_say297,Tbl3_say298,Tbl3_say299,Tbl3_say300,Tbl3_say301
        DW      Tbl3_say302,Tbl3_say303,Tbl3_say304,Tbl3_say305,Tbl3_say306
        DW      Tbl3_say307,Tbl3_say308,Tbl3_say309,Tbl3_say310,Tbl3_say311
        DW      Tbl3_say312,Tbl3_say313,Tbl3_say314,Tbl3_say315,Tbl3_say316
        DW      Tbl3_say317,Tbl3_say318,Tbl3_say319,Tbl3_say320,Tbl3_say321
        DW      Tbl3_say322,Tbl3_say323,Tbl3_say324,Tbl3_say325,Tbl3_say326
        DW      Tbl3_say327,Tbl3_say328,Tbl3_say329,Tbl3_say330,Tbl3_say331
        DW      Tbl3_say332,Tbl3_say333,Tbl3_say334,Tbl3_say335,Tbl3_say336
        DW      Tbl3_say337,Tbl3_say338,Tbl3_say339,Tbl3_say340,Tbl3_say341
        DW      Tbl3_say342,Tbl3_say343,Tbl3_say344,Tbl3_say345,Tbl3_say346
        DW      Tbl3_say347,Tbl3_say348,Tbl3_say349,Tbl3_say350,Tbl3_say351
        DW      Tbl3_say352,Tbl3_say353,Tbl3_say354,Tbl3_say355,Tbl3_say356
        DW      Tbl3_say357,Tbl3_say358,Tbl3_say359,Tbl3_say360,Tbl3_say361
        DW      Tbl3_say362,Tbl3_say363,Tbl3_say364,Tbl3_say365,Tbl3_say366
        DW      Tbl3_say367,Tbl3_say368,Tbl3_say369,Tbl3_say370,Tbl3_say371
        DW      Tbl3_say372,Tbl3_say373,Tbl3_say374,Tbl3_say375,Tbl3_say376
        DW      Tbl3_say377,Tbl3_say378,Tbl3_say379,Tbl3_say380,Tbl3_say381
```

```
        DW      Tbl3_say382,Tbl3_say383
;
;
Spch_grp4:
        DW      Tbl4_say384
        DW      Tbl4_say385,Tbl4_say386,Tbl4_say387,Tbl4_say388,Tbl4_say389
        DW      Tbl4_say390,Tbl4_say391,Tbl4_say392,Tbl4_say393,Tbl4_say394
        DW      Tbl4_say395,Tbl4_say396,Tbl4_say397,Tbl4_say398,Tbl4_say399
        DW      Tbl4_say400,Tbl4_say401,Tbl4_say402,Tbl4_say403,Tbl4_say404
        DW      Tbl4_say405,Tbl4_say406,Tbl4_say407,Tbl4_say408,Tbl4_say409
        DW      Tbl4_say410,Tbl4_say411,Tbl4_say412,Tbl4_say413,Tbl4_say414
        DW      Tbl4_say415,Tbl4_say416,Tbl4_say417,Tbl4_say418,Tbl4_say419
        DW      Tbl4_say420,Tbl4_say421,Tbl4_say422,Tbl4_say423,Tbl4_say424
        DW      Tbl4_say425,Tbl4_say426,Tbl4_say427,Tbl4_say428,Tbl4_say429
        DW      Tbl4_say430,Tbl4_say431,Tbl4_say432,Tbl4_say433,Tbl4_say434
        DW      Tbl4_say435,Tbl4_say436,Tbl4_say437,Tbl4_say438,Tbl4_say439
        DW      Tbl4_say440,Tbl4_say441,Tbl4_say442,Tbl4_say443,Tbl4_say444
        DW      Tbl4_say445,Tbl4_say446,Tbl4_say447,Tbl4_say448,Tbl4_say449
        DW      Tbl4_say450,Tbl4_say451,Tbl4_say452,Tbl4_say453,Tbl4_say454
        DW      Tbl4_say455,Tbl4_say456,Tbl4_say457,Tbl4_say458,Tbl4_say459
        DW      Tbl4_say460,Tbl4_say461,Tbl4_say462,Tbl4_say463,Tbl4_say464
        DW      Tbl4_say465,Tbl4_say466,Tbl4_say467,Tbl4_say468,Tbl4_say469
        DW      Tbl4_say470,Tbl4_say471,Tbl4_say472,Tbl4_say473,Tbl4_say474
        DW      Tbl4_say475,Tbl4_say476,Tbl4_say477,Tbl4_say478,Tbl4_say479
        DW      Tbl4_say480,Tbl4_say481,Tbl4_say482,Tbl4_say483,Tbl4_say484
        DW      Tbl4_say485,Tbl4_say486,Tbl4_say487,Tbl4_say488,Tbl4_say489
        DW      Tbl4_say490,Tbl4_say491,Tbl4_say492,Tbl4_say493,Tbl4_say494
        DW      Tbl4_say495,Tbl4_say496,Tbl4_say497,Tbl4_say498,Tbl4_say499
        DW      Tbl4_say500,Tbl4_say501,Tbl4_say502,Tbl4_say503,Tbl4_say504
        DW      Tbl4_say505,Tbl4_say506,Tbl4_say507,Tbl4_say508,Tbl4_say509
        DW      Tbl4_say510,Tbl4_say511

;*****************************************************************************
***
;*****************************************************************************
***
;*****************************************************************************
***
;*****************************************************************************
***
;*****************************************************************************
***

; ALL SPEECH SAYSENT START HERE ;;;;;;;;;
;; Saysent groups for Tbl 1


; The first line of each group is the speech speed command.
; This is a number from 40 - 55 where 46 is standard speed
;
;
; The next line is PITCH control which works as follows:
; Actual numeric value for TI pitch control

;  bit 7 set = subtract value from current course value
;        clr = add value to current course value
;  bit 6 set = select music pitch table
;        clr = select normal speech pitch table
;  bit 0-5 value to change course value (no change = 0)
```

```
;      8Fh    ;hi voice  (8f is very squeeeeeke) (8F=143)
;      81h    ;one step higher than normal  use range 81-8F (129-143)
;      00     ;normal voice
;      01     ;one step lower than normal
;      2fh    ;lo voice ( very low)   use range 01-2F (01-47)
;

; A math routine in 'say_0' converts the value for + or -
; if <80 then subtracts from 80 to get the minus version of 00
; ie, if number is 70 then TI gets 10 (which is -10)
; If number is 80 or > 80 then get sent literal as positive.

; NOTE: MAX POSITIVE IS 8B
;       MAX NEGATIVE is 2F ( 80h - 2Fh or 51h)
; 8Bh is hi voice  (8f is very squeeeeeke)
; 2Fh lo voice ( very low)


; When entering changes, 'Voice' holds the current pitch for Furby
; and it is modified by adding or subtracting a pitch change :::

; ex:  Voice+8   increases the pitch from the current voice by 8

; ex:  Voice-10   decreases the pitch from the current voice by 10


; The next group of entries are the speech words.
; The last line is the terminator of 'FF'

; (BOTTOM)
;
;  1 is very fast
;  46 is average
;  255 is very slow
;
;  DB   46       (speed of speech)
;  DB   123      (do sound 123)
;  DB   43       (do sound 43)
;  DB   FFH
;         PITCH PROGRAMMING RANGE:
;         Voice+8 (highest)
;         Voice-20 (lowest)
;;
Tbl1_say000:
        DB      46
        DB      Voice
        DB      163
        DB      FFH


;GEORGE 07/03/98
Tbl1_say001:                                        ;dON START  SEQ1 AGE1
        DB      46      ;speech speed
        DB      Voice+8
        DB      149,162,162,164,149   ;DONE 1FRONT SEQ1
        DB      FFH     ;end

;
Tbl1_say002:
```

```
        DB      52      ;speech speed
        DB      Voice+8   ;system pitch setting
        DB      117,59                  ;DONE 1FRONT SEQ2 age1
        DB      FFH     ;end
;
Tbl1_say003:
        DB      46      ;speech speed
        DB      Voice-4   ;system pitch setting
        DB      118             ;1front seq3 - seq4-part1-SEQ7PART2
        DB      FFH     ;end
;
Tbl1_say004:
        DB      46      ;speech speed
        DB      Voice   ;system pitch setting
        DB      62,22,85                        ;1front seq3 part2
        DB      FFH     ;end
;
Tbl1_say005:
        DB      50      ;speech speed
        DB      Voice+8   ;system pitch setting
        DB      58,39      ;1front seq4 part 2
        DB      FFH     ;end
;
Tbl1_say006:
        DB      46      ;speech speed
        DB      Voice   ;pitch control
        DB      162,162,99,117          ;seq5 age1 front   part of seq6
        DB      FFH     ;end
;
Tbl1_say007:
        DB       55       ;speech speed
        DB      Voice+8   ;system pitch setting
        DB      156       ;seq6 age1 front back part
        DB      FFH       ;end
;
Tbl1_say008:
        DB      46      ;speech speed
        DB     Voice  ;pitch control
        DB     162,162,99,10,39       ;SEQ7 FRONT AGE1 ADD SAY 003
        DB      FFH     ;end
;
Tbl1_say009:
        DB      46      ;speech speed
        DB       Voice  ;system pitch setting
        DB      99,99,145                       ;SEQ8 FRONT AGE1
        DB      FFH     ;end
;
Tbl1_say010:
        DB      46      ;speech speed
        DB      Voice   ;system pitch setting
        DB      98                               ;seq9 FRONT AGE1
        DB      FFH     ;end
;
Tbl1_say011:
        DB      30    ;speech speed
        DB  Voice+8   ;system pitch setting
        DB    96,165,165,165,129,149          ;seq10 FRONT AGE1 ADD SAY20
        DB  FFH     ;end
;
Tbl1_say012:
```

```
        DB      50      ;speech speed
        DB      Voice   ;system pitch   tting
        DB      136,165,162,45          , eq11 FRONT AGE1 ADD SAY20
        DB      FFH     ;end
;
Tbl1_say013:
        DB      58      ;speech speed
        DB      Voice   ;system pitch setting
        DB      1?8,136,117                     ;seq12 FRONT AGE1 ADD
SAY20 ON FRONTPART
        DB      FFH     ;end
;
Tbl1_say014:
        DB      60      ;sp
        DB      Voice+8     .  .    itch set ing
        DB      145,162                                 ;seq13 FRONT AGE1
ADD SAY22
        DB      FFH     ;end
;
Tbl1_say015:
        DB      46      ;speech speed
        DB      Voice+8   ;system pitch setting
        DB      156                     ;seq14 FRONT AGE1
        DB      FFH     ;end
;
Tbl1_say016:
        DB      46      ;speech speed
        DB      Voice+7   ;system pitch setting
        DB      119,58                          ;seq15 FR  T AGE1
        DB      FFH     ;end
;
Tbl1_say017:
        DB      46      ;speech speed
        DB      Voice   ;system pitch setting
        DB      37      ;seq16 FRONT AGE1 BETWEEN 2(SAY20)ADDSAY37
        DB      FFH     ;end
;
Tbl1_say018:
        DB      46      ;speech speed
        DB      Voice   ;system pitch setting
        DB      123       ;SEQ16 FRONT AGE1
        DB      FFH     ;end
;
Tbl1_say019:
        DB      46      ;speech speed
        DB      Voice   ;system pitch setting
        DB      118         ;SEQ1 FRONT AGE2 REPEAT 22
        DB      FFH     ;end
;
Tbl1_say020:
        DB      46      ;speech speed
        DB      Voice+7   ;system pitch setting
        DB      77,35   ;SEQ2 FRONT ADD  20 TO FRONT
        DB      FFH     ;end
;
Tbl1_say021:
        DB      46      ;speech speed
        DB      Voice   ;system pitch setting
        DB       39,39      ;SEQ3AGE2 FRONT ADD SEQ9AGE1
        DB      FFH     ;end
```

```
;
Tbl1_say022:
        DB      56              ;speech speed
        DB      Voice+7         ;system pitch setting
        DB      156         ;SEQ4 AGE2 FRONT
        DB      FFH     ;end
;
Tbl1_say023:
        DB      46      ;speech speed
        DB      Voice+7   ;system pitch setting
        DB      8,162,22        ;SEQ4 AGE2 FRONT
        DB      FFH     ;end
;
Tbl1_say024:
        DB      46      ;speech speed
        DB      Voice+7   ;system pitch setting
        DB      117,81,27               ;SEQ5 AGE2 FRONT
        DB      FFH     ;end
;
Tbl1_say025:
        DB      46      ;speech speed
        DB      Voic      ;system pitch setting
        DB      99,35,48,164,77      ;SEQ6 AGE2 FRONT
        DB      FFH     ;end
;
Tbl1_say026:
        DB      46      ;speech speed
        DB      Voice+8   ;system pitch setting
        DB      99          ;SEQ 7 AGE2 FRONT PART 1
        DB      FFH     ;end
;
Tbl1_say027:
        DB      46      ;speech speed
        DB      Voice+7   ;system pitch setting
        DB      60,39,117           ;SEQ 7 AGE2 FRONT PART 2
        DB      FFH     ;end
;
Tbl1_say028:
        DB      46      ;speech speed
        DB      Voice   ;system pitch setting
        DB      145                 ;SEQ 8 AGE2 FRONT say45(2)+22
        DB      FFH     ;end
;
Tbl1_say029:
        DB      46      ;speech speed
        DB      Voice+5   ;system pitch setting
        DB      149,162,162,164,149   ;FRONT SEQ9 AGE2
        DB      FFH     ;end
;
Tbl1_say030:
        DB      60      ;speech speed
        DB      Voice+7   ;system pitch setting
        DB      96,163,163,129              ;SEQ10 FRONT AGE 2 ADD 48
        DB      FFH     ;end
;
Tbl1_say031:
        DB      60      ;speech speed
        DB      Voice+8   ;system pitch setting
        DB      39,63                   ;SEQ11 FRONT AGE 2
        DB      FFH     ;end
```

```
;
Tbl1_say032:
        DB      46      ;speech speed
        DB      Voice+7   ;system pitch setting
        DB      128,117                         ;SEQ12 FRONT AGE 2 ADD 20
        DB      FFH     ;end
;
Tbl1_say033:
        DB      56      ;speech speed
        DB      Voice+7    ;system pitch setting
        DB      99,55,162,28        ;SEQ14 FRONT AGE2
        DB      FFH     ;end
;
Tbl1_say034:
        DB      46      ;speech speed
        DB      Voice+6  ;system pitch setting
        DB      136,34              ;SEQ15 FRONT AGE2 ADD 20
        DB      FFH     ;end
;
Tbl1_say035:
        DB      56      ;speech speed
        DB      Voice+6   ;system pitch setting
        DB      35,162,48,162,93,133    ;SEQ16 FRONT AGE2 ADD20 TO
BEGGINING
        DB      FFH     ;end
;
Tbl1_say036:
        DB      50      ;speech speed
        DB      Voice+3   ;system pitch setting
        DB      162,1                           ;SEQ1 FRONT AGE3 ADD 20
        DB      FFH     ;end
;
Tbl1_say037:
        DB      46      ;speech speed
        DB      Voice   ;system pitch setting
        DB      81,77,52    ;SEQ2 FRONT AGE3
        DB      FFH     ;end
;
Tbl1_say038:
        DB      46      ;speech speed
        DB      Voice+8   ;system pitch setting
        DB      1,1               ;SEQ3 FRONT AGE3 ADD29
        DB      FFH     ;end
;
Tbl1_say039:
        DB      50      ;speech speed
        DB      Voice+6   ;system pitch setting
        DB      162,14,27           ;SEQ4 FRONT AGE4 ADD41
        DB      FFH     ;end
;

;
;ERROR
;Tbl1_say040:
;        DB      46      ;speech speed
;        DB      Voice   ;system pitch setting
;        DB
;        DB      FFH     ;end

;
```

```
Tbl1_say040:
        DB      46      ;speech speed
        DB      Voice   ;system pitch setting
        DB      99,35,47,58       ;SEQ6 FRONT AGE3
        DB      FFH     ;end
;
Tbl1_say041:
        DB      46      ;speech speed
        DB      Voice   ;system pitch setting
        DB      99,60,77,23     ;SEQ7 FRONT AGE3 ADD 22
        DB      FFH     ;end
;
Tbl1_say042:
        DB      46      ;speech speed
        DB      Voice   ;system pitch setting
        DB      99,145    ;SEQ8 FRONT AGE3 ADD 22
        DB      FFH     ;end
;
;ERROR
;Tbl1_say044:
;        DB      46      ;speech speed
;        DB      Voice   ;system pitch setting
;        DB      4  GO TO 22
;        DB      FFH     ;end


;
Tbl1_say043:
            DB  30      ;speech speed
            DB  Voice+8   ;system pitch setting
            DB    96,165,165,165,129,149          ;seq10 FRONT AGE3 ADD
SAY20
            DB  FFH     ;end
;
Tbl1_say044:
        DB      50      ;speech speed
        DB      Voice+4   ;system pitch setting
        DB      145       ;SEQ11 FRONT AGE3
        DB      FFH     ;end
;
Tbl1_say045:
        DB      46      ;speech speed
        DB      Voice   ;system pitch setting
        DB      119,77          ;SEQ12 FRONT AGE3 (HEEY,TICKLE ME) ADD20
        DB      FFH     ;end
;
Tbl1_say046:
        DB      46      ;speech speed
        DB      Voice   ;system pitch setting
        DB      128             ;SEQ13 fRONT AGE3 (NANNY,NANNY) ADD20
        DB      FFH     ;end
;
Tbl1_say047:
        DB      46      ;speech speed
        DB      Voice   ;system pitch setting
        DB      136,117  ;SEQ13 fRONT AGE3 (RASBERRY+ hE HE HE ) ADD20
        DB      FFH     ;end
;
Tbl1_say048:
        DB      46      ;speech speed
```

```
        DB      Voice   ;system pitch setting
        DB      35,162,47       ;SEQ16 kAH LOVE FRONT AGE3 ADD 20
        DB      FFH     ;end
;
Tbl1_say049:
        DB      56      ;speech speed
        DB      Voice+6   ;system pitch setting
        DB      81,133     ;SEQ16 (U-NYE QUICK KISS) FRONT AGE3 ADD20
        DB      FFH     ;end
;
Tbl1_say050:
        DB      46      ;speech speed
        DB      Voice   ;system pitch setting
        DB      77     ; SEQ2 (TICKLE) FRONT AGE4
        DB      FFH     ;end
;
Tbl1_say051:
        DB      46      ;speech speed
        DB      Voice+6   ;system pitch setting
        DB      1       ;SEQ2 (AGAIN) FRONT AGE4
        DB      FFH     ;end
;
Tbl1_say052:
        DB      46      ;speech speed
        DB      Voice   ;system pitch setting
        DB      93      ;SEQ3 (YOU) FRONT AGE4
        DB      FFH     ;end
;
Tbl1_say053:
        DB      46      ;speech speed
        DB      Voice   ;system pitch setting
        DB      52       ;SEQ3 (ME) FRONT AGE4
        DB      FFH     ;end
;
Tbl1_say054:
        DB      46      ;speech speed
        DB      Voice   ;system pitch setting
        DB      47      ;SEQ4 (LOVE) FRONT AGE4
        DB      FFH     ;end
;
Tbl1_say055:
        DB      46      ;speech speed
        DB      Voice+8   ;system pitch setting
        DB      117       ;SEQ5 (HE HE HE) FRONT AGE4
        DB      FFH     ;end
;
Tbl1_say056:
        DB      46      ;speech speed
        DB      Voice   ;system pitch setting
        DB      8,27       ;SEQ5 (BIG FUN) FRONT AGE4 ADD26
        DB      FFH     ;end
;
Tbl1_say057:
        DB      46      ;speech speed
        DB      Voice   ;system pitch setting
        DB      60       ;SEQ8 (NO) FRONT  AGE4
        DB      FFH     ;end
;
Tbl1_say058:
        DB      46      ;speech speed
```

```
        DB      Voice   ;system pitch setting
        DB      68      ;SEQ8 (PLEASE) FRONT  AGE4
        DB      FFH     ;end
;
Tbl1_say059:
        DB      46      ;speech speed
        DB      Voice+8   ;system pitch setting
        DB      119      ;SEQ9 (HEEY) FRONT AGE4 ADD71
        DB      FFH     ;end
;
Tbl1_say060:
        DB      46      ;speech speed
        DB      Voice   ;system pitch setting
        DB      66      ;SEQ14 (PARTY) FRONT AGE4
        DB      FFH     ;end
;
Tbl1_say061:
        DB      46      ;speech speed
        DB      Voice   ;system pitch setting
        DB      108     ;SEQ15 (WA WA WA) fRONT AGE4 ADD 22
        DB      FFH     ;end
;END GEORGE 07/03/98
;
;GEORGE 07/04/98
;START SAY FORTUNE
Tbl1_say062:
        DB      46                      ;speech speed
        DB      Voice-6                 ;system pitch setting
        DB      3                       ;FORTUNE TELL (ASK)
      DB    FFH   ;end
;
Tbl1_say063:
        DB      46                      ;speech speed
        DB      Voice                   ;system pitch setting
        DB      92                      ;FORTUNE TELL (YES)
      DB    FFH   ;end
;
Tbl1_say064:
        DB      46                      ;speech speed
        DB      Voice                   ;system pitch setting
        DB      8                       ;FORTUNE TELL (BIG)
      DB    FFH   ;end
;
Tbl1_say065:
        DB      46                      ;speech speed
        DB      Voice+8                 ;system pitch setting
        DB      84,8                    ;FORTUNE TELL (VERY,BIG)
      DB    FFH   ;end
;
Tbl1_say066:
        DB      100                     ;speech speed
        DB      Voice                   ;system pitch setting
        DB      162,70                  ;FORTUNE TELL (SEE YES)
      DB    FFH   ;end
;
Tbl1_say067:
        DB      100                     ;speech speed
        DB      Voice-4                 ;system pitch setting
        DB      157,162,157             ;Fortune tell (SLOW WHINE)
      DB    FFH   ;end
```

```
;
Tbl1_say068:
        DB      46                      ;speech speed
        DB      Voice                   ;system pitch setting
        DB      64                      ;FORTUNE TELL (O2WHA)
      DB    FFH   ;end
;
Tbl1_say069:
        DB      46                      ;speech speed
        DB      Voice+5                 ;system pitch setting
        DB      60                      ;FORTUNE TELL  (NO)
        DB      FFH       ;end
;
Tbl1_say070:
        DB      46                      ;speech speed
        DB      Voice+7                 ;system pitch setting
        DB      90                      ;FORTUNE (WORRY)
      DB    FFH   ;end
;
Tbl1_say071:
        DB      46                      ;speech speed
        DB      Voice+7                 ;system pitch setting
        DB      73                      ;FORTUNE (SOUND)
      DB    FFH   ;end
;
Tbl1_say072:
        DB      46                      ;speech speed
        DB      Voice                   ;system pitch setting
        DB      28                      ;FORTUNE (GOOD)
      DB    FFH   ;end
;
Tbl1_say073:
        DB      46                      ;speech speed
        DB      Voice                   ;system pitch setting
        DB      84                      ;FORTUNE (VERY)
      DB    FFH   ;end
;
Tbl1_say074:
        DB      50                      ;speech speed
        DB      Voice+8                 ;system pitch setting
        DB      159                     ;FORTUNE (WH0OPEE)
      DB    FFH   ;end
;
Tbl1_say075:
        DB      46                      ;speech speed
        DB      Voice-5                 ;system pitch setting
        DB       28                     ;FORTUNE (GOOD)
      DB    FFH   ;end
;
Tbl1_say076:
        DB      56                      ;speech speed
        DB      Voice+7                 ;system pitch setting
        DB      136                     ;FORTUNE (RASPBERRY)
      DB    FFH   ;end
;
Tbl1_say077:
        DB      50                      ;speech speed
        DB      Voice                   ;system pitch setting
        DB      129                     ;FORTUNE (oH oH)
      DB    FFH   ;end
```

```
;
Tbl1_say078:
        DB      50                      ;speech speed
        DB      Voice+7                 ;system pitch setting
        DB      49                      ;FORTUNE (MAY BEE)
      DB    FFH   ;end

;END SAY FORTUNE
;END GEORGE 07/04/98

;START HANGOUT
;GEORGE 07/04/98
Tbl1_say079:
        DB      56              ;speech speed
        DB      Voice+8         ;system pitch setting
        DB      110             ;SEQ1 HANGING(DE DE DE ,DUM  DUM DUM
DUM) AGE1
      DB    FFH   ;end
;
Tbl1_say080:
        DB      60              ;speech speed
        DB      Voice+8         ;system pitch setting
        DB      109             ;SEQ1 HANGING( DUM DUM DUM) AGE1; ADD 83
      DB    FFH   ;end
;
Tbl1_say081:
        DB      56      ;speech speed
        DB      Voice+8   ;system pitch setting
        DB      116     ;SEQ2 HANGING (BEEDO)
      DB    FFH   ;end
;
Tbl1_say082:
        DB      46              ;speech speed
        DB      Voice+7         ;system pitch setting
        DB      113             :SEQ3 HANGING (YA DA DA )
      DB    FFH   ;end
;
Tbl1_say083:
        DB      53                      ;speech speed
        DB      Voice+5                 ;system pitch setting
        DB      162,114,162,114         :SEQ3 HANGING (OMPAH  bRUMM)
      DB    FFH   ;end
;
Tbl1_say084:
        DB      46      ;speech speed
        DB      Voice+8 ;system pitch setting
        DB      115     :SEQ3 HANGING (YA DA DA OMPAH  bRUMM BABABUM)
      DB    FFH   ;end
;
Tbl1_say085:
        DB      60      ;speech speed
        DB      Voice+5 ;system pitch setting
        DB      126,163 ;SEQ4 HANGING (LA LA)
      DB    FFH   ;end
;
Tbl1_say086:
        DB      56      ;speech speed
        DB      Voice+5 ;system pitch setting
        DB      127     ;SEQ4 HANGING (LA LA)
      DB    FFH   ;end
```

```
;
Tbl1_say087:
        DB      46      ;speech speed
      DB    Voice ;system pitch setting
        DB      101     ;SEQ5 HANGING (HUMMMMMM)
      DB    FFH   ;end
;
Tbl1_say088:
        DB      46      ;speech speed
      DB    Voice ;system pitch setting
        DB      11      ;SEQ5 HANGING (BO DAH WA LO)
      DB    FFH   ;end
;
Tbl1_say089:
        DB      46      ;speech speed
        DB      Voice+7 ;system pitch setting
        DB      143,163 ;SEQ6 HANGING (SNORE)
      DB    FFH   ;end
;
Tbl1_say090:
        DB      46      ;speech speed
      DB    Voice ;system pitch setting
        DB      148     ;SEQ6 HANGING (SHOUT)
      DB    FFH   ;end
;
Tbl1_say091:
        DB      46      ;speech speed
      DB    Voice ;system pitch setting
        DB      63,75   ;SEQ6 HANGING (OK,KAH)
      DB    FFH   ;end
;
Tbl1_say092:
        DB      46      ;speech speed
      DB    Voice ;system pitch setting
        DB      82      ;SEQ6 HANGING (U-TYE)
      DB    FFH   ;end
;
Tbl1_say093:
        DB      60      ;speech speed
        DB      Voice+8   ;system pitch setting
        DB      144       ;SEQ7 HANGING (SOFTER)
      DB    FFH   ;end
;
Tbl1_say094:
        DB      46      ;speech speed
        DB      Voice-4   ;system pitch setting
        DB      144       ;SEQ7 HANGING (SOFTER)
      DB    FFH   ;end
;
Tbl1_say095:
        DB      46      ;speech speed
        DB      Voice   ;system pitch setting
        DB      124,162       ;SEQ8 hANGING (kITTY KITTY)
      DB    FFH   ;end
;
Tbl1_say096:
        DB      56
        DB      Voice   ;system pitch setting
        DB      112     ;SEQ9 HANGING (DO BE DOBE DO)
      DB    FFH   ;end
```

```
;
Tbl1_say097:
        DB      60      ;speech speed
        DB      Voice+7   ;system pitch setting
        DB      161,164,164,161   ;SEQ10 HANGING (YAWN)
      DB    FFH   ;end
;
Tbl1_say098:
        DB      100      ;speech speed
        DB      Voice+6   ;system pitch setting
        DB      140      ;SEQ11 AND SEQ12 HANGING (SIGH)
      DB    FFH   ;end
;
Tbl1_say099:
        DB      46      ;speech speed
        DB      Voice+8   ;system pitch setting
        DB      100       ;SEQ13 SEQ14 HANGING (HAA)
      DB    FFH   ;end
;
Tbl1_say100:
        DB      46      ;speech speed
      DB    Voice ;system pitch setting
        DB      119     ;SEQ14 HANGING (HEEY)
      DB    FFH   ;end
;
Tbl1_say101:
        DB      46      ;speech speed
      DB    Voice ;system pitch setting
        DB      132,165,132     ;SEQ16 HANGING (PHONE) ADD20
      DB    FFH   ;end
;
Tbl1_say102:
        DB      46      ;speech speed
      DB    Voice ;system pitch setting
        DB      165,165,165,165     ;SEQ16 HANGING (PAUSE) ADD20
      DB    FFH   ;end
;
Tbl1_say103:
        DB      46      ;speech speed
        DB      Voice+5  ;system pitch setting
        DB      83      ;SEQ6 HANGING (UP)
      DB    FFH   ;end
;
Tbl1_say104:
        DB      46      ;speech speed
      DB    Voice ;system pitch setting
        DB      52      ;SEQ6 HANGING AGE3  (ME)
      DB    FFH   ;end
;
Tbl1_say105:
        DB      46      ;speech speed
      DB    Voice ;system pitch setting
        DB      63      ;SEQ6 HANGING AGE3 (OK)
      DB    FFH   ;end
;
Tbl1_say106:
        DB      46      ;speech speed
      DB    Voice ;system pitch setting
        DB      13      ;SEQ5 HANGING AGE3 AND 4
      DB    FFH   ;end
```

```
;END HANGOUT
;
;
;
Tbl1_say107:
        DB      46      ;speech speed
        DB      Voice   ;system pitch setting
        DB      165,165 ;Fortune delay
      DB    FFH   ;end
;
;END GEORGE 07/04/98
;START FEED
;GEORGE 07/05/98

;-----------------------------------------sTART fEED
;
; spch_grp2 was here
;; Saysent groups for Tbl 2


;STARTS AT 128
Tbl1_say108:
        DB      100     ;speech speed
      DB    Voice ;system pitch setting
      DB    166   ;SEQ1 FEED AGE1 (UUMMM)
      DB    FFH   ;end

;NOT USED
;Tbl2_say129:
;        DB      46      ;speech speed
;        DB      Voice+8   ;system pitch setting
;;         DB          ;SEQ1 FEED AGE1 (AY-TAY)
;        DB      FFH     ;end
;

Tbl1_say109:
        DB      100     ;speech speed
      DB    Voice ;system pitch setting
        DB      167,167     ;SEQ1 FEED AGE1 (AAAAH)
      DB    FFH   ;end
;
Tbl1_say110:
        DB      56      ;speech speed
        DB      Voice+3   ;system pitch setting
      DB    39    ;SEQ2 FEED AGE1 (KOH-KOH)
      DB    FFH   ;end
;
Tbl1_say111:
        DB      56      ;speech speed
        DB      Voice+7   ;system pitch setting
        DB      55      ;SEQ2 FEED AGE1 (MEE MEE)
      DB    FFH   ;end
;
Tbl1_say112:
        DB      50      ;speech speed
      DB    Voice ;system pitch setting
        DB      25      ;SEQ2 FEED AGE1 (E-DAY)
      DB    FFH   ;end
;
```

```
Tbl1_say113:
        DB      58      ;speech speed
        DB      Voice+7   ;system pitch setting
      DB    23    ;SEQ2 FEED AGE1 (DO MOH)
      DB    FFH   ;end
;
Tbl1_say114:
        DB      58      ;speech speed
      DB    Voice ;system pitch setting
      DB    79    ;TOH-DYE
      DB    FFH   ;end
;
Tbl1_say115:
        DB      46      ;speech speed
      DB    Voice ;system pitch setting
      DB    97    ;BURP
      DB    FFH   ;end
;
Tbl1_say116:
        DB      46      ;speech speed
      DB    Voice ;system pitch setting
      DB    140   ;SIGH
      DB    FFH   ;end
;
Tbl1_say117:
        DB      46      ;speech speed
      DB    Voice ;system pitch setting
        DB      10      ;BOO
      DB    FFH   ;end
;
Tbl1_say118:
        DB      46      ;speech speed
      DB    Voice ;system pitch setting
      DB    85    ;WAH
      DB    FFH   ;end
;
Tbl1_say119:
        DB      60      ;speech speed
        DB      Voice+8   ;system pitch setting
        DB      80      ;TOH-LOO
      DB    FFH   ;end
;
Tbl1_say120:
        DB      46      ;speech speed
        DB      Voice+8   ;system pitch setting   ;A TAY
        DB      7
      DB    FFH   ;end
;
Tbl1_say121:
        DB      46      ;speech speed
      DB    Voice ;system pitch setting
        DB      33      ;SEQ1 FEED AGE2 HUNGRY
      DB    FFH   ;end
;
;143 SAME AS TBL1_SAY072
;Tbl2_say143:
;       DB      46      ;speech speed
;       DB      Voice   ;system pitch setting
;       DB      28      ;SEQ2 FEED AGE3 (GOOD)
;       DB      FFH     ;end
```

```
;144 SAME AS TBL1_SAY058
;Tbl2_say144:
;         DB       46       ;speech speed
;         DB       Voice+7    ;system pitch setting
;         DB       68       ;SEQ2 FEED AGE3 PLEASE
;         DB       FFH      ;end
;;
Tbl1_say122:
         DB       46       ;speech speed
         DB       Voice-2    ;system pitch setting
         DB       43       ;SEQ2 FEED AGE3 LIKE
       DB     FFH    ;end

;Tbl2_say118:
;         DB       60       ;speech speed
;         DB       Voice-8    ;system pitch setting
;         DB       161,164,161    ;SEQ10 HANGING (YAWN)
;         DB       FFH      ;end
;
;Tbl2_say119:
;         DB       60       ;speech speed
;         DB       55       ;speech speed
;         DB       Voice+3    ;system pitch setting
;         DB       165,165,144,165,144,165,144,165,144
;
;         DB       Voice    ;system pitch setting
;         DB       144
;         DB       FFH      ;end

Tbl1_say123:
         DB       46       ;speech speed
       DB     Voice ;system pitch setting
         DB       20       ;seq4 feed done
       DB     FFH    ;end
;END GEORGE 07/05/98
;END FEED
;
;
;
;WAKE
;GEORGE 07/06/98
;
;
;START AT 2
Tbl1_say124:                ;SG DONE
         DB       70       ;speech speed
         DB       Voice+6    ;pitch control
         DB       165,161
       DB     FFH    ;end
;PASS
Tbl1_say125:                ;SG DONE
         DB       55       ;speech speed
         DB       Voice-2        ;pitch control
         DB       162,63,35
       DB     FFH    ;end
;PASS
Tbl1_say126:                ;SG DONE
         DB       55       ;speech speed
       DB     Voice ;system pitch setting
```

```
        DB      82
      DB    FFH   ;end
;PASS
Tbl1_say127:              ;SG DONE
        DB      55      ;speech speed
      DB    Voice ;system pitch setting
        DB      164,83
      DB    FFH   ;end
;
Tbl2_say128:              ;SG DONE
        DB      55      ;speech speed
      DB    Voice ;system pitch setting
        DB      63,52
      DB    FFH   ;end
;
Tbl2_say129:              ;SG DONE
        DB      40      ;speech speed
      DB    Voice ;system pitch setting
        DB      163,139
      DB    FFH   ;end
;TBL1_SAY55
;Tbl1_say8:              ;SG DONE
;         DB      46      ;speech speed
;         DB      Voice+8   ;system pitch setting
;         DB      117
;         DB      FFH     ;end
;
Tbl2_say130:              ;SG DONE
        DB      55      ;speech speed
        DB      Voice-2 ;system pitch setting
        DB      63
      DB    FFH   ;end
;
Tbl2_say131:              ;SG DONE
        DB      46      ;speech speed
      DB    Voice ;system pitch setting
        DB      86
        DB      FFH     ;end
;
Tbl2_say132:              ;SG DONE
        DB      46      ;speech speed
      DB    Voice ;system pitch setting
        DB      79
      DB    FFH   ;end
;TBL1_SAY123
;Tbl1_say12:             ;SG DONE
;         DB      46      ;speech speed
;         DB      Voice   ;system pitch setting
;         DB      20
;         DB      FFH     ;end
;
Tbl2_say133:              ;SG DONE
        DB      46      ;speech speed
      DB    Voice ;system pitch setting
        DB      72
      DB    FFH   ;end
;
Tbl2_say134:              ;SG DONE
        DB      55      ;speech speed
        DB      Voice+3 ;system pitch setting
```

```
        DB      158
      DB    FFH   ;end
;
Tbl2_say135:              ;SG DONE
        DB      46        ;speech speed
      DB    Voice ;system pitch setting
        DB      35
      DB    FFH   ;end
;
Tbl2_say136:               ;SG DONE
        DB      46         ;speech speed
        DB      Voice+5    ;system pitch setting
        DB      52
        DB      FFH        ;end
;
Tbl2_say137:              ;SG DONE
        DB      55        ;speech speed
        DB      Voice+8    ;system pitch setting
        DB      8
      DB    FFH   ;end
;
Tbl2_say138:              ;SG DONE
        DB      45        ;speech speed
        DB      Voice+8    ;system pitch setting
        DB      137,137,137,138
      DB    FFH   ;end
;
Tbl2_say139:              ;SG DONE
        DB      60        ;speech speed
      DB    Voice ;system pitch setting
        DB      149
      DB    FFH   ;end
;
Tbl2_say140:              ;SG DONE
        DB      40        ;speech speed
        DB      Voice-3    ;system pitch setting
        DB      16
        DB      FFH       ;end
;
Tbl2_say141:              ;SG DONE
        DB      20        ;speech speed
        DB      Voice+5    ;system pitch setting
        DB      161
      DB    FFH   ;end
;
Tbl2_say142:              ;SG DONE
        DB      46        ;speech speed
        DB      Voice-9    ;system pitch setting
        DB      74
      DB    FFH   ;end
;
Tbl2_say143:              ;SG DONE
        DB      80        ;speech speed
        DB      Voice+4    ;system pitch setting
        DB      82
      DB    FFH   ;end
;
Tbl2_say144:              ;SG DONE
        DB      46        ;speech speed
      DB    Voice ;system pitch setting
```

```
        DB      14
      DB    FFH   ;end
;
Tbl2_say145:              ;SG DONE
        DB      46      ;speech speed
      DB    Voice ;pitch control
        DB      6
      DB    FFH   ;end
;
Tbl2_say146:              ;SG DONE
        DB      46      ;speech speed
      DB    Voice ;system pitch setting
        DB      83
      DB    FFH   ;end
;
Tbl2_say147:              ;SG DONE
        DB      70      ;speech speed
      DB    Voice ;pitch control
        DB      76
      DB    FFH   ;end
;
Tbl2_say148:              ;SG DONE
        DB      60      ;speech speed
      DB    Voice ;system pitch setting
        DB      37
      DB    FFH   ;end
;TBL1_SAY53
;Tbl1_say29:               ;SG DONE
;         DB       46      ;speech speed
;         DB       Voice   ;system pitch setting
;         DB       52
;         DB       FFH     ;end
;
Tbl2_say149:              ;SG DONE
        DB      30      ;speech speed
        DB      Voice+5   ;system pitch setting
        DB      47
      DB    FFH   ;end
;
Tbl2_say150:              ;SG DONE
        DB      60      ;speech speed
        DB      Voice-3   ;system pitch setting
        DB      81
      DB    FFH   ;end
;
Tbl2_say151:              ;SG DONE
        DB      55      ;speech speed
        DB      Voice-7   ;system pitch setting
        DB      53
      DB    FFH   ;end
;
Tbl2_say152:              ;SG DONE
        DB      40      ;speech speed
        DB      Voice-10  ;system pitch setting
        DB      35
      DB    FFH   ;end

Tb  _say153:              ;SG DONE
        DB      46      ;speech speed
        DB      Voice-10  ;system pitch setting
```

```
        DB      39
      DB    FFH   ;end
;
Tbl2_say154:              ;SG DONE
        DB      55      ;speech speed
        DB      Voice+3   ;system pitch setting
        DB      165,165,144,165,144,165,144,165,165,165,165,144
      DB    FFH   ;end
;
Tbl2_say155:              ;SG DONE
        DB      46      ;speech speed
      DB    Voice ;system pitch setting
        DB      72
      DB    FFH   ;end
;
Tbl2_say156:              ;SG DONE
        DB      60      ;speech speed
      DB    Voice ;system pitch setting
        DB      1
      DB    FFH   ;end


;TBL1_SAY53
;Tbl1_say38:              ;SG DONE
;         DB      46      ;speech speed
;         DB      Voice   ;system pitch setting
;         DB      52
;         DB      FFH     ;end
;END GEORGE 07/06/98
;END WAKE
;
;
;GEORGE 07/06/98
;HUNGER
Tbl2_say157:              ;SG DONE  ;HUNGER
        DB      65      ;speech speed
        DB      Voice+8  ;system pitch setting
        DB      68
      DB    FFH   ;end
;
Tbl2_say158:              ;SG DONE
        DB      75      ;speech speed
      DB    Voice ;system pitch setting
        DB      23
      DB    FFH   ;end
;
Tbl2_say159:              ;SG DONE
        DB      40      ;speech speed
        DB      Voice-7   ;system pitch setting
        DB      7
      DB    FFH   ;end
;
Tbl2_say160:              ;SG DONE
        DB      55      ;speech speed
      DB    Voice ;system pitch setting
        DB      33
        DB      FFH     ;end
;
Tbl2_say161:              ;SG DONE
        DB      75      ;speech speed
```

```
      DB    Voice ;system pitch setting
        DB      55
      DB    FFH   ;end
;
Tbl2_say162:              ;SG DONE
        DB      40       ;speech speed
        DB      Voice-15   ;system pitch setting
        DB      84
      DB    FFH   ;end
;
Tbl2_say163:              ;SG DONE
        DB      65       ;speech speed
        DB      Voice+8    ;system pitch setting
        DB      157
      DB    FFH   ;end
;
Tbl2_say164:              ;SG DONE
        DB      55       ;speech speed
        DB      Voice+6    ;system pitch setting
        DB      119
      DB    FFH   ;end
;
Tbl2_say165:              ;SG DONE
        DB      65       ;speech speed
        DB      Voice+8    ;system pitch setting
        DB      85
      DB    FFH   ;end
;
Tbl2_say166:              ;SG DONE
        DB      55       ;speech speed
      DB    Voice ;system pitch setting
        DB      14
      DB    FFH   ;end
;
Tbl2_say167:              ;SG DONE
        DB      40       ;speech speed
      DB    Voice ;system pitch setting
        DB      8
      DB    FFH   ;end

Tbl2_say168:              ;SG DONE  ;SAME AS SAY135 WITH DIFFERENT MOTOR
POS.
        DB      46       ;speech speed
      DB    Voice ;system pitch setting
        DB      35
      DB    FFH   ;end
;END GEORGE 07/06/98
;END HUNGER

;
;
;GEORGE 07/07/98
;INVERT
;WAS68
Tbl2_say169:              ;SG DONE  ;INVERT
        DB      85       ;speech speed
      DB    Voice ;system pitch setting
        DB      36
            DB    FFH   ;end
;
```

```
Tbl2_say170:                ;SG DONE
        DB      55          ;speech speed
        DB      Voice+8     ;system pitch setting
        DB      94
      DB    FFH   ;end
;
Tbl2_say171:                ;SG DONE
        DB      70          ;speech speed
        DB      Voice+8     ;system pitch setting
        DB      158
      DB    FFH   ;end
;
Tbl2_say172:                ;SG DONE
        DB      55          ;speech speed
        DB      Voice+8     ;system pitch setting
        DB      148
      DB    FFH   ;end
;
Tbl2_say173:                ;SG DONE
        DB      100         ;speech speed
        DB      Voice+8     ;system pitch setting
        DB      97
      DB    FFH   ;end
;
Tbl2_say174:                ;SG DONE
        DB      50          ;speech speed
        DB      Voice+5     ;system pitch setting
        DB      8
      DB    FFH   ;end
;
Tbl2_say175:                ;SG DONE
        DB      55          ;speech speed
        DB      Voice-5     ;system pitch setting
        DB      9
      DB    FFH   ;end
;
Tbl2_say176:                ;SG DONE
        DB      50          ;speech speed
        DB      Voice-10    ;system pitch setting
        DB      54
      DB    FFH   ;end
;
Tbl2_say177:                ;SG DONE
        DB      70          ;speech speed
        DB      Voice-6     ;system pitch setting
        DB      57
      DB    FFH   ;end
;
Tbl2_say178:                ;SG DONE
        DB      24          ;speech speed
      DB    Voice ;system pitch setting
        DB      24
      DB    FFH   ;end
;
Tbl2_say179:                ;SG DONE
        DB      55          ;speech speed
        DB      Voice-5     ;system pitch setting
        DB      10
      DB    FFH   ;end
;
```

```
Tbl2_say180:                ;SG DONE
        DB      65          ;speech speed
        DB      Voice-5     ;system pitch setting
        DB      80
      DB    FFH    ;end
;
Tbl2_say181:                ;SG DONE
        DB      55          ;speech speed
        DB      Voice-10    ;system pitch setting
        DB      60
      DB    FFH    ;end
;
Tbl2_say182:                ;SG DONE
        DB      55          ;speech speed
        DB      Voice-10    ;system pitch setting
        DB      43
      DB    FFH    ;end
;
Tbl2_say183:                ;SG DONE
        DB      75          ;speech speed
        DB      Voice-8     ;system pitch setting
        DB      90
      DB    FFH    ;end
;
Tbl2_say184:                ;SG DONE
        DB      75          ;speech speed
        DB      Voice-4     ;system pitch setting
        DB      29
      DB    FFH    ;end
;
Tbl2_say185:                ;SG DONE
        DB      55          ;speech speed
        DB      Voice+5     ;system pitch setting
        DB      34
        DB      FFH         ;end
;
Tbl2_say186:                ;SG DONE
        DB      65          ;speech speed
        DB      Voice+2     ;system pitch setting
        DB      45
      DB    FFH    ;end
;
Tbl2_say187:                ;SG DONE
        DB      65          ;speech speed
        DB      Voice-7     ;system pitch setting
        DB      39
      DB    FFH    ;end
;
Tbl2_say188:                ;SG DONE
        DB      35          ;speech speed
        DB      Voice       ;system pitch setting
        DB      130
      DB    FFH    ;end
;Tbl2_say158:
;Tbl1_say88:                 ;SG DONE
;         DB      75         ;speech speed
;         DB      Voice      ;system pitch setting
;         DB      23
;         DB      FFH        ;end
;
```

```
Tbl2_say189:                ;SG DONE
        DB      55          ;speech speed
      DB    Voice ;system pitch setting
        DB      1
      DB    FFH   ;end
;
Tbl2_say190:
        DB      100             ;speech speed
        DB      Voice           ;system pitch setting
        DB      97
      DB    FFH   ;end
;
Tbl2_say191:
        DB      100             ;speech speed
        DB      Voice-10        ;system pitch setting
        DB      97
      DB    FFH   ;end
;
Tbl2_say192:
        DB      100             ;speech speed
        DB      Voice-20        ;system pitch setting
        DB      97
      DB    FFH   ;end
;END GEORGE 07/07/98
;END INVERT

;start at 202
Tbl2_say193:    ;SG DONE  ;BACKSG
        DB      70          ;speech speed
      DB    Voice ;system pitch setting
        DB      153
      DB    FFH   ;end
;
Tbl2_say194:    ;SG DONE
        DB      75          ;speech speed
      DB    Voice ;system pitch setting
        DB      2
        DB      FFH     ;end
;
Tbl2_say195:    ;SG DONE
        DB      55          ;speech speed
      DB    Voice ;system pitch setting
        DB      39
      DB    FFH   ;end
;
Tbl2_say196:    ;SG DONE
        DB      65          ;speech speed
        DB      Voice+4   ;system pitch setting
        DB      67        ; PET
      DB    FFH   ;end
;
Tbl2_say197:    ;SG DONE
        DB      75          ;speech speed
        DB      Voice+5   ;system pitch setting
        DB      1
      DB    FFH   ;end
;
Tbl2_say198:    ;SG DONE
        DB      55          ;speech speed
        DB      Voice-10   ;system pitch setting
```

```
        DB      146
      DB    FFH   ;end
;
Tbl2_say199:    ;SG DONE
        DB      55      ;speech speed
        DB      Voice+5   ;system pitch setting
        DB      35
      DB    FFH   ;end
;
Tbl2_say200:    ;SG DONE
        DB      80      ;speech speed
        DB      Voice-5   ;system pitch setting
        DB      55
      DB    FFH   ;end
;
Tbl2_say201:    ;SG DONE
        DB      70      ;speech speed
        DB      Voice-5   ;system pitch setting
        DB      62
      DB    FFH   ;end
;
Tbl2_say202:    ;SG DONE
        DB      80      ;speech speed
        DB      Voice-5   ;system pitch setting
        DB      84
      DB    FFH   ;end
;
;Tbl2_say148
;
;Tbl2_say212:    ;SG DONE
;         DB      70      ;speech speed
;         DB      Voice-5   ;system pitch setting
;         DB      29
;         DB      FFH     ;end
;
Tbl2_say203:    ;SG DONE
        DB      70      ;speech speed
      DB    Voice ;system pitch setting
        DB      37
      DB    FFH   ;end
;
Tbl2_say204:    ;SG DONE
        DB      55      ;speech speed
      DB    Voice ;system pitch setting
        DB      152
      DB    FFH   ;end
;
Tbl2_say205:    ;SG DONE
        DB      65      ;speech speed
        DB      Voice-5   ;system pitch setting
        DB      52
      DB    FFH   ;end
;
Tbl2_say206:    ;SG DONE
        DB      65      ;speech speed
        DB      Voice+2   ;system pitch setting
        DB      47
      DB    FFH   ;end
;
Tbl2_say207:    ;SG DONE
```

```
        DB      65      ;speech speed
        DB      Voice-3   ;system pitch setting
        DB      81
      DB    FFH   ;end
;
Tbl2_say208:    ;SG DONE
        DB      70      ;speech speed
        DB      Voice+6   ;system pitch setting
        DB      48
      DB    FFH   ;end
;
Tbl2_say209:    ;SG DONE
        DB      70      ;speech speed
        DB      Voice+3   ;system pitch setting
        DB      161
      DB    FFH   ;end
;
Tbl2_say210:    ;SG DONE
        DB      55      ;speech speed
      DB    Voice ;system pitch setting
        DB      15
      DB    FFH   ;end
;
Tbl2_say211:    ;SG DONE
        DB      45      ;speech speed
        DB      Voice-10   ;system pitch setting
        DB      8
      DB    FFH   ;end
;
Tbl2_say212:    ;SG DONE
        DB      55      ;speech speed
        DB      Voice-10   ;system pitch setting
        DB      42
      DB    FFH   ;end
;
Tbl2_say213:    ;SG DONE
        DB      65      ;speech speed
        DB      Voice-15   ;system pitch setting
        DB      57
        DB      FFH     ;end
;
Tbl2_say214:    ;SG DONE
        DB      50      ;speech speed
      DB    Voice ;system pitch setting
        DB      75
      DB    FFH   ;end
;
Tbl2_say215:    ;SG DONE
        DB      55      ;speech speed
      DB    Voice ;system pitch setting
        DB      101
      DB    FFH   ;end
;
Tbl2_say216:    ;SG DONE
        DB      70      ;speech speed
        DB      Voice-3   ;system pitch setting
        DB      49
      DB    FFH   ;end
;
Tbl2_say217:    ;SG DONE
```

```
        DB      75      ;speech speed
        DB      Voice+5   ;system pitch setting
        DB      86
      DB    FFH   ;end
;
Tbl2_say218:    ;SG DONE
        DB      55      ;speech speed
      DB    Voice ;system pitch setting
        DB      72
      DB    FFH   ;end
;
Tbl2_say219:    ;SG DONE
        DB      55      ;speech speed
        DB      Voice+5   ;system pitch setting
        DB      150
      DB    FFH   ;end
;
Tbl2_say220:    ;SG DONE
        DB      55      ;speech speed
        DB      Voice+5   ;system pitch setting
        DB      151
      DB    FFH   ;end
;
Tbl2_say221:    ;SG DONE
        DB      55      ;speech speed
      DB    Voice ;system pitch setting
        DB      97
      DB    FFH   ;end
;
Tbl2_say222:    ;SG DONE
        DB      70      ;speech speed
      DB    Voice ;system pitch setting
        DB      165,149
      DB    FFH   ;end
;
Tbl2_say223:    ;SG DONE
        DB      55      ;speech speed
        DB      Voice   ;system pitch setting
        DB      129
      DB    FFH   ;end
;
Tbl2_say224:    ;SG DONE
        DB      75      ;speech speed
        DB      Voice-4   ;system pitch setting
        DB      50
      DB    FFH   ;end
;
Tbl2_say225:    ;SG DONE
        DB      55      ;speech speed
        DB      Voice+5   ;system pitch setting
        DB      32
      DB    FFH   ;end
;
Tbl2_say226:    ;SG DONE
        DB      55      ;speech speed
        DB      Voice+5   ;system pitch setting
        DB      165,140
      DB    FFH   ;end
;
Tbl2_say227:    ;SG DONE
```

```
        DB      65      ;speech speed
      DB    Voice ;system pitch setting
        DB      144
      DB    FFH   ;end
;
Tbl2_say228:    ;SG DONE
        DB      85      ;speech speed
      DB    Voice ;system pitch setting
        DB      18
      DB    FFH   ;end
;
Tbl2_say229:    ;SG DONE
        DB      50      ;speech speed
        DB      Voice+8   ;system pitch setting
        DB      118
      DB    FFH   ;end
;
Tbl2_say230:    ;SG DONE
        DB      65      ;speech speed
      DB    Voice ;system pitch setting
        DB      66
      DB    FFH   ;end
;
Tbl2_say231:    ;SG DONE
        DB      70      ;speech speed
        DB      Voice+8   ;system pitch setting
        DB      87
      DB    FFH   ;end
;
Tbl2_say232:    ;SG DONE
        DB      60      ;speech speed
        DB      Voice+8   ;system pitch setting
        DB      71
      DB    FFH   ;end
;
Tbl2_say233:    ;SG DONE
        DB      55      ;speech speed
      DB    Voice ;system pitch setting
        DB      93
      DB    FFH   ;end
;
Tbl2_say234:            ;SG DONE
        DB      46      ;speech speed
        DB      Voice-20   ;system pitch setting
        DB      161
      DB    FFH   ;end
;
Tbl2_say235:
        DB      70      ;speech speed
        DB      Voice   ;system pitch setting
        DB      81
      DB    FFH   ;end
;
Tbl2_say236:
        DB      70      ;speech speed
        DB      Voice   ;system pitch setting
        DB      93
      DB    FFH   ;end

;
```

```
;SICK
;GEORGE 07/08/98
;start at 39
Tbl2_say237:                ;SG DONE  ;SICK1
        DB      55          ;speech speed
        DB      Voice+5     ;system pitch setting
        DB      165,141
      DB    FFH   ;end
;Tbl2_say135
;Tbl1_say40:                ;SG DONE
;        DB      46          ;speech speed
;        DB      Voice       ;system pitch setting
;        DB      35
;        DB      FFH         ;end
;Tbl1_say117
;Tbl1_say41:                ;SG DONE
;        DB      46          ;speech speed
;        DB      Voice       ;system pitch setting
;        DB      10
;        DB      FFH         ;end
;
Tbl2_say238:                ;SG DONE
        DB      46          ;speech speed
      DB    Voice ;system pitch setting
        DB      40
      DB    FFH   ;end
;
Tbl2_say239:                ;SG DONE
        DB      46          ;speech speed
        DB      Voice-5     ;system pitch setting
        DB      60
      DB    FFH   ;end
;
Tbl2_say240:                ;SG DONE
        DB      50          ;speech speed
      DB    Voice ;system pitch setting
        DB      30
      DB    FFH   ;end
;Tbl1_say53
;Tbl1_say45:                ;SG DONE
;        DB      46          ;speech speed
;        DB      Voice       ;system pitch setting
;        DB      52
;        DB      FFH         ;end
;
Tbl2_say241:                ;SG DONE
        DB      70          ;speech speed
        DB      Voice-8     ;system pitch setting
        DB      17
      DB    FFH   ;end
;
Tbl2_say242:                ;SG DONE
        DB      80          ;speech speed
        DB      Voice-10    ;system pitch setting
        DB      46
      DB    FFH   ;end
;
Tbl2_say243:                ;SG DONE
        DB      55          ;speech speed
        DB      Voice-8     ;system pitch setting
```

```
        DB      8
      DB    FFH   ;end
;
Tbl2_say244:              ;SG DONE
        DB      40       ;speech speed
        DB      Voice-8   ;system pitch setting
        DB      73
      DB    FFH   ;end
;
Tbl2_say245:              ;SG DONE
        DB      75       ;speech speed
        DB      Voice-5   ;system pitch setting
        DB      80
      DB    FFH   ;end
;Tbl2_say182
;Tbl1_say51:              ;SG DONE
;        DB      55       ;speech speed
;        DB      Voice-10   ;system pitch setting
;        DB      43
;        DB      FFH      ;end
;
Tbl2_say246:              ;SG DONE
        DB      70       ;speech speed
      DB    Voice ;system pitch setting
        DB      9
      DB    FFH   ;end
;
Tbl2_say247:              ;SG DONE
        DB      60       ;speech speed
        DB      Voice-12   ;system pitch setting
        DB      90,165
        DB      FFH      ;end
;
Tbl2_say248:              ;SG DONE
        DB      100      ;speech speed
      DB    Voice ;system pitch setting
        DB      140
      DB    FFH   ;end
;
Tbl2_say249:              ;SG DONE
        DB      40       ;speech speed
        DB      Voice-20   ;system pitch setting
        DB      162,129
      DB    FFH   ;end
;
Tbl2_say250:               ;SG DONE
        DB      100      ;speech speed
      DB    Voice ;system pitch setting
        DB      142
      DB    FFH   ;end
;END GEORGE 07/08/98
;END SICK
;
;
;LIGHT
;GEORGE 07/08/98
;starts at 2
Tbl2_say251:
        DB      40     ;speech speed    DONE RB       BEGIN LIGHT
D.(BRIGHTER)
```

```
      DB    Voice ;pitch control
        DB      119,18
        DB      FFH     ;end
;
;Tbl1_say252:
;       DB      40      ;speech speed           DO NOT USE
;       DB              ;pitch control          SEE SAY 15
;       DB      FFH     ;end

;
Tbl2_say252:
        DB      75      ;speech speed      Done RB
        DB      Voice+5  ;system pitch setting
        DB      142
      DB    FFH   ;end
;
Tbl2_say253:
        DB      46      ;speech speed   done RB
      DB    Voice ;system pitch setting
        DB      158,165,165,14,6
      DB    FFH   ;end
;
Tbl2_say254:
        DB      46      ;speech speed   done RB
      DB    Voice ;system pitch setting
        DB      102,149
      DB    FFH   ;end
;
Tbl2_say255:
        DB      46      ;speech speed DONE RB
        DB      Voice+8   ;system pitch setting
        DB      119,35,164,5,81
      DB    FFH   ;end
;
Tbl3_say256:
        DB      46      ;speech speed  DONE RB
        DB      Voice-4   ;system pitch setting
        DB      148,163,145
      DB    FFH   ;end
;
Tbl3_say257:
        DB      46      ;speech speed                DONE RB
      DB    Voice ;system pitch setting
        DB      131,164,95,149,123
      DB    FFH   ;end
;
Tbl3_say258:
        DB      55      ;speech speed      SEQ 4, AGE 2 DONE RB
        DB      Voice-4   ;system pitch setting
        DB      158,163,8,6
      DB    FFH   ;end
;
Tbl3_say259:
        DB      45      ;speech sp  d   SEQ 6, AGE 2  DONE RB
        DB      Voice+8   ;system pitch setting
        DB      119,35,70,81
      DB    FFH   ;end
;
Tbl3_say260:
        DB      46      ;speech speed       RB      DONE
```

```
        DB      Voice+8   ;system pitch setting   SEQ 1, AGE 3
        DB      119,66
      DB    FFH   ;end
;
Tbl3_say261:
        DB      46      ;speech speed   SEQ 4, AGE 3  RB DONE
        DB      Voice-3   ;system pitch setting
        DB      158,14,42
      DB    FFH   ;end
;
Tbl3_say262:
        DB      46      ;speech speed      SEQ 6 AGE 3   RB DONE
        DB      Voice-3   ;system pitch setting
        DB      119,35,5,93
      DB    FFH   ;end
;
Tbl3_say263:
        DB      60      ;speech speed    SEQ 2, AGE 1     RB DONE
        DB      Voice+8   ;system pitch setting
        DB      131,95,149
      DB    FFH   ;end
;
Tbl3_say264:
        DB      46      ;speech speed          RB DONE
        DB      Voice-4   ;system pitch setting
        DB      158,8,42
      DB    FFH   ;end
;
Tbl3_say265:    ;                                                RB DONE
        DB      46      ;speech speed
        DB      Voice-4   ;system pitch setting
        DB      119,35,70,93
      DB    FFH   ;end
;END GEORGE 07/08/98
;END LIGHT
;DARK
;GEORGE 07/08/98

Tbl3_say266:
        DB      52      ;speech speed          BEGIN LIGHT D. (DARKER)
        DB      Voice +8  ;system pitch setting      SEQ 1 AGE 1 RB DONE
        DB      119,10,162,6
      DB    FFH   ;end
;
Tbl3_say267:
        DB      46      ;speech speed          SEQ 2 AGE 1 DONE RB
        DB      Voice+8   ;system pitch setting
        DB      119,6,21
      DB    FFH   ;end
;
Tbl3_say268:
        DB      55      ;speech speed
        DB      Voice+8   ;system pitch setting SEQ 3 AGE 1 DONE RB
        DB      119,6,163,82,163,23
      DB    FFH   ;end
;
Tbl3_say269:
        DB      40      ;speech speed
        DB      Voice+8   ;system pitch setting  SEQ 4 AGE 1 DONE RB
        DB      158,101,163,104
```

```
        DB      FFH     ;               end
;
Tbl3_say270:
        DB      70      ;speech speed
        DB      Voice+8   ;system pitch setting
        DB      148,10,6,148
      DB    FFH   ;end
;
Tbl3_say271:
        DB      59      ;speech speed
        DB      Voice+4   ;system pitch setting
        DB      149,163,21,21   ;SEQ6 AGE4/SEQ14 AGE4 LIGHT js
        DB      FFH     ;end
;
Tbl3_say272:
        DB      52      ;speech speed
        DB      Voice+8   ;system pitch setting
        DB      119,35,162,10,5,81
        DB      FFH     ;end  DONE RB
;
Tbl3_say273:
        DB      60      ;speech speed
        DB      Voice+8   ;pitch control   ONE RB
        DB      63,163,149,163,163,51,35,152
      DB    FFH   ;end
;
Tbl3_say274:
        DB      52      ;speech speed
        DB      Voice+2   ;system pitch setting
        DB      119,60,6
      DB    FFH   ;end
;
Tbl3_say275:
        DB      52      ;speech speed
        DB      Voice+2   ;pitch control
        DB      119,60,45,85
        DB      FFH     ;end DONE RB
;
Tbl3_say276:
        DB      60      ;speech speed
        DB      Voice+2   ;system pitch setting    DONE RB
        DB      119,42,82,23
      DB    FFH   ;end
;
Tbl3_say277:
        DB      70      ;speech speed
        DB      Voice+2   ;system pitch setting
        DB      148,60,6,148
        DB      FFH     ;end   DONE RB
;
Tbl3_say278:
        DB      52      ;speech speed
        DB      Voice+2   ;system pitch setting  DONE RB
        DB      119,52,60,70,81
      DB    FFH   ;end
;
Tbl3_say279:
        DB      52      ;speech speed
      DB    Voice ;system pitch setting
        DB      119,10,42
```

```
        DB      FFH     ;end        DONE RB
;
Tbl3_say280:
        DB      52      ;speech speed
        DB      Voice   ;system pitch setting DONE RB
        DB      119,10,34,85
      DB    FFH   ;end
;
Tbl3_say281:
        DB      60      ;speech speed
      DB    Voice ;system pitch setting
        DB      119,42,83,23
        DB      FFH     ;end   DONE RB
;
Tbl3_say282:
        DB      52      ;speech speed
      DB    Voice ;system pitch setting
        DB      119,52,60,5,93
        DB      FFH     ;end   DONE RB
;
Tbl3_say283:
        DB      60      ;speech speed    !!NOTE!!  PRINTED TA  T HAD
WRONG WORD NUMBER FOR "KISS"
      DB    Voice ;system pitch setting
        DB      63,149,162,38,35,152
        DB      FFH     ;end   DONE RB
;
Tbl3_say284:
        DB      52      ;speech speed
      DB    Voice ;system pitch setting
        DB      119,60,42
        DB      FFH     ;end   DONE RB
;
Tbl3_say285:
        DB      52      ;speech speed
        DB      Voice-3   ;system pitch setting
        DB      119,60,34,85
      DB    FFH   ;end
;
Tbl3_say286:
        DB      60      ;speech speed
      DB    Voice ;system pitch setting
        DB      119,42,8?,68
      DB    FFH   ;end
;
Tbl3_say287:
        DB      70      ;speech speed
      DB    Voice ;system pitch setting
        DB      148,60,42,148
      DB    FFH   ;end
;
Tbl3_say288:
        DB      46      ;speech speed
      DB    Voice ;system pitch setting
        DB      119,163,52,60,70,93    ;SEQ7 AGE4/SEQ15 AGE 4 LIGHT js
      DB    FFH   ;end
;
Tbl3_say289:
        DB      50      ;speech speed
      DB    Voice ;system pitch setting
```

```
        DB      63,165,149,38,52,152    ;SEQ8 AGE4/SEQ16 AGE 4 LIGHT js
      DB    FFH   ;end

;END GEORGE 07/08/98
;END DARK
;SOUND
;
;start 43
Tbl3_say290:
        DB      50      ;speech speed
      DB    Voice ;system pitch setting
        DB      163,148,165,17          ;S1-A1/S9-A1/S1-A2 SOUND js
        DB      FFH     ;end            ;S9-A2/S1-A3/S9-A3 SOUND js
;
Tbl3_say291:
        DB      46      ;speech speed
      DB    Voice ;system pitch setting
        DB      85,165,165,165      ;S2-A1/S10-A1/S2-A2 SOUND js
        DB      165,165,140         ;S10-A2/S2-A3/S10-A3 SOUND js
        DB      FFH     ;end        ;S2-A4/S10-A4 SOUND js
;
Tbl3_say292:
        DB      50      ;speech speed
      DB    Voice ;system pitch setting
        DB      121,165,164,14,163,41,21    ;S3-A1/S11-A1 SOUND js
      DB    FFH   ;end
;
Tbl3_say293:
        DB      46      ;speech speed
      DB    Voice ;system pitch setting
        DB      163,129,164,5,162,41        ;S4-A1/S12-A1 SOUND js
      DB    FFH   ;end
;
Tbl3_say294:
        DB      46      ;speech speed
      DB    Voice ;system pitch setting
        DB      35,163,89   ;S5-A1/S13-A1 SOUND (with say/m2) js
      DB    FFH   ;end
;
Tbl3_say295:
        DB      53      ;speech speed
      DB    Voice ;system pitch setting
        DB      163,148,163,36  ;S6-A1/S14-A1/S6-A2 SOUND js
        DB      FFH     ;end    ;S14-A2/S6-A3/S14-A3 SOUND js
;
Tbl3_say296:
        DB      53      ;speech speed
      DB    Voice ;system pitch setting
        DB      17      ;S7-A1/S15-A1 SOUND (with say/m2) js
      DB    FFH   ;end
;
Tbl3_say297:
        DB      60      ;speech speed
      DB    Voice ;system pitch setting
        DB      122,164,21,164,21   ;S8-A1/S16-A1 SOUND js
        DB      FFH     ;end        ;S8-A3/S16-A3 SOUND js
;
Tbl3_say298:
        DB      46      ;speech speed
      DB    Voice ;system pitch setting
```

```
        DB      121,165,164,8,167,41,21      ;S3-A2/S11-A2 SOUND js
      DB    FFH   ;end
;
Tbl3_say299:
        DB      46      ;speech speed
      DB    Voice ;system pitch setting
        DB      163,129,164,5,165,73         ;S4-A2/S12-A2 SOUND js
      DB    FFH   ;end
;
Tbl3_say300:
        DB      46      ;speech speed
      DB    Voice ;system pitch setting
        DB      35,165,31       ;S5-A2/S13-A2/S5-A3 SOUND (with say/m2)
js
        DB      FFH     ;end    ;S13-A3/S5-A4/S13-A4 SOUND (with say/m2)
js
;
Tbl3_say301:
        DB      46      ;speech speed
      DB    Voice ;system pitch setting
        DB      8,162,41,163,85          ;S7-A2/S15-A2 SOUND (with
say/m2) js
      DB    FFH   ;end
;
Tbl3_say302:
        DB      60      ;speech speed
      DB    Voice ;system pitch setting
        DB      122,164,21       ;S8-A2/S16-A2 SOUND js
      DB    FFH   ;end
;
Tbl3_say303:
        DB      46      ;speech speed
      DB    Voice ;system pitch setting
        DB      121,165,164,14,163,73,21         ;S3-A3/S11-A3 SOUND js
        DB      FFH     ;end
;
Tbl3_say304:
        DB      46      ;speech speed
      DB    Voice ;system pitch setting
        DB      163,129,164,35,165,44        ;S4-A3/S12-A3 SOUND js
        DB      FFH     ;end                 ;S4-A4/S12-A4 SOUND js
;
Tbl3_say305:
        DB      46      ;speech speed
      DB    Voice ;system pitch setting
        DB      8,73,164,85     ;S7-A3/S15-A3 SOUND (with say/m2)js
        DB      FFH     ;end    ;S7-A4/S15-A4 SOUND (with say/m2)js
;
Tbl3_say306:
        DB      55      ;speech speed
      DB    Voice ;system pitch setting
        DB      164,148,164,163,46           ;S1-A4/S9-A4 SOUND js
      DB    FFH   ;end
;
Tbl3_say307:
        DB      46      ;speech speed
      DB    Voice ;system pitch setting
        DB      121,165,164,8,163,73,21         ;S3-A4/S11-A4 SOUND js
      DB    FFH   ;end
;
```

```
Tbl3_say308:
        DB      55      ;speech speed
      DB    Voice ;system pitch setting
        DB      164,148,164,163,54          ;S6-A4/S14-A4 SOUND js
      DB    FFH   ;end
;
Tbl3_say309:
        DB      60      ;speech speed
      DB    Voice ;system pitch setting
        DB      122,164,163,88,164,21       ;S8-A4/S16-A4 SOUND js
      DB    FFH   ;end
;
;
;END SOUND
;
;TILT
;GEORGE 07/09/98
Tbl3_say310:
        DB      56      ;speech speed
        DB      Voice+8   ;pitch control
        DB      160             ;S1 A1 TILT/S4 A1 TILT/S14 A1 TILT js
      DB    FFH   ;end
;
Tbl3_say311:
        DB      46      ;speech speed
        DB      Voice     ;pitch control
        DB      157,36          ;S2 A1 TILT js
      DB    FFH   ;end
;
Tbl3_say312:
        DB      46      ;speech speed
      DB    Voice ;system pitch setting
        DB      158,9           ;S3 A1 TILT js
      DB    FFH   ;end
;
Tbl3_say313:
        DB      46      ;speech speed
        DB      Voice+8   ;system pitch setting
        DB      154             ;S5 A1/S4 A2/S2 A3/S2 A4 TILT js
      DB    FFH   ;end
;
Tbl3_say314:
        DB      46      ;speech speed
      DB    Voice ;system pitch setting
        DB      159,82,39       ;S6 A1 TILT js
      DB    FFH   ;end
;
Tbl3_say315:
        DB      46      ;speech speed
      DB    Voice ;system pitch setting
        DB      155,39,39       ;S7 A1 TILT/S6 A2 TILT js
      DB    FFH   ;end
;
Tbl3_say316:
        DB      46      ;speech speed
      DB    Voice ;system pitch setting
        DB      37,152          ;S8 A1 TILT (with say/m5) js
      DB    FFH   ;end
;
Tbl3_say317:
```

```
        DB      46      ;speech speed
      DB    Voice ;system pitch setting
        DB      154,120             ;S9 A1 TILT/S9 A2 TILT js
      DB    FFH   ;end
;
Tbl3_say318:
        DB      46      ;speech speed
      DB    Voice ;system pitch setting
        DB      155,120,120     ;S10 A1 TILT/S10 A2 TILT js
      DB    FFH   ;end
;
Tbl3_say319:
        DB      46      ;speech speed
      DB    Voice ;system pitch setting
        DB      35,57           ;S11 A1 TILT (with say/m2) js
      DB    FFH   ;end
;
Tbl3_say320:
        DB      48      ;speech speed
      DB    Voice ;system pitch setting
        DB      158,10,80       ;S12 A1 TILT js
      DB    FFH   ;end
;
Tbl3_say321:
        DB      46      ;speech speed
      DB    Voice ;system pitch setting
        DB      119,160             ;S13 A1 / S15 A3 TILT js
      DB    FFH   ;end
;
Tbl3_say322:
        DB      46      ;speech speed
      DB    Voice ;system pitch setting
        DB      160,9               ;S15 A1 TILT js
      DB    FFH   ;end
;
Tbl3_say323:
        DB      46      ;speech speed
      DB    Voice ;system pitch setting
        DB      154,149     ;S16 A1 / S15 A2 / S13 A3 TILT js
      DB    FFH   ;end
;
Tbl3_say324:
        DB      46      ;speech speed
      DB    Voice ;system pitch setting
        DB      160             ;S1 A2/S3 A2/S1 A3/S1 A4 TILT js
      DB    FFH   ;end
;
Tbl3_say325:
        DB      46      ;speech speed
      DB    Voice ;system pitch setting
        DB      52,9            ;S2 A1 TILT (with say/m16) js
      DB    FFH   ;end
;
Tbl3_say326:
        DB      46      ;speech speed
      DB    Voice ;system pitch setting
        DB      159,83,39       ;S5 A2 TILT js
      DB    FFH   ;end
;
Tbl3_say327:
```

```
        DB      46      ;speech speed
      DB    Voice ;system pitch setting
        DB      52,48,81,152        ;S7 A2 TILT (with say/m5) js
      DB    FFH   ;end
;
Tbl3_say328:
        DB      46      ;speech speed
      DB    Voice ;system pitch setting
        DB      155             ;S8 A2 TILT (with say/m5) js
      DB    FFH   ;end
;
Tbl3_say329:
        DB      46      ;speech speed
      DB    Voice ;system pitch setting
        DB      52,57           ;S11 A2 TILT (with say/m2) js
      DB    FFH   ;end
;
Tbl3_say330:
        DB      46      ;speech speed
      DB    Voice ;system pitch setting
        DB      158,60,80       ;S12 A2 TILT js
      DB    FFH   ;end
;
Tbl3_say331:
        DB      46      ;speech speed
      DB    Voice ;system pitch setting
        DB      163,156             ;S13 A2  TILT (with say/m5) js
      DB    FFH   ;end
;
Tbl3_say332:
        DB      46      ;speech speed
      DB    Voice ;system pitch setting
        DB      8,22,85         ;S14 A2 TILT js
      DB    FFH   ;end
;
Tbl3_say333:
        DB      46      ;speech speed
      DB    Voice ;pitch control
        DB      154,118,163,145,165,162,118     ;S16 A2/S14 A3/S14 A4
TILT js
      DB    FFH   ;end
;
Tbl3_say334:
        DB      46      ;speech speed
      DB    Voice ;system pitch setting
        DB      159                     ;S3 A3 TILT js
      DB    FFH   ;end
;
Tbl3_say335:
        DB      46      ;speech speed
      DB    Voice ;pitch control
        DB      83,1            ;S4 A3/S4 A4 TILT (with say/m26) js
      DB    FFH   ;end
;
Tbl3_say336:
        DB      46      ;speech speed
      DB    Voice ;system pitch setting
        DB      155,52,62,85            ;S5 A3 TILT js
      DB    FFH   ;end
;
```

```
Tbl3_say337:
        DB      50      ;speech speed
      DB    Voice ;system pitch setting
        DB      52,48,93,152        ;S6 A3 TILT (with say/m5) js
      DB    FFH   ;end
;
Tbl3_say338:
        DB      46      ;speech speed
      DB    Voice ;system pitch setting
        DB      155             ;S7 A3/S7 A4 TILT (with say/m5) js
      DB    FFH   ;end
;
Tbl3_say339:
        DB      46      ;speech speed
      DB    Voice ;system pitch setting
        DB      155,120,163,149     ;S8 A3/S8 A4 TILT js
      DB    FFH   ;end
;
Tbl3_say340:
        DB      46      ;speech speed
      DB    Voice ;system pitch setting
        DB      165,129         ;S9 A3/S9 A4 TILT (with say/m9) js
      DB    FFH   ;end
;
Tbl3_say341:
        DB      46      ;speech speed
      DB    Voice ;system pitch setting
        DB      160,163,120,120     ;S10 A3/S10 A4 TILT (with say/m16)js
      DB    FFH   ;end
;
Tbl3_say342:
        DB      46      ;speech speed
      DB    Voice ;system pitch setting
        DB      163,23          ;S11 A3/S15 A4 TILT (with say/m2&21) js
      DB    FFH   ;end
;
Tbl3_say343:
        DB      55      ;speech speed
      DB    Voice ;system pitch setting
        DB      164,156             ;S12 A3 TILT (with say/m5) js
      DB    FFH   ;end
;
Tbl3_say344:
        DB      46      ;speech speed
      DB    Voice ;system pitch setting
        DB      163,1,163,1,117     ;S16 A3 TILT (with say/m5) js
      DB    FFH   ;end
;
Tbl3_say345:
        DB      46      ;speech speed
      DB    Voice ;system pitch setting
        DB      27,162,149          ;S3 A4 TILT (with say/m26) js
      DB    FFH   ;end
;
Tbl3_say346:
        DB      46      ;speech speed
      DB    Voice ;system pitch setting
        DB      155,52,29,163,85        ;S5 A4 TILT js
      DB    FFH   ;end
;
```

```
Tbl3_say347:
        DB      46      ;speech speed
      DB    Voice ;system pitch setting
        DB      52,47,93,164,152        ;S6 A4 TILT (with say/m5) js
      DB    FFH   ;end
;
Tbl3_say348:
        DB      46      ;speech speed
      DB    Voice ;system pitch setting
        DB      52,24,68                ;S11 A4 TILT (with say/m2) js
      DB    FFH   ;end
;
Tbl3_say349:
        DB      46      ;speech speed
      DB    Voice ;system pitch setting
        DB      22,149          ;S13 A4 TILT (with say/m5)  js
      DB    FFH   ;end
;
Tbl3_say350:
        DB      46      ;speech speed
      DB    Voice ;system pitch setting
        DB      163,1,163,39,163,117        ;S16 A4 TILT (with say/m5) js
      DB    FFH   ;end

;END GEORGE 07/09/98
;
;GEORGE
;IR 07/09/98
Tbl3_say351:
        DB      46      ;speech speed
        DB      Voice+8   ;pitch control
        DB      40        ;SEQ1,seq2,seq3,seq4 ir age 1
      DB    FFH   ;end
;
Tbl3_say352:
        DB      46      ;speech speed
        DB      Voice      ;pitch control
        DB      66,162,85     ;seq5, ir age1
      DB    FFH   ;end
;
Tbl3_say353:
        DB      46      ;speech speed
      DB    Voice ;system pitch setting
        DB      19,85   ;seq6, ir age1          DANCE WAH
      DB    FFH   ;end
;
Tbl3_say354:
        DB      46      ;speech speed
        DB      Voice+8   ;system pitch setting
        DB      162,164,134,134  ;seq6, ir age1 DO DO DO
      DB    FFH   ;end
;
Tbl3_say355:
        DB      46      ;speech speed
        DB      Voice+2   ;system pitch setting
        DB      134,134,25,19   ;seq7 ir age1
      DB    FFH   ;end
;
Tbl3_say356:
        DB      50      ;speech speed
```

```
        DB      Voice+8   ;system pitch setting
        DB      162
        DB      FFH     ;end          EMPTY SPACE
;
Tbl3_say357:
        DB      42      ;speech speed
        DB      Voice   ;system pitch setting
        DB      102,97,118,34       ;seq8 ir age1
      DB    FFH   ;end
;
Tbl3_say358:
        DB      50      ;speech speed
      DB    Voice ;system pitch setting
        DB      117,34,22       ;seq9 ir age1
      DB    FFH   ;end
;
Tbl3_say359:
        DB      50      ;speech speed
      DB    Voice ;system pitch setting
        DB      34,78,145,145   ;seq10,11 ir age1
      DB    FFH   ;end
;
Tbl3_say360:
        DB      50      ;speech speed
      DB    Voice ;system pitch setting
        DB      150,151,93,71   ;seq12 ir age1 TWINKLE
      DB    FFH   ;end
;
Tbl3_say361:
        DB      46      ;speech speed
      DB    Voice ;system pitch setting
        DB      91,31,165,165,165,165,165,165,128,31       ;seq13,14 ir
age1
      DB    FFH   ;end
;
Tbl3_say362:
        DB      46      ;speech speed
      DB    Voice ;system pitch setting
        DB      161,72,161      ;seq15 ir age1
      DB    FFH   ;end
;
Tbl3_say363:
        DB      60      ;speech speed
      DB    Voice ;system pitch setting
        DB      144,144,144,144                 ;seq16 ir age1
      DB    FFH   ;end
;
Tbl3_say364:
        DB      46      ;speech speed
        DB      Voice+5   ;system pitch setting
        DB      81,40   ;seq1,2,3 ir age2
      DB    FFH   ;end
;
Tbl3_say365:
        DB      46      ;speech speed
        DB      Voice+8   ;system pitch setting
        DB      81,40    ;seq4,5 ir age2
      DB    FFH   ;end
;
Tbl3_say366:
```

```
        DB      46      ;speech speed
        DB      Voice+8   ;system pitch setting
        DB      66,159  ;seq6 ir age2
      DB    FFH   ;end
;
Tbl3_say367:
        DB      46      ;speech speed
        DB      Voice+7   ;system pitch setting
        DB      19,165,165,165,164,85,134,165,135          ;seq7,8 ir
age2
      DB    FFH   ;end
;
Tbl3_say368:
        DB      46      ;speech speed
        DB      Voice+3  ;system pitch setting
        DB      118,25,34                                  ;seq9 ir age2
      DB    FFH   ;end
;
Tbl3_say369:
        DB      51      ;speech speed
        DB      Voice+8   ;system pitch setting
        DB      102,97,118              ;seq10 ir age2
      DB    FFH   ;end
;
Tbl3_say370:
        DB      46      ;speech speed
        DB      Voice+5   ;system pitch setting
        DB      117,34,22               ;SEQ11 ir age2
      DB    FFH   ;end
;
Tbl3_say371:
        DB      48      ;speech speed
      DB    Voice ;system pitch setting
        DB      91,31,165,165,165,165,165,165,124,31    ;seq13,14 ir
age2
      DB    FFH   ;end
;
Tbl3_say372:
        DB      55      ;speech speed
      DB    Voice ;system pitch setting
        DB      161,72,161                                 ;seq15 ir age2
      DB    FFH   ;end
;
Tbl3_say373:
        DB      50      ;speech speed
      DB    Voice ;system pitch setting
        DB      143,144,143                        ;seq16 ir age2
      DB    FFH   ;end
;
Tbl3_say374:
        DB      50      ;speech speed
      DB    Voice ;pitch control
        DB      14,40           ;seq1,2,3,4,5 ir age3
      DB    FFH   ;end
;
Tbl3_say375:
        DB      46      ;speech speed
        DB      Voice+5   ;system pitch setting
        DB      35,48,66        ;seq6 ir age3
      DB    FFH   ;end
```

```
;
Tbl3_say376:
        DB      50      ;speech speed
        DB      Voice+8   ;pitch control
        DB      19,12,134,134    :seq7,8 ir age3
      DB    FFH   ;end
;
Tbl3_say377:
        DB      46      ;speech speed
        DB      Voice+3   ;system pitch setting
        DB      34,85,99        ;SEQ9 ir age3
      DB    FFH   ;end
;
Tbl3_say378:
        DB      46      ;speech speed
        DB      Voice+2   ;system pitch setting
        DB      156,25,34       ;seq11 ir age3
      DB    FFH   ;end
;
Tbl3_say379:
        DB      50      ;speech speed
        DB      Voice+3   ;system pitch setting
        DB      63,165,165,165,165,165,124,31    ;seq13,14 ir age3
      DB    FFH   ;end
;
Tbl3_say380:
        DB      70      ;speech speed
        DB      Voice+4   ;system pitch setting
        DB      35,72,162,162,162,162,162,162,162,162,162,162,161
      DB    FFH   ;end
;
Tbl3_say381:
        DB      58      ;speech speed
        DB      Voice+5   ;system pitch setting
        DB      40,85   ;SEQ1,2,3,4,5 IR AGE4
      DB    FFH   ;end
;
Tbl3_say382:
        DB      46      ;speech speed
        DB      Voice+6   ;system pitch setting
        DB      81,66,21        ;seq6 ir age4
      DB    FFH   ;end
;
Tbl3_say383:
        DB      46      ;speech speed
        DB      Voice+7   ;system pitch setting
        DB      134,134,25,19   ;seq7,8 ir age4
      DB    FFH   ;end
;
Tbl4_say384:
        DB      50      ;speech speed
        DB      Voice+8   ;system pitch setting
        DB      34,78,145,145   ;seq9 ir age4
      DB    FFH   ;end
;
Tbl4_say385:
        DB      50      ;speech speed
        DB      Voice+8   ;system pitch setting
        DB      119,44,52,71,150        seq10 ir age4
        DB      FFH     ;end   SAY NUMBERS MODIFIED TO MATCH CORRECT
```

```
DIALOGUE

;
Tbl4_say386:
        DB      46      ;speech speed
        DB      Voice+8   ;system pitch setting
        DB      34,85,99        seq11 ir age4
      DB    FFH   ;end
;
Tbl4_say387:
        DB      50      ;speech speed
        DB      Voice+1   ;system pitch setting
        DB      119,124,31      ;seq12 ir age4
      DB    FFH   ;end
;
Tbl4_say388:
        DB      56      ;speech speed
        DB      Voice+3   ;system pitch setting
        DB      162,63      ;seq14 ir age4
      DB    FFH   ;end
;
Tbl4_say389:
        DB      60      ;speech speed
        DB      Voice-8   ;system pitch setting
        DB      161,164,161   ;SEQ10 HANGING (YAWN)
;        DB      46      ;speech speed
;        DB      Voice+3   ;system pitch setting
;        DB      161,144,144     ;seq15 ir age4
      DB    FFH   ;end
;
;Tbl1_say41:
;        DB      46      ;speech speed
;        DB      Voice+4   ;system pitch setting
;        DB      143,144,143   ;seq16 ir age4
;        DB      FFH     ;end
;
;Tbl1_say42:
;        DB      46      ;speech speed
;        DB      Voice   ;system pitch setting
;        DB      4
;        DB      FFH     ;end
;
;
Tbl4_say390:
        DB      55      ;speech speed
        DB      Voice+3   ;system pitch setting
        DB      165,165,144,165,144,165,144,165,144
      DB    FFH   ;end
;END IR
;END GEORGE

; ADDED BY DMH (FOR FURBY SAYS)
Tbl4_say391:
        DB      46      ;speech speed
        DB      Voice   ;system pitch setting
        DB      42      ; LIGHT (FURBY SAYS)
        DB      FFH     ;end

; ADDED BY DMH (FOR FURBY SAYS)
```

```
Tbl4_say392:
        DB      52      ;speech speed
      DB    Voice ;system pitch setting
        DB      60,42   ;no light
        DB      FFH     ;end
;
Tbl4_say393:
        DB      55      ;speech speed
      DB    Voice ;system pitch setting
        DB      164,163,46 ; LOUD SOUND
      DB    FFH   ;end
;
;
Tbl4_say394:                    ; DMH
        DB      46      ;speech speed
        DB      Voice   ;system pitch setting
        DB      164,163,44      ; LISTEN (FURBY SAYS)
        DB      FFH     ;end
;
Tbl4_say395:
        DB      46      ;speech speed
        DB      Voice   ;system pitch setting
        DB      52,163  ;(ME) with names (dmh)
        DB      FFH     ;end
;
Tbl4_say396:
        DB      56      ;speech speed
        DB      Voice   ;system pitch setting
        DB      162,55  ;name (MEE MEE) (dmh)
      DB    FFH   ;end
;
Tbl4_say397:
        DB      58      ;speech speed
        DB      Voice   ;system pitch setting
      DB    163,23      ;(DO MOH)
      DB    FFH   ;end
;
Tbl4_say398:
        DB      60      ;speech speed
        DB      Voice   ;system pitch setting
        DB      80      ;TOH-LOO
      DB    FFH   ;end
;
Tbl4_say399:
        DB      60      ;speech speed
        DB      Voice   ;system pitch setting
        DB      165     ; DELAY 1 SECOND DMH
      DB    FFH   ;end
;
; start of diagnostic tables dmh
Tbl4_say400:
        DB      0               ;speech speed
        DB      Voice+16        ;system pitch setting
        DB      168,168,168     ; used at start of diagnostics
      DB    FFH   ;end
;
Tbl4_say401:
        DB      20              ;speech speed
        DB      Voice+13        ;system pitch setting
        DB      169,165         ;key beep
```

```
;       DB      1
        DB      FFH     ;end
;
Tbl4_say402:
          DB        20                      ;speech speed
        DB      Voice+5                 ;system pitch setting
        DB      169,163,169,163,169       ;pass test
;       DB      2
        DB      FFH     ;end
;
Tbl4_say403:
          DB        96                      ;speech speed
          DB        Voice-40                ;system pitch setting
          DB        169,163                 ;fail test tone
        DB      FFH     ;end
;
Tbl4_say404:
          DB        46                      ;speech speed
        DB      Voice                   ;system pitch setting
          DB        169                     ;speaker tone test
        DB      FFH                     ;end
;
Tbl4_say405:
          DB        46      ;speech speed
        DB      Voice ;system pitch setting
        DB      163     ; no sound for start of motor cal
        DB      FFH     ;end
;
Tbl4_say406:
          DB        20                      ;speech speed
        DB      Voice+5                 ;system pitch setting
        DB      169,163,169,163,169       ;feed1
        DB      FFH     ;end
;
Tbl4_say407:
          DB        20      ;speech speed
        DB      Voice+5     ;system pitch setting
        DB      169,163,169,163,169       ;pass feed sw
        DB      FFH     ;end
;
Tbl4_say408:
          DB        20      ;speech speed
        DB      Voice+5     ;system pitch setting
        DB      169,163,169,163,169       ;pass light test
        DB      FFH     ;end
;
Tbl4_say409:
          DB        20      ;speech speed
        DB      Voice+5     ;system pitch setting
        DB      169,163,169,163,169       ;pass sound test
        DB      FFH     ;end
;
Tbl4_say410:
          DB        20      ;speech speed
        DB      Voice+5     ;system pitch setting
        DB      169,163,169,163,169       ;pass all test complete
        DB      159
        DB      FFH     ;end
;
Tbl4_say411:
```

```
        DB      60      ;speech speed   ; HIDE ME (HIDE AND SEEK) DHM
        DB      Voice+3   ;system pitch setting
        DB      31,52                   ; HIDE ME
      DB    FFH   ;end
;
Tbl4_say412:
        DB      100     ;speech speed
      DB    Voice ;system pitch setting
        DB      167,167,167     ;SEQ1 FEED AGE1 (AAAAH)
      DB    FFH   ;end
;
Tbl4_say413:
;
Tbl4_say414:
;
Tbl4_say415:
;
Tbl4_say416:
;
Tbl4_say417:
;
Tbl4_say418:
;
Tbl4_say419:
;
Tbl4_say420:
;
Tbl4_say421:
;
Tbl4_say422:
;
Tbl4_say423:
;
Tbl4_say424:
;
Tbl4_say425:
;
Tbl4_say426:
;
Tbl4_say427:
;
Tbl4_say428:
;
Tbl4_say429:
;
Tbl4_say430:
;
Tbl4_say431:
;
Tbl4_say432:
;
Tbl4_say433:
;
Tbl4_say434:
;
Tbl4_say435:
;
Tbl4_say436:
;
Tbl4_say437:
```

```
;
Tbl4_say438:
;
Tbl4_say439:
;
Tbl4_say440:
;
Tbl4_say441:
;
Tbl4_say442:
;
Tbl4_say443:
;
Tbl4_say444:
;
Tbl4_say445:
;
Tbl4_say446:
;
Tbl4_say447:
;
Tbl4_say448:
;
Tbl4_say449:
;
Tbl4_say450:
;
Tbl4_say451:
;
Tbl4_say452:
;
Tbl4_say453:
;
Tbl4_say454:
;
Tbl4_say455:
;
Tbl4_say456:
;
Tbl4_say457:
;
Tbl4_say458:
;
Tbl4_say459:
;
Tbl4_say460:
;
Tbl4_say461:
;
Tbl4_say462:
;
Tbl4_say463:
;
Tbl4_say464:
;
Tbl4_say465:
;
Tbl4_say466:
;
Tbl4_say467:
```

```
;
Tbl4_say468:
;
Tbl4_say469:
;
Tbl4_say470:
;
Tbl4_say471:
;
Tbl4_say472:
;
Tbl4_say473:
;
Tbl4_say474:
;
Tbl4_say475:
;
Tbl4_say476:
;
Tbl4_say477:
;
Tbl4_say478:
;
Tbl4_say479:
;
Tbl4_say480:
;
Tbl4_say481:
;
Tbl4_say482:
;
Tbl4_say483:
;
Tbl4_say484:
;
Tbl4_say485:
;
Tbl4_say486:
;
Tbl4_say487:
;
Tbl4_say488:
;
Tbl4_say489:
;
Tbl4_say490:
;
Tbl4_say491:
;
Tbl4_say492:
;
Tbl4_say493:
;
Tbl4_say494:
;
Tbl4_say495:
;
Tbl4_say496:
;
Tbl4_say497:
```

```
;
Tbl4_say498:
;
Tbl4_say499:
;
Tbl4_say500:
;
Tbl4_say501:
;
Tbl4_say502:
;
Tbl4_say503:
;
Tbl4_say504:
;
Tbl4_say505:
;
Tbl4_say506:
;
Tbl4_say507:
;
Tbl4_say508:
;
Tbl4_say509:
;
Tbl4_say510:
;
Tbl4_say511:
; ON POWER UP, UNTIL WAKE-UP TABLE INSTALLED (Dave)
        DB      46      ;speech speed
        DB      Voice
        DB      165
        DB      FFH     ;end

;

;*****************************************************************
;*****************************************************************
;*****************************************************************
;*****************************************************************

; Motor tables

; Offsett pointer :

Motor_grp1:

        DW      Tbl1_M000
        DW      Tbl1_M001,Tbl1_M002,Tbl1_M003,Tbl1_M004,Tbl1_M005
        DW      Tbl1_M006,Tbl1_M007,Tbl1_M008,Tbl1_M009,Tbl1_M010
        DW      Tbl1_M011,Tbl1_M012,Tbl1_M013,Tbl1_M014,Tbl1_M015
        DW      Tbl1_M016,Tbl1_M017,Tbl1_M018,Tbl1_M019,Tbl1_M020
        DW      Tbl1_M021,Tbl1_M022,Tbl1_M023,Tbl1_M024,Tbl1_M025
        DW      Tbl1_M026,Tbl1_M027,Tbl1_M028,Tbl1_M029,Tbl1_M030
        DW      Tbl1_M031,Tbl1_M032,Tbl1_M033,Tbl1_M034,Tbl1_M035
        DW      Tbl1_M036,Tbl1_M037,Tbl1_M038,Tbl1_M039,Tbl1_M040
        DW      Tbl1_M041,Tbl1_M042,Tbl1_M043,Tbl1_M044,Tbl1_M045
        DW      Tbl1_M046,Tbl1_M047,Tbl1_M048,Tbl1_M049,Tbl1_M050
        DW      Tbl1_M051,Tbl1_M052,Tbl1_M053,Tbl1_M054,Tbl1_M055
        DW      Tbl1_M056,Tbl1_M057,Tbl1_M058,Tbl1_M059,Tbl1_M060
```

```
        DW      Tbl1_M061,Tbl1_M062,Tbl1_M063,Tbl1_M064,Tbl1_M065
        DW      Tbl1_M066,Tbl1_M067,Tbl1_M068,Tbl1_M069,Tbl1_M070
        DW      Tbl1_M071,Tbl1_M072,Tbl1_M073,Tbl1_M074,Tbl1_M075
        DW      Tbl1_M076,Tbl1_M077,Tbl1_M078,Tbl1_M079,Tbl1_M080
        DW      Tbl1_M081,Tbl1_M082,Tbl1_M083,Tbl1_M084,Tbl1_M085
        DW      Tbl1_M086,Tbl1_M087,Tbl1_M088,Tbl1_M089,Tbl1_M090
        DW      Tbl1_M091,Tbl1_M092,Tbl1_M093,Tbl1_M094,Tbl1_M095
        DW      Tbl1_M096,Tbl1_M097,Tbl1_M098,Tbl1_M099
      DW    Tbl1_M100,Tbl1_M101,Tbl1_M102,Tbl1_M103,Tbl1_M104
      DW    Tbl1_M105,Tbl1_M106,Tbl1_M107,Tbl1_M108,Tbl1_M109
      DW    Tbl1_M110,Tbl1_M111,Tbl1_M112,Tbl1_M113,Tbl1_M114
      DW    Tbl1_M115,Tbl1_M116,Tbl1_M117,Tbl1_M118,Tbl1_M119
      DW    Tbl1_M120,Tbl1_M121,Tbl1_M122,Tbl1_M123,Tbl1_M124
      DW    Tbl1_M125,Tbl1_M126,Tbl1_M127
;
Motor_grp2:

      DW    Tbl2_M128
      DW    Tbl2_M129,Tbl2_M130,Tbl2_M131,Tbl2_M132,Tbl2_M133
      DW    Tbl2_M134,Tbl2_M135,Tbl2_M136,Tbl2_M137,Tbl2_M138
      DW    Tbl2_M139,Tbl2_M140,Tbl2_M141,Tbl2_M142,Tbl2_M143
      DW    Tbl2_M144,Tbl2_M145,Tbl2_M146,Tbl2_M147,Tbl2_M148
      DW    Tbl2_M149,Tbl2_M150,Tbl2_M151,Tbl2_M152,Tbl2_M153
      DW    Tbl2_M154,Tbl2_M155,Tbl2_M156,Tbl2_M157,Tbl2_M158
      DW    Tbl2_M159,Tbl2_M160,Tbl2_M161,Tbl2_M162,Tbl2_M163
      DW    Tbl2_M164,Tbl2_M165,Tbl2_M166,Tbl2_M167,Tbl2_M168
      DW    Tbl2_M169,Tbl2_M170,Tbl2_M171,Tbl2_M172,Tbl2_M173
      DW    Tbl2_M174,Tbl2_M175,Tbl2_M176,Tbl2_M177,Tbl2_M178
      DW    Tbl2_M179,Tbl2_M180,Tbl2_M181,Tbl2_M182,Tbl2_M183
      DW    Tbl2_M184,Tbl2_M185,Tbl2_M186,Tbl2_M187,Tbl2_M188
      DW    Tbl2_M189,Tbl2_M190,Tbl2_M191,Tbl2_M192,Tbl2_M193
      DW    Tbl2_M194,Tbl2_M195,Tbl2_M196,Tbl2_M197,Tbl2_M198
      DW    Tbl2_M199,Tbl2_M200,Tbl2_M201,Tbl2_M202,Tbl2_M203
      DW    Tbl2_M204,Tbl2_M205,Tbl2_M206,Tbl2_M207,Tbl2_M208
      DW    Tbl2_M209,Tbl2_M210,Tbl2_M211,Tbl2_M212,Tbl2_M213
      DW    Tbl2_M214,Tbl2_M215,Tbl2_M216,Tbl2_M217,Tbl2_M218
      DW    Tbl2_M219,Tbl2_M220,Tbl2_M221,Tbl2_M222,Tbl2_M223
      DW    Tbl2_M224,Tbl2_M225,Tbl2_M226,Tbl2_M227,Tbl2_M228
      DW    Tbl2_M229,Tbl2_M230,Tbl2_M231,Tbl2_M232,Tbl2_M233
      DW    Tbl2_M234,Tbl2_M235,Tbl2_M236,Tbl2_M237,Tbl2_M238
      DW    Tbl2_M239,Tbl2_M240,Tbl2_M241,Tbl2_M242,Tbl2_M243
      DW    Tbl2_M244,Tbl2_M245,Tbl2_M246,Tbl2_M247,Tbl2_M248
      DW    Tbl2_M249,Tbl2_M250,Tbl2_M251,Tbl2_M252,Tbl2_M253
      DW    Tbl2_M254,Tbl2_M255
;
;
Motor_grp3:

      DW    Tbl3_M256
      DW    Tbl3_M257,Tbl3_M258,Tbl3_M259,Tbl3_M260,Tbl3_M261
      DW    Tbl3_M262,Tbl3_M263,Tbl3_M264,Tbl3_M265,Tbl3_M266
      DW    Tbl3_M267,Tbl3_M268,Tbl3_M269,Tbl3_M270,Tbl3_M271
      DW    Tbl3_M272,Tbl3_M273,Tbl3_M274,Tbl3_M275,Tbl3_M276
      DW    Tbl3_M277,Tbl3_M278,Tbl3_M279,Tbl3_M280,Tbl3_M281
      DW    Tbl3_M282,Tbl3_M283,Tbl3_M284,Tbl3_M285,Tbl3_M286
      DW    Tbl3_M287,Tbl3_M288,Tbl3_M289,Tbl3_M290,Tbl3_M291
      DW    Tbl3_M292,Tbl3_M293,Tbl3_M294,Tbl3_M295,Tbl3_M296
      DW    Tbl3_M297,Tbl3_M298,Tbl3_M299,Tbl3_M300,Tbl3_M301
      DW    Tbl3_M302,Tbl3_M303,Tbl3_M304,Tbl3_M305,Tbl3_M306
      DW    Tbl3_M307,Tbl3_M308,Tbl3_M309,Tbl3_M310,Tbl3_M311
```

```
        DW      Tbl3_M312,Tbl3_M313,Tbl3_M314,Tbl3_M315,Tbl3_M316
        DW      Tbl3_M317,Tbl3_M318,Tbl3_M319,Tbl3_M320,Tbl3_M321
        DW      Tbl3_M322,Tbl3_M323,Tbl3_M324,Tbl3_M325,Tbl3_M326
        DW      Tbl3_M327,Tbl3_M328,Tbl3_M329,Tbl3_M330,Tbl3_M331
        DW      Tbl3_M332,Tbl3_M333,Tbl3_M334,Tbl3_M335,Tbl3_M336
        DW      Tbl3_M337,Tbl3_M338,Tbl3_M339,Tbl3_M340,Tbl3_M341
        DW      Tbl3_M342,Tbl3_M343,Tbl3_M344,Tbl3_M345,Tbl3_M346
        DW      Tbl3_M347,Tbl3_M348,Tbl3_M349,Tbl3_M350,Tbl3_M351
        DW      Tbl3_M352,Tbl3_M353,Tbl3_M354,Tbl3_M355,Tbl3_M356
        DW      Tbl3_M357,Tbl3_M358,Tbl3_M359,Tbl3_M360,Tbl3_M361
        DW      Tbl3_M362,Tbl3_M363,Tbl3_M364,Tbl3_M365,Tbl3_M366
        DW      Tbl3_M367,Tbl3_M368,Tbl3_M369,Tbl3_M370,Tbl3_M371
        DW      Tbl3_M372,Tbl3_M373,Tbl3_M374,Tbl3_M375,Tbl3_M376
        DW      Tbl3_M377,Tbl3_M378,Tbl3_M379,Tbl3_M380,Tbl3_M381
        DW      Tbl3_M382,Tbl3_M383
;
;
Motor_grp4:
        DW      Tbl4_M384
        DW      Tbl4_M385,Tbl4_M386,Tbl4_M387,Tbl4_M388,Tbl4_M389
        DW      Tbl4_M390,Tbl4_M391,Tbl4_M392,Tbl4_M393,Tbl4_M394
        DW      Tbl4_M395,Tbl4_M396,Tbl4_M397,Tbl4_M398,Tbl4_M399
        DW      Tbl4_M400,Tbl4_M401,Tbl4_M402,Tbl4_M403,Tbl4_M404
        DW      Tbl4_M405,Tbl4_M406,Tbl4_M407,Tbl4_M408,Tbl4_M409
        DW      Tbl4_M410,Tbl4_M411,Tbl4_M412,Tbl4_M413,Tbl4_M414
        DW      Tbl4_M415,Tbl4_M416,Tbl4_M417,Tbl4_M418,Tbl4_M419
        DW      Tbl4_M420,Tbl4_M421,Tbl4_M422,Tbl4_M423,Tbl4_M424
        DW      Tbl4_M425,Tbl4_M426,Tbl4_M427,Tbl4_M428,Tbl4_M429
        DW      Tbl4_M430,Tbl4_M431,Tbl4_M432,Tbl4_M433,Tbl4_M434
        DW      Tbl4_M435,Tbl4_M436,Tbl4_M437,Tbl4_M438,Tbl4_M439
        DW      Tbl4_M440,Tbl4_M441,Tbl4_M442,Tbl4_M443,Tbl4_M444
        DW      Tbl4_M445,Tbl4_M446,Tbl4_M447,Tbl4_M448,Tbl4_M449
        DW      Tbl4_M450,Tbl4_M451,Tbl4_M452,Tbl4_M453,Tbl4_M454
        DW      Tbl4_M455,Tbl4_M456,Tbl4_M457,Tbl4_M458,Tbl4_M459
        DW      Tbl4_M460,Tbl4_M461,Tbl4_M462,Tbl4_M463,Tbl4_M464
        DW      Tbl4_M465,Tbl4_M466,Tbl4_M467,Tbl4_M468,Tbl4_M469
        DW      Tbl4_M470,Tbl4_M471,Tbl4_M472,Tbl4_M473,Tbl4_M474
        DW      Tbl4_M475,Tbl4_M476,Tbl4_M477,Tbl4_M478,Tbl4_M479
        DW      Tbl4_M480,Tbl4_M481,Tbl4_M482,Tbl4_M483,Tbl4_M484
        DW      Tbl4_M485,Tbl4_M486,Tbl4_M487,Tbl4_M488,Tbl4_M489
        DW      Tbl4_M490,Tbl4_M491,Tbl4_M492,Tbl4_M493,Tbl4_M494
        DW      Tbl4_M495,Tbl4_M496,Tbl4_M497,Tbl4_M498,Tbl4_M499
        DW      Tbl4_M500,Tbl4_M501,Tbl4_M502,Tbl4_M503,Tbl4_M504
        DW      Tbl4_M505,Tbl4_M506,Tbl4_M507,Tbl4_M508,Tbl4_M509
        DW      Tbl4_M510,Tbl4_M511

;*****************************************************************
;*****************************************************************
;*****************************************************************

; Each motor table has the following format:
; The first line is the delay between motor steps.
; The next group of lines are the motor steps.
; The last line is the terminator command.

; Delay table - a number from 0 - 255. THe entry is multiplied by
; a 2.9 mSec timer. Therefore 1=2.9mSec 2=5.8msec 255=739mSec.
;
; The motor step is entered as a decimal number of 10-190.
; '00' is a PAUSE command base on the motor delay setting.
```

```
; 'FF' or '255' is the end of table command.
;
;
;TABLES WITH ENDING STEP NOT WITHIN REQUIRED RANGE(10-20),(132,136)
;-------------------------------------------------------------------
;M94,M127,M131,M139,M140,M143,M146
;
;WITH DUPLICATE STEPS PUT CONSECUTIVELY
;-------------------------------------
;M187,M193,M219,M220,M229,M237,M241,M242
;M250,M310,M321,M369

Tbl1_M000:
        DB      50              ;motor delay between steps
        DB      10,135
      DB    FFH   ;end


;GEORGE 07/03/98
Tbl1_M001:                              ;dON START  SEQ1 AGE1
        DB      1               ;motor delay between steps
        DB      190,133
        DB      FFH
;
Tbl1_M002:                              ;dON START  SEQ2 AGE1
        DB      1               ;motor delay between steps
        DB      150,145,138,120,145,133,147,133
        DB      FFH     ;end
;
Tbl1_M003:
        DB      10              ;motor delay between steps
        DB      90,100,0,0,0,100,0,0,0,0,133    ;CONNECTED M23 ;dON START
SEQ3 AGE1
;        DB      145,160,0,0,0,160
        DB      FFH     ;end
;
Tbl1_M004:
        DB      1               ;motor delay between steps
        DB      200,190,160,100,133         ;CONNECTED M22 ;dON START
SEQ3 AGE1
        DB      FFH     ;end
;
Tbl1_M005:
        DB      5               ;motor delay between steps
        DB      170,130,90,100,133      ; DONE conected m22 seq4 age1
        DB      FFH     ;end
;
Tbl1_M006:
        DB      10              ;motor delay between steps
        DB      150,200,0,0,150,133     ;seq5 front1 age1
        DB      FFH     ;end
;
Tbl1_M007:
        DB      1               ;motor delay between steps
        DB      120,150,133     ;SEQ6 FRONT1 AGE1 HORSE LAUGH
        DB      FFH  ;end
;
Tbl1_M008:
        DB      10              ;motor delay between steps
        DB      150,200,150,170,133             ;SEQ7 FRONT AGE1
```

```
        DB      FFH     ;end
;
Tbl1_M009:
        DB      10              ;motor delay between steps
        DB      150,200,150,190,170,120,133     ;SEQ8,FRONT AGE1
        DB      FFH     ;end
;
Tbl1_M010:
        DB      1               ;motor delay between steps
        DB      180,100,133                     ;SEQ9,FRONT AGE1
        DB      FFH     ;end
;
Tbl1_M011:
        DB      1               ;motor delay between steps
        DB      80,0,150,0,125,0,0,133          ;SEQ10,FRONT AGE1
        DB      FFH     ;end
;
Tbl1_M012:
        DB      10              ;motor delay between steps
        DB      125,0,0,0,0,0,0,0,133,80,133  ;SEQ11,FRONT AGE1
        DB      FFH     ;end
;
Tbl1_M013:
        DB      20              ;motor delay between steps
        DB      145,133,145,133,145,133,145
        DB      125,0,0,0,0,0,130,0,0,90,133    ;seq12 FRONT AGE1 ADD
SAY20 TO FRONT
        DB      FFH     ;end
;
Tbl1_M014:
        DB      10                              ;motor delay between
steps
        DB      90,130,120,0,0,133              ;seq13 FRONT AGE1 ADD
SAY 22
        DB      FFH     ;end
;
Tbl1_M015:
        DB      10                              ;motor delay between
steps
        DB      125,110,133                     ;seq14 FRONT AGE1 ADD
SAY22
        DB      FFH     ;end
;
Tbl1_M016:
        DB      1               ;motor delay between steps
        DB      160,0,0,133,125,150,133          ;seq15 FRONT AGE1
        DB      FFH     ;end
;
Tbl1_M017:
        DB      10               ;motor delay between steps
        DB      120,133,125,150,120,0,0,0,0,0,0,0,133     ;seq16 FRONT
AGE1 ADD 37
        DB      FFH     ;end
;
Tbl1_M018:
        DB      1               ;motor delay between steps
        DB      124,0,115,0,133,120,133                   ;seq16 FRONT
AGE1 ADD 37
        DB      FFH     ;end
;
```

```
Tbl1_M019:
        DB      10              ;motor delay between steps
;        DB      90,100,0,0,0,100,0,0,0,0,133        ;SEQ1 FRONT AGE2
        DB      175,160,0,0,0,160,0,0,0,0,133
        DB      FFH     ;end
;
Tbl1_M020:
        DB      10              ;motor delay between steps
        DB      143,150,133,155,133             ;SEQ2 fRONT AGE2
        DB      FFH     ;end
;
Tbl1_M021:
        DB      1               ;motor delay between steps
        DB      180,133,180,133
;       DB      100,70,10,133           ;SEQ3AGE2 FRONT ADD SEQ9AGE1
        DB      FFH     ;end
;
Tbl1_M022:
        DB      10              ;motor delay between steps
        DB      140,150,133             ;SEQ4 AGE2 FRONT
        DB      FFH     ;end
;
Tbl1_M023:
        DB      1               ;motor delay between steps
        DB      120,133,0,0,0,0,0,0,0,140,150,133       ;SEQ4 AGE2
        DB      FFH     ;end
;
Tbl1_M024:
        DB      5               ;motor delay between steps
;                               ;SEQ5 AGE2 FRONT
        DB      150,140,138,120,145,133,0,147,133
        DB      FFH     ;end
;
Tbl1_M025:
        DB      1               ;motor delay between steps
        DB      150,200,0,0,150,133,143,133,143
        DB      133,110,133     ;SEQ6 AGE2 FRONT
        DB      FFH     ;end
;
Tbl1_M026:
        DB      10                      ;motor delay between steps
        DB      142,150,133             ;SEQ 7 AGE2 FRONT PART1
        DB      FFH     ;end
;
Tbl1_M027:
        DB      1               ;motor delay between steps
                ;SEQ 7 AGE2 FRONT PART2
        DB      150,145,160,133,145,133,145,133
        DB      FFH     ;end

; danger always followed by 003: dmh
Tbl1_M028:
        DB      1               ;motor delay between steps
        DB      30,70 ;<- OK    ;SEQ8 MIDDLE OF 22,AND 4SOMETHING
        DB      FFH     ;end    ;
;
Tbl1_M029:
        DB      1               ;motor delay between steps
        DB      190,133         ;SEQ 9 TITTER
        DB      FFH     ;end
```

```
;
Tbl1_M030:
        DB      1               ;motor delay between steps
        DB      120,133,140,150,133     ; SEQ10 FRONT AGE2
        DB      FFH     ;end
;
Tbl1_M031:
        DB      5               ;motor delay between steps
        DB      180,160,133,115,105,133                 ;SEQ11 FRONT
AGE 2  ADD 41
        DB      FFH     ;end
;
Tbl1_M032:
        DB      10               ;motor delay between steps
        DB      145,133,145,133,145,133,0,120,115,133
        DB      FFH              ;SEQ12 FRONT AGE 2 ADD 20
;
Tbl1_M033:
        DB      1               ;motor delay between steps
        DB      150,170,190,133,120,133,135,133,150,0,0,133  ;SEQ14
FRONT
        DB      FFH     ;end
;
Tbl1_M034:
        DB      10               ;motor delay between steps
        DB      125,0,0,0,0,0,133,145,133    ;SEQ15 FRONT AGE2 ADD 20
        DB      FFH     ;end
;
Tbl1_M035:
        DB      1               ;motor delay between steps
        DB      120,0,0,0,0,0,0,0,133,145
        DB      133,0,150,133,110,133,120,0,0,133   ;SEQ16 FRONT AGE2
ADD 20
        DB      FFH     ;end
;
Tbl1_M036:
        DB      1               ;motor delay between steps
        DB      155,0,0,0,133                ;SEQ1 FRONT AGE3
        DB      FFH     ;end
;
Tbl1_M037:
        DB      1               ;motor delay between steps
        DB      140,150,133,120,133,110,133     ;SEQ2 FRONT AGE3
        DB      FFH     ;end
;
Tbl1_M038:
        DB      1               ;motor delay between steps
        DB      155,0,0,0,133,155,0,0,0,133     ;SEQ3 FRONT AGE3
        DB      FFH     ;end
;
Tbl1_M039:
        DB      1               ;motor delay between steps
        DB      190,0,0,133                     ;SEQ4 FRONT AGE3
        DB      FFH     ;end
;ERROR
;Tbl1_M040:
;        DB      10               ;motor delay between steps
;        DB      140,150,133                     ;SEQ5 FRONT AGE3 ADD
SEQ14AGE1
;        DB      FFH     ;end
```

```
;
Tbl1_M040:
        DB      10              ;motor delay between steps
        DB      150,200,0,0,150,133,143,133
        DB      143,133,110,0,0,133         ;SEQ6 FRONT AGE3
        DB      FFH     ;end
;
Tbl1_M041:
        DB      1               ;motor delay between steps
        DB      160,140,0,150,133,160,140,133
        DB      150,160,133     ;SEQ7 FRONT AGE3
        DB      FFH     ;end
;
Tbl1_M042:
        DB      1               ;motor delay between steps
        DB      30,70,120         ;SEQ?
;       DB      160,140,0,150,133,160,140,133
        DB      FFH     ;end
;
Tbl1_M043:
        DB      10              ;motor delay between steps
        DB      80,0,150,0,125,0,0,133          ;SEQ10 FRONT AGE3
        DB      FFH     ;end
;
Tbl1_M044:
        DB      1               ;motor delay between steps
        DB      100,133,120,133           ;SEQ11
        DB      FFH     ;end
;
Tbl1_M045:
        DB      10              ;motor delay between steps
        DB      150,0,0,133,120,100,133       ;SEQ12 FRONT AGE3
(HEEY,TICKLE ME) ADD20          DB      4
        DB      FFH     ;end
;
Tbl1_M046:
        DB      10              ;motor delay between steps
        DB      145,133,145,133,145,133         ;SEQ13 fRONT AGE3
(NANNY,NANNY) ADD20
        DB      FFH     ;end
;
Tbl1_M047:
        DB      1               ;motor delay between steps
        DB      125,0,130,0,0,90,133    ;SEQ13 fRONT AGE3 (RASBERRY, HE
HE HE ) ADD20
        DB      FFH     ;end
;
Tbl1_M048:
        DB      1               ;motor delay between steps
        DB      200,0,0,133     ;SEQ16 FRONT AGE3
        DB      FFH     ;end
;
Tbl1_M049:
        DB      1               ;motor delay between steps
        DB      120,110,133,115,133     ;SEQ16
        DB      FFH     ;end
;
Tbl1_M050:
        DB      10              ;motor delay between steps
        DB      140,150,133     ; SEQ2 (TICKLE) FRONT AGE4
```

```
        DB      FFH     ;end
;
Tbl1_M051:
        DB      10              ;motor delay between steps
        DB      125,100,133         ; SEQ2 (AGAIN) FRONT AGE4
        DB      FFH     ;end
;
Tbl1_M052:
        DB      1               ;motor delay between steps
        DB      120,133         ;SEQ3 (YOU) FRONT AGE4
        DB      FFH     ;end
;
Tbl1_M053:
        DB      10              ;motor delay between steps
        DB      160,133        ;SEQ3 (ME) FRONT AGE4
        DB      FFH     ;end
;
Tbl1_M054:
        DB      20              ;motor delay between steps
        DB      150,133        ;SEQ4 (LOVE) FRONT AGE4 ADD45 74 71 20
        DB      FFH     ;end
;
Tbl1_M055:
        DB      10              ;motor delay between steps
        DB      135,133,150 0,0,133        ;SEQ5 (HE HE HE) FRONT AGE4
ADD26
        DB      FFH     ;end
;
Tbl1_M056:
        DB      10              ;motor delay between steps
        DB      154,133,115,0,0,0,0,0,0,133        ;SEQ5 (BIG FUN) FRONT
AGE4 ADD26
        DB      FFH     ;end
;
Tbl1_M057:
        DB      10              ;motor delay between steps
        DB      120,133         ;SEQ8 (NO) FRONT AGE4
        DB      FFH     ;end
;
Tbl1_M058:
        DB      1               ;motor delay between steps
        DB      100,133                ;SEQ8 (PLEASE) FRONT AGE4
        DB      FFH     ;end
;
Tbl1_M059:
        DB      10              ;motor delay between steps
        DB      150,0,0,0,133    ;SEQ9 (HEEY) FRONT AGE4 ADD71
        DB      FFH     ;end
;
Tbl1_M060:
        DB      1              ;motor delay between steps
        DB      120,100,133     ;SEQ14 (PARTY) AGE4 ADD45
        DB      FFH     ;end
;
Tbl1_M061:
        DB      10              ;motor delay between steps
        DB      143,150,170,133        ;SEQ15 (WA WA WA) FRONT AGE4 ADD22
        DB      FFH     ;end
;END GEORGE 07/03/98
;
```

```
;
; (BOTTOM)
;GEORGE 07/04/98
Tbl1_M062:
        DB      20                          ;motor delay between steps
        DB      150,0,0,0,133               ;FORTUNE ASK
      DB    FFH   ;end
;
Tbl1_M063:
        DB      1                           ;motor delay between steps
        DB      150,0,0,133                 ;FORTUNE ASK
      DB    FFH   ;end
;
Tbl1_M064:
        DB      1                           ;motor delay between steps
        DB      150,0,0,0,133               ;FORTUNE TELL (BIG)
      DB    FFH   ;end
;
Tbl1_M065:
        DB      10                          ;motor delay between steps
        DB      190,150,0,0,133             ;FORTUNE TELL (VERY,BIG)
      DB    FFH   ;end
;
Tbl1_M066:
        DB      1                           ;motor delay between steps
        DB      120,0,0,0,0,0,0,0,0,133 ;FORTUNE TELL (SEE)
      DB    FFH   ;end

; danger always followed by 68: dmh
Tbl1_M067:
        DB      10                          ;motor delay between steps
        DB      30,10,30,10,30,10,70        ;<- OK  ;FORTUNE WHINE START
      DB    FFH   ;end
;
Tbl1_M068:
        DB      1                           ;motor delay between steps
        DB      100,133,150,133,150,133 ;FORTUNE WHINE START
      DB    FFH   ;end
;
Tbl1_M069:
        DB      1                           ;motor delay between steps
        DB      150,133                     ;FORTUNE TELL (NO)
      DB    FFH   ;end
;
Tbl1_M070:
        DB      1                           ;motor delay between steps
        DB      125,100,133                 ;FORTUNE TELL (WORRY)
      DB    FFH   ;end
;
Tbl1_M071:
        DB      10                          ;motor delay between steps
        DB      110,120,133                 ;FORTUNE (SOUND)
      DB    FFH   ;end
;
Tbl1_M072:
        DB      1                           ;motor delay between steps
        DB      150,133                     ;FORTUNE (GOOD)
      DB    FFH   ;end
;
```

```
Tbl1_M073:
        DB      1                               ;motor delay between steps
        DB      150,0,133                       ;FORTUNE TELL (VERY)
      DB    FFH   ;end
;
Tbl1_M074:
        DB      1                               ;motor delay between steps
        DB      145,133,150,0,0,0,0,0,133       ;FORTUNE (WHOOPEE)
        DB      FFH     ;end
;
Tbl1_M075:
        DB      1                               ;motor delay between steps
        DB      115,133                         ;FORTUNE (GOOD)
      DB    FFH   ;end
;
Tbl1_M076:
        DB      1                               ;motor delay between steps
        DB      120,0,0,0,0,133                 ;FORTUNE (RASPBERRY)
      DB    FFH   ;end
;
Tbl1_M077:
        DB      1                               ;motor delay between steps
        DB      150,115,133                     ;FORTUNE (OH OH)
      DB    FFH   ;end
;
Tbl1_M078:
        DB      1                               ;motor delay between steps
        DB      150,115,133                     ;FORTUNE (MAY BEE)
        DB      FFH             ;end

;END GEORGE 07/04/98
;START HANGOUT
;GEORGE 07/04/98
;
Tbl1_M079:
        DB      1                               ;motor delay between steps
        DB      150,133,135,150,133             ;SEQ1 HANGING(DE DE DE ,DUM  DUM
DUM DUM) AGE1
      DB    FFH   ;end
;
Tbl1_M080:
        DB      1                               ;motor delay between steps
        DB      190,133                          ;SEQ1 HANGING(DUM  DUM DUM DUM)
AGE1
      DB    FFH   ;end
;
Tbl1_M081:
        DB      1                               ;motor delay between steps
        DB      120,100,133
        DB      120,100,133                     ;SEQ1 HANGING (bEEDO)
      DB    FFH   ;end
;
Tbl1_M082:
        DB      1                               ;motor delay between steps
        DB       143,150,170,0,0,0,0,190    ;-133
;        DB      120,100,160,133                ;SEQ1 HANGING  (YA DA DA )
        DB      FFH     ;end
;
Tbl1_M083:
        DB      1                               ;mot  elay between steps
```

```
        DB      190,120,133
        DB      150,133,150,133          ;SEQ3 HANGING ( OMPAH  bRUMM
BABABUM)
        DB      FFH      ;end
;
Tbl1_M084:
        DB      10                       ;motor delay between steps
        DB      125,120,125,115,133      ;SEQ3 HANGING (bRUMM BABABUM)
      DB    FFH    ;end
;
Tbl1_M085:
        DB      1                        ;motor delay between steps
        DB      115,125,110,125,100,133 ;SEQ4 HANGING (LA LA)
      DB    FFH    ;end
;
Tbl1_M086:
        DB      1                        ;motor delay between steps
        DB      120,130,115              ;SEQ4 HANGING (LA LA)
        DB      100,125,115,125,115,125,115,125,115,133
      DB    FFH    ;end
;
Tbl1_M087:
        DB      1                        ;motor delay between steps
        DB      120,0,0,0,0,0,0,0,133    ;SEQ5 HANGING (HUMM BO DAH WAY-
LOH)
      DB    FFH    ;end
;
;Tbl1_M088:
;        DB      10                       ;motor delay between steps
;        DB      115,133,139,155,160,133 ;SEQ5 HANGING (HUMM BO DAH WAY-
LOH)
;        DB      FFH      ;end

Tbl1_M088:
        DB      10                       ;motor delay between steps
        ;DB      115,133,139,155,160,133 ;SEQ5 HANGING (HUMM BO DAH WAY-
LOH)
        DB      115,133,160,133 ;SEQ5 HANGING (HUMM BO DAH WAY-LOH)
        DB      FFH      ;end
;
Tbl1_M089:
        DB      60                       ;motor delay between steps
        DB      190,170,150,133,0,0,0,0,0,0    ;SEQ6 HANGING (SNORE)
      DB    FFH    ;end
;
Tbl1_M090:
        DB      10                       ;motor delay between steps
        DB      150,133                  ;SEQ6 HANGING (SHOUT)
      DB    FFH    ;end
;
Tbl1_M091:
        DB      1                        ;motor delay between steps
        DB      143,150,140,0,150,0,0,133       ;SEQ6 HANGING (OK KAH)
      DB    FFH    ;end
;
Tbl1_M092:
        DB      5                        ;motor delay between steps
        DB      110,133                  ;SEQ6 HANGING (U-TYE)
      DB    FFH    ;end
;
```

```
Tbl1_M093:
        DB      60                              ;motor delay between steps
        DB      190,180,170,150,133             ;SEQ7 HANGING (SOFTER)
      DB    FFH   ;end
;
; danger sleep
Tbl1_M094:
        DB      50                              ;motor delay between steps
        DB      190,170,150,10                 ;SEQ7 HANGING (SOFTER)
      DB    FFH   ;end
;
Tbl1_M095:
        DB      20                              ;motor delay between steps
        DB      145,133,115,0,133               ;SEQ8 HANGING ADD 76
      DB    FFH   ;end
;
Tbl1_M096:
        DB      1                               ;motor delay between steps
        DB      150,115,150,133                 ;SEQ9 HANGING (DO BE DOBE DO)
      DB    FFH   ;end
;
Tbl1_M097:
        DB      46                              ;motor delay between steps
        DB      170,0,0,0,200,150,0,0,150,0,133     ;SEQ10 HANGING
(YAWN)
      DB    FFH   ;end
;
Tbl1_M098:
        DB      255                             ;motor delay between steps
        DB      150,133                         ;SEQ11 AND SEQ12 HANGING (SIGH)
      DB    FFH   ;end
;
Tbl1_M099:
        DB      1                               ;motor delay between steps
        DB      144,133                         ;SEQ13 SEQ14 HANGING (HA)
      DB    FFH   ;end
;
Tbl1_M100:
        DB      10                              ;motor delay between steps
        DB      104,0,0,0,133
      DB    FFH   ;end
;
Tbl1_M101:
        DB      20                              ;motor delay between steps
        DB      100,133,0,0,0,100,133                   ;SEQ16
      DB    FFH   ;end

  anger, USED IN ONE CASE, HANGING OUT, FOLLOWED BY 101
Tbl1_M102:
        DB      10                              ;motor delay between steps
        DB      0                               ;SEQ16 HANGING (PAUSE) ADD20
      DB    FFH   ;end
;
Tbl1_M103:
        DB      1               ;motor delay between steps
        DB      114,133         ;SEQ6 HANGING (UP)
      DB    FFH   ;end
;
Tbl1_M104:
        DB      1                               ;motor delay between steps
```

```
        DB      120,133                 ;SEQ6 HANGING (ME)
      DB    FFH   ;end
;
Tbl1_M105:
        DB      1                       ;motor delay between steps
        DB      120,133                 ;UP
      DB    FFH   ;end
;
Tbl1_M106:
        DB      10                      ;motor delay between steps
        DB      125,104,133             ;SEQ5 BORING
      DB    FFH   ;end
;    ------------------------------------------------

;END HANGOUT
;
;
; danger, OK PAUSE FOR FORTUNE TELLING
Tbl1_M107:
        DB      1               ;motor delay between steps
        DB      0
        DB       FH     ;end    ;Fortune pause
;END GEORGE 07/04/98
;FEED
;GEORGE 07/05/98
Tbl1_M108:
      DB    10          ;motor delay between steps
        DB      115,0,0,0,0,0,0,0,0,0,0,0,0,0
        DB      0,0,0,0,0,0,0,0,0,0,0,0,0,0
        DB      133 ;SEQ1 FEED AGE1 (UUMMMMM)
      DB    FFH   ;end

Tbl1_M109:
        DB      1             ;motor delay between steps
;        DB      140
        DB      165,0,0,0,0,0,0,150,0,0,165,0,0,0,0,0,0,133     ;SEQ1
FEED AGE1 (AAAAH)
      DB    FFH   ;end
;
Tbl1_M110:
        DB      1               ;motor delay between steps
        DB      120,130,110,133       ;SEQ2 FEED AGE1 (KOH KOH)
      DB    FFH   ;end
;
Tbl1_M111:
        DB      1               ;motor delay between steps
        DB      120,130,120,133  ;ME ME
      DB    FFH   ;end
;
Tbl1_M112:
        DB      1               ;motor delay between steps
        DB      145,133,150,133  ;E-DAY
      DB    FFH   ;end
;
Tbl1_M113:
      DB    1           ;motor delay between steps
        DB      115,130,110,133   ;DO MOH
      DB    FFH   ;end

Tbl1_M114:
```

```
        DB      1               ;motor delay between steps
        DB      115,130,120,133   ;TOH DYE
      DB    FFH   ;end
;
Tbl1_M115:
      DB    10          ;motor delay between steps
        DB      110,133                   ;BURP
      DB    FFH   ;end
;
Tbl1_M116:
        DB      1               ;motor delay between steps
        DB      145,133        ;SIGH
      DB    FFH   ;end
;
Tbl1_M117:
      DB    10          ;motor delay between steps
      DB    150,133
      DB    FFH   ;end
;
Tbl1_M118:
      DB    10          ;motor delay between steps
      DB    120,0,0,0,133
      DB    FFH   ;end
;
Tbl1_M119:
        DB      1               ;motor delay between steps
        DB      120,130,110,133 ;TOH LOO
      DB    FFH   ;end
;
Tbl1_M120:
        DB      1               ;motor delay between steps
        DB      120,133,120,133
      DB    FFH   ;end
;
Tbl1_M121:
        DB      1               ;motor delay between steps
        DB      145,130,120,133   ;HUNGRY
      DB    FFH   ;end
;
Tbl1_M122:
        DB      1                ;motor delay between steps
        DB      150,133          ;LIKE
      DB    FFH   ;end
;
Tbl1_M123:
        DB      1                ;motor delay between steps
        DB      150,0,0,133      ;seq4 feed done
        DB      FFH              ;end
;
;END FEED
;END GEORGE 07/05/98
;
;
;WAKE
;GEORGE 07/06/98
Tbl1_M124:                        ;SG DONE
        DB      255             ;motor delay between steps
        DB      95,133
        DB      FFh
; danger
```

```
Tbl1_M125:                          ;SG DONE
        DB      1                  ;motor delay between steps
        DB      75,90    ;<- OK
        DB      FFh
Tbl1_M126:                          ;SG DONE
        DB                         ;motor delay between steps
        DB      135,120,135
      DB    FFh
Tbl1_M127:                          ;SG DONE
      DB    1             ;motor delay between steps
        DB      80,133
      DB    FFh
; danger
Tbl2_M128:                          ;SG DONE
        DB      1                  ;motor delay between steps
        DB      75,90              ;<-OK
        DB      FFh

Tbl2_M129:                          ;SG DONE
        DB      1                  ;motor delay between steps
        DB      90,110,133
        ;DB      90,110,70
      DB    FFh

Tbl2_M130:                          ;SG DONE
        DB      1                 ;motor delay between steps
        DB      115,133
      DB    FFh
; danger
Tbl2_M131:                          ;SG DONE
        DB      1                 ;motor delay between steps
        DB      90,70
      DB    FFh
Tbl2_M132:                          ;SG DONE
      DB    1             ;motor delay between steps
        DB      95,133
      DB    FFh


Tbl2_M133:                          ;SG DONE
        DB      1                  ;motor delay between steps
        DB      115,133
      DB    FFh

; danger
Tbl2_M134:                          ;SG DONE
        DB      1                 ;motor delay between steps
        DB      185
      DB    FFh
; danger
Tbl2_M135:                          ;SG DONE
        DB      1                ;motor delay between steps
        DB      133
      DB    FFh
; danger
Tbl2_M136:                          ;SG DONE
        DB      1                 ;motor delay between steps
        DB      133
      DB    FFh
; danger
```

```
Tbl2_M137:                          ;SG DONE
        DB      1                   ;motor delay between steps
        DB      145
      DB    FFh
; danger
Tbl2_M138:                          ;SG DONE
        DB      1                   ;motor delay between steps
        DB      120,133,120,133,120,133,120,133,120,133,120,133,70,85
        DB      0,0,70,0,0,0,0,0,0
        DB      FFh
; danger
Tbl2_M139:                          ;SG DONE
        DB      1                   ;motor delay between steps
        DB      82,70
      DB    FFh
; danger
Tbl2_M140:                          ;SG DONE
        DB      1                   ;motor delay between steps
        DB      120,115,130,120,70
        DB      FFH     ;end
;
; danger
Tbl2_M141:                          ;SG DONE
        DB      1                   ;motor delay between steps
        DB      133
      DB    FFH   ;end
; danger
Tbl2_M142:                          ;SG DONE
        DB      1                   ;motor delay between steps
        DB      75
      DB    FFH   ;end
;
Tbl2_M143:                          ;SG DONE
        DB      1                   ;motor delay between steps
       ; DB      90,80,100,75
        DB      90,80,100,133
        DB      FFH     ;end
;
; danger
Tbl2_M144:                          ;SG DONE
        DB      1                   ;motor delay between steps
        DB      120
      DB    FFH   ;end
;
; danger
Tbl2_M145:                          ;SG DONE
        DB      1                   ;motor delay between steps
        DB      110,75
      DB    FFH   ;end
;
Tbl2_M146:                          ;SG DONE
        DB      1                   ;motor delay between steps
        ;DB      90,75

        DB      90,133
        DB      FFH     ;end
;
;danger
Tbl2_M147:                          ;SG DONE
        DB      1                   ;motor delay between steps
```

```
        DB      70,90,75
      DB    FFH   ;end
;
Tbl2_M148:                          ;SG DONE
        DB      1                   ;motor delay between steps
        DB      120,130,115,126,115,140,110,0,0,0,0,0,0,0,0,0,133
      DB    FFH   ;end
;
; danger
Tbl2_M149:                          ;SG DONE
        DB      1                   ;motor delay between steps
        DB      75
      DB    FFH   ;end
;
Tbl2_M150:                          ;SG DONE
        DB      1                   ;motor delay between steps
        DB      146,135
      DB    FFH   ;end
;
Tbl2_M151:                          ;SG DONE
        DB      1                   ;motor delay between steps
        DB      120,133,70,0,135
        DB      FFH     ;end
;
; danger
Tbl2_M152:                          ;SG DONE
        DB      1                   ;motor delay between steps
        DB      75
      DB    FFH   ;end

; danger
Tbl2_M153:                          ;SG DONE
        DB      1                   ;motor delay between steps
        DB      115,75
      DB    FFH   ;end
; danger sleep
Tbl2_M154:                           ;SG DONE
        DB      100                  ;motor delay between steps
        DB
0,0,0,85,30,0,20,0,85,30,0,20,0,85,30,0,20,0,75,0,0,0,0,85
        DB      30,0,20,0,10
      DB    FFH   ;end
; danger
Tbl2_M155:                          ;SG DONE
        DB      1                   ;motor delay between steps
        DB      90,70
      DB    FFH   ;end
; danger
Tbl2_M156:                          ;SG DONE
        DB      1                   ;motor delay between steps
        DB      115,75
      DB    FFH   ;end
;END WAKE
;END GEORGE 07/06/98


;HUNGER
;GEORGE 07/06/98

Tbl2_M157:                           ;SG DONE     ;HUNGER
```

```
        DB      50                  ;motor delay between steps
        ;DB      120,120,133
        DB      120,0,133
        DB      FFH     ;end
;
Tbl2_M158:                          ;SG DONE
        DB      1                   ;motor delay between steps
        DB      180,133
      DB    FFH   ;end
;
Tbl2_M159:                          ;SG DONE
        DB      1                   ;motor delay between steps
        DB      115,110,133
      DB    FFH   ;end
;
Tbl2_M160:                          ;SG DONE
        DB      1                   ;motor delay between steps
        DB      75,133
      DB    FFH   ;end
;
Tbl2_M161:                          ;SG DONE
        DB      1                   ;motor delay between steps
        DB      115,130,115,130
      DB    FFH   ;end
;
Tbl2_M162:                          ;SG DONE
        DB      1                   ;motor delay between steps
        DB      115,110,133
      DB    FFH   ;end
;
Tbl2_M163:                          ;SG DONE
        DB      50                  ;motor delay between steps
        DB      190,133
      DB    FFH   ;end
;
Tbl2_M164:                          ;SG DONE
        DB      50                  ;motor delay between steps
        ;DB      148,148,133

        DB      148,0,133
        DB      FFH     ;end
;
Tbl2_M165:                          ;SG DONE
        DB      50                  ;motor delay between steps
        ;DB      150,150,150,133

        DB      150,0,0,133
        DB      FFH     ;end
;
Tbl2_M166:                          ;SG DONE
        DB      1                   ;motor delay between steps
        DB      120,133
      DB    FFH   ;end
;
Tbl2_M167:                          ;SG DONE
        DB      1                   ;motor delay between steps
        DB      115,133
      DB    FFH   ;end

Tbl2_M168:                          ;SG DONE
```

```
        DB      1               ;motor delay between steps
        DB      115,133
      DB    FFh

;END GEORGE 07/06/98
;END HUNGER

;INVERT
;GEORGE 07/07/98
Tbl2_M169:                      ;SG DONE    ;INVERT
        DB      1               ;motor delay between steps
        DB      110, 122, 75,130,117,133
      DB    FFH   ;end
;
Tbl2_M170:                      ;SG DONE
      DB    10          ;motor delay between steps
        ;DB      165,165,133

        DB      165,0,133
        DB      FFH     ;end
;
Tbl2_M171:                      ;SG DONE
        DB      10              ;motor delay between steps
        DB      105,133
      DB    FFH   ;end
;
Tbl2_M172:                      ;SG DONE
        DB      1               ;motor delay between steps
        DB      150,133
      DB    FFH   ;end
;
Tbl2_M173:                      ;SG DONE
        DB      1               ;motor delay between steps
        DB      155,190,133
      DB    FFH   ;end
;
Tbl2_M174:                      ;SG DONE
        DB      1               ;motor delay between steps
        DB      145,133
      DB    FFH   ;end
;
Tbl2_M175:                      ;SG DONE
        DB      1               ;motor delay between steps
        DB      150,135,145,133
      DB    FFH   ;end
;
Tbl2_M176:                      ;SG DONE
        DB      1               ;motor delay between steps
        DB      75,133
      DB    FFH   ;end
;
Tbl2_M177:                      ;SG DONE
        DB      1               ;motor delay between steps
        DB      110,133,115,133
      DB    FFH   ;end
;
Tbl2_M178:                      ;SG DONE
        DB      1               ;motor delay between steps
        DB      115,133
      DB    FFH   ;end
```

```
;
Tbl2_M179:                          ;SG DONE
        DB      1                   ;motor delay between steps
        DB      115,133
      DB    FFH   ;end
;
Tbl2_M180:                          ;SG DONE
        DB      1                  ;motor delay between steps
        DB      110,125,115,133
      DB    FFH   ;end
;
Tbl2_M181:                          ;SG DONE
        DB      1                   ;motor delay between steps
        DB      150,133
      DB    FFH   ;end
;
Tbl2_M182:                          ;SG DONE
        DB      1                   ;motor delay between steps
        DB      115,133
      DB    FFH   ;end
;
Tbl2_M183:                          ;SG DONE
        DB      1                   ;motor delay between steps
        DB      115,130,110,133
      DB    FFH   ;end
;
Tbl2_M184:                          ;SG DONE
        DB      1                   ;motor delay between steps
        DB      75,133
      DB    FFH   ;end
;
Tbl2_M185:                          ;SG DONE
        DB      1                   ;motor delay between steps
        ;DB      150,150,133

        DB      150,0,133
        DB      FFH     ;end
;
Tbl2_M186:                          ;SG DONE
        DB      1                   ;motor delay between steps
        DB      115,130,115,133
      DB    FFH   ;end
;
Tbl2_M187:                          ;SG DONE
        DB      1                   ;motor delay between steps
        DB      115,130,115,133
      DB    FFH   ;end
;
Tbl2_M188:                          ;SG DONE
        DB      1                   ;motor delay between steps
        DB      145,135,145,133
      DB    FFH   ;end

;
Tbl2_M189:                          ;SG DONE
        DB      1                   ;motor delay between steps
        DB      120,105,133
      DB    FFH   ;end
;
Tbl2_M190:
```

```
        DB      1               ;motor delay between steps
        DB      155,190,133
      DB    FFH   ;end
;
Tbl2_M191:
        DB      1               ;motor delay between steps
        DB      155,190,133
      DB    FFH   ;end

   2_M192:
        DB      1               ;motor delay between steps
        DB      155,190,133
      DB    FFH   ;end
;END GEORGE 07/07/98
;END INVERT

;start at 202
Tbl2_M193:      ;BACKSG                 ;SG DONE
        DB      100               ;motor delay between steps
        ;DB      200,200,200,200,133
        DB      200,0,0,0,133
        DB      FFH     ;end
;
Tbl2_M194:                      ;SG DONE
        DB      1               ;motor delay between steps
        DB      75,133
      DB    FFH   ;end
;
Tbl2_M195:                      ;SG DONE
        DB      1               ;motor delay between steps
        DB      115,125,115,133
      DB    FFH   ;end
;
Tbl2_M196:                      ;SG DONE
      DB    10          ;motor delay between steps
        DB      148,133
      DB    FFH   ;end
;
Tbl2_M197:                      ;SG DONE
        DB      1               ;motor delay between steps
        DB      115,125,115,133
      DB    FFH   ;end
;
Tbl2_M198:                      ;SG DONE
        DB      100              ;motor delay between steps
        DB      145,0,0,133
      DB    FFH   ;end
;
Tbl2_M199:                      ;SG DONE
        DB      10              ;motor delay between steps
        DB      110,133
      DB    FFH   ;end
;
Tbl2_M200:                      ;SG DONE
        DB      1               ;motor delay between steps
        DB      75,133
      DB    FFH   ;end
;
Tbl2_M201:                      ;SG DONE
      DB    10          ;motor delay between steps
```

```
        DB      115,125,115,133
      DB    FFH   ;end
;
Tbl2_M202:                          ;SG DONE
        DB      1                   ;motor delay between steps
        DB      75,133
      DB    FFH   ;end
;
; danger
Tbl2_M203:                          ;SG DONE
        DB      1                   ;motor delay between steps
        DB      120,128,79,133,146,0,0,0,133,145
      DB    FFH   ;end
;
Tbl2_M204:                          ;SG DONE
      DB    10          ;motor delay between steps
        DB      190,0,133
      DB    FFH   ;end
;
Tbl2_M205:                          ;SG DONE
        DB      1                   ;motor delay between step
        DB      115,133
      DB    FFH   ;end
;
; danger
Tbl2_M206:                          ;SG DONE
        DB      1                   ;motor delay between steps
        DB      75
      DB    FFH   ;end
; danger
Tbl2_M207:                          ;SG DONE
      DB    10          ;motor delay between steps
        DB      150
      DB    FFH   ;end
;
Tbl2_M208:                          ;SG DONE
      DB    10          ;motor delay between steps
        DB      75,133
      DB    FFH   ;end
;
Tbl2_M209:                          ;SG DONE
        DB      100                  ;motor delay between steps
        DB      150,0,0,0,133
      DB    FFH   ;end
;
Tbl2_M210:                          ;SG DONE
      DB    10          ;motor delay between steps
        DB      123,110,75,133,115,133
      DB    FFH   ;end
; danger
Tbl2_M211:                          ;SG DONE
        DB      1                   ;motor delay between steps
        DB      75
      DB    FFH   ;end
; danger
Tbl2_M212:                          ;SG DONE
        DB      1                   ;motor delay between steps
        DB      133
      DB    FFH   ;end
;
```

```
Tbl2_M213:                          ;SG DONE
        DB      10                  ;motor delay between steps
        DB      115,150,133
        DB      FFH     ;end
;
Tbl2_M214:                          ;SG DONE
        DB      1                   ;motor delay between steps
        DB      80,133
      DB    FFH   ;end
;
; danger
Tbl2_M215:                          ;SG DONE
        DB      100                  ;motor delay between steps
        DB      138
      DB    FFH   ;end
;
Tbl2_M216:                          ;SG DONE
      DB    10          ;motor delay between steps
        DB      75,133
      DB    FFH   ;end
;
Tbl2_M217:                          ;SG DONE
        DB      1                   ;motor delay between steps
        DB      115,130,115,133
      DB    FFH   ;end
;
Tbl2_M218:                          ;SG DONE
        DB      50                  ;motor delay between steps
        DB      114,133
        DB      FFH     ;end
;
Tbl2_M219:                          ;SG DONE
      DB    10          ;motor delay between steps
;       DB
120,130,120,130,120,130,120,130,120,130,120,130,115,115,133
        DB
120,130,120,130,120,130,120,130,120,130,120,130,115,0,133
        DB      FFH     ;end
;
Tbl2_M220:                          ;SG DONE
      DB    10          ;motor delay between steps
       ;DB
120,130,120,130,120,130,120,130,120,130,120,130,115,115,133
        DB
120,130,120,130,120,130,120,130,120,130,120,130,115,0,133
        DB      FFH     ;end
;
Tbl2_M221:                          ;SG DONE
        DB      10                  ;motor delay between steps
        DB      145,133
      DB    FFH   ;end
;
Tbl2_M222:                          ;SG DONE
        DB      50                  ;motor delay between steps
        DB      0,0,0,0,115,133
      DB    FFH   ;end
;
Tbl2_M223:                          ;SG DONE
        DB      1                   ;motor delay between steps
        DB      115,125,115,133
```

```
        DB      FFH     ;end
;
Tbl2_M224:                          ;SG DONE
        DB      1                   ;motor delay between steps
        DB      75,133
        DB      FFH     ;end
;
Tbl2_M225:                          ;SG DONE
        DB      1                   ;motor delay between steps
        DB      110,133
        DB      FFH     ;end
;
Tbl2_M226:                          ;SG DONE
        DB      100                  ;motor delay between steps
        DB      120,133
        DB      FFH     ;end
;
Tbl2_M227:                          ;SG DONE
        DB      30                  ;motor delay between steps
        DB      190,120,125,120,125,120,125,133
        DB      FFH     ;end
;
Tbl2_M228:                          ;SG DONE
        DB      1                   ;motor delay between steps
        DB      115,130,110,130,115,133
        DB      FFH     ;end
;
Tbl2_M229:                          ;SG DONE
        DB      30                  ;motor delay between steps
        ;DB      115,120,110,110,110,133
        DB      115,120,110,0,0,133
        DB      FFH     ;end
;
Tbl2_M230:                          ;SG DONE
        DB      1                   ;motor delay between steps
        DB      110,125,115,133
        DB      FFH     ;end
;
Tbl2_M231:                          ;SG DONE
        DB      1                   ;motor delay between steps
        DB      75,133
        DB      FFH     ;end
;
Tbl2_M232:                          ;SG DONE
        DB      1                   ;motor delay between steps
        DB      110,133
        DB      FFH     ;end
; danger
Tbl2_M233:                          ;SG DONE
        DB      1                   ;motor delay between steps
        DB      145
        DB      FFH     ;end
;
; danger sleep
Tbl2_M234:                          ;SG DONE
        DB      10                  ;motor delay between steps
        DB      10
        DB      FFH     ;end
;
Tbl2_M235:                          ;SG DONE
```

```
        DB      10              ;motor delay between steps
        DB      115,125,110,133
      DB    FFH   ;end
;
Tbl2_M236:
        DB      10              ;motor delay between steps
        DB      115,133
      DB    FFH   ;end

Tbl2_M237:                      ;SG DONE    ;SICK2
        DB      100             ;motor delay between steps
        ;DB     133,140,140,150,150,180,133
        DB      133,140,0,150,0,180,133
        DB      FFH     ;end
;
Tbl2_M238:                      ;SG DONE
        DB      1               ;motor delay between steps
        DB      120,110,133
      DB    FFH   ;end
;
Tbl2_M239:                      ;SG DONE
        DB      1               ;motor delay between steps
        DB      115,133
      DB    FFH   ;end
;
Tbl2_M240:                      ;SG DONE
        DB      10              ;motor delay between steps
        DB      115,0,0,0,0,133
      DB    FFH   ;end
;
Tbl2_M241:                      ;SG DONE
        DB      1               ;motor delay between steps
        ;DB     124,133,120,133,115,115,0,0,133
        DB      124,133,120,133,115,0,0,0,133
        DB      FFH     ;end
;
Tbl2_M242:                      ;SG DONE
        DB      50              ;motor delay between steps
        ;DB     115,70,120,120,133
        DB      115,70,120,0,133
        DB      FFH     ;end
;
; danger
Tbl2_M243:                      ;SG DONE
        DB      50              ;motor delay between steps
        DB      70
      DB    FFH   ;end
;
Tbl2_M244:                      ;SG DONE
        DB      50              ;motor delay between steps
        DB      120,133
      DB    FFH   ;end
;
Tbl2_M245:                      ;SG DONE
        DB      50              ;motor delay between steps
        DB      75,133
        DB      FFH     ;end
;
Tbl2_M246:                      ;SG DONE
      DB    10          ;motor delay between steps
```

```
        DB      70,133
      DB    FFH   ;end
;
Tbl2_M247:                          ;SG DONE
      DB    10          ;motor delay between steps
        DB      110,133,0,0
      DB    FFH   ;end
;
Tbl2_M248:                          ;SG DONE
        DB      10                  ;motor delay between steps
        DB      145,0,0,0,133
      DB    FFH   ;end
;
Tbl2_M249:                          ;SG DONE
        DB      1                 ;motor delay between steps
        DB      115,0,0,0,133
      DB    FFH   ;end
;
Tbl2_M250:                          ;SG DONE
        DB      10                  ;motor delay between steps
        ;DB      150,150,150,190,0,133

        DB      150,0,0,190,0,133
        DB      FFH     ;end
;GEORGE 07/08/98
;LIGHT
;
Tbl2_M251:
        DB      5             ;motor delay between steps SGTEST
        DB      115,132,125,110,132
        DB      FFh

Tbl2_M252:
        DB      1                 ;motor delay between steps
        DB      190,133
        DB      FFh
Tbl2_M253:
      DB    1           ;motor delay between steps
        DB      10,152,133,160,0,133
      DB    FFh
Tbl2_M254:
        DB      1                 ;motor delay between steps
    ;   DB       143,137,143,137,150,133,155,133
        DB      143,137,143,137,150,0,0,0,133,155,133
      DB    FFh
Tbl2_M255:
        DB      1             ;motor delay between steps
        DB      60,90,60,85,90,60,90,133
      DB    FFh
Tbl3_M256:
        DB      10                ;motor delay between st   DONE RB
        DB      180,165,185,133
      DB    FFh
Tbl3_M257:
      DB    10          ;motor delay between steps
        DB      190,133,105,133,105,160,133     ;WOW          DONE
      DB    FFh
Tbl3_M258:
        DB      4               ;motor delay between steps  DONE
        DB      60,133,0,0,0,0,0,155,133,145,133
```

```
      DB    FFh
Tbl3_M259:
        DB      1               ;motor delay between steps      DONE
        DB      160,133,180,133,147,160,133
      DB    FFh

Tbl3_M260:
        DB      1                ;motor delay between steps
        DB      160,133,90,133
      DB    FFh

Tbl3_M261:
        DB      7               ;motor delay between steps
        DB      190,133,100,133
        DB      FFh
Tbl3_M262:
        DB      7               ;motor delay between steps
        DB      60,133,140,153,0,0,133,150,133
      DB    FFh
Tbl3_M263:
        DB      1               ;MOTOR DELAY BETWEEN STEPS
        DB      155,133,160,133,120,110,133
        DB      FFh
Tbl3_M264:
        DB      10              ;motor delay between stepS
        DB      190,133,0,0,0,0,110,0,0,0,133
      DB    FFh
Tbl3_M265:
        DB      1               ;motor delay between steps
        DB      60,133,180,133
      DB    FFh
;END LIGHT
;END GEORGE 07/08/98
;
;DARK
;GEORGE 07/08/98
Tbl3_M266:
        DB      1               ;motor delay between steps
        DB      150,133,160,133,120,112,0,0,0,0,0,0,0,133
        DB      FFh
Tbl3_M267:
        DB      1               ;motor delay between steps DONE RB
        DB      150,133,120,112,0,0,0,0,0,133,149,0,0,133
      DB    FFh
;
Tbl3_M268:
      DB    10          ;motor delay between steps
        DB      150,133,112,133,120,133,148,133,118,0,0,0,133,146,133
        DB      147,0,0,0,0,0,0,133
        DB      FFH     ;end DONE RB
;
Tbl3_M269:
        DB      1               ;motor delay between steps DONE RB
        DB      10,20,123,115,123,115,123,115,133
        DB      FFH     ;end
;
Tbl3_M270:
        DB      1               ;motor delay between steps   DONE
        DB      190,133,120,133,112,0,0,0,0,0,0,130,112,133
      DB    FFH   ;end
```

```
;
Tbl3_M271:
        DB      1               ;motor delay between steps
        DB      147,155,139,149
        DB      133,149,0,0,0,133       ;SEQ6 AGE4/SEQ14 AGE 4 LIGHT js
      DB    FFH   ;end
;
Tbl3_M272:
        DB      1               ;motor delay between steps
        DB      150,133,0,0,0,159,133,150,0,0,133
        DB      145,137,144,133,117,125,117,133
        DB      FFH     ;end  DONE

Tbl3_M273:
        DB      1               ;motor delay between steps
        DB      145,155,133,120,115,133,190,133
        DB      0,0,0,150,0,0,0,0,0,0,0,0,133
        DB      0,0,0,0,0,0,0,0,0,0,115,133
        DB      FFH     ;end
;
Tbl3_M274:
        DB      1               ;motor delay between steps
        DB      150,133,150,0,0,0,133,0,0,0,0,120,115,0,0,0,0,0,0,133
      DB    FFH   ;end
;
Tbl3_M275:
      DB    10          ;motor delay between steps
        DB
150,133,0,0,0,150,0,0,0,133,0,120,133,120,133,155,0,0,0,0,133
      DB    FFH   ;end
;
Tbl3_M276:
        DB      1               ;motor delay between steps
        DB      190,0,0,0,0,133,0,0,0,0,0,148,133,118,133,0,0,0
        DB      146,133,147,0,0,0,0,0,0,133
      DB    FFH   ;end
;
Tbl3_M277:
        DB      1               ;motor delay between steps
        DB      190,133,120,133,112,0,0,0,0,0,0,130,112,133
      DB    FFH   ;end
;
Tbl3_M278:
        DB      1               ;motor delay between steps
        DB      60,133,60,133,146,154,133
      DB    FFH   ;end
;
Tbl3_M279:
        DB      1              ;motor delay between steps
        DB      190,133,0,0,0,110,0,0,0,0,133
      DB    FFH   ;end
;
Tbl3_M280:
        DB      10              ;motor delay between steps
        DB      150,133,0,0,0,116,0,0,0,133,190,155,0,0,0,133
      DB    FFH   ;end
;
Tbl3_M281:
        DB      1               ;motor delay between steps
        DB      190,155,0,0,0,133,119,0,0,0,0,0,0,133
```

```
        DB      146,133,147,0,0,0,0,0,0,133
      DB    FFH   ;end
;
Tb13_M282:
        DB      1               ;motor delay between steps
        DB      60,133,75,83,78,83,78,133
      DB    FFH   ;end
;
Tb13_M283:
        DB      1               ;motor delay between steps
        DB     145,155,133,120,115,133,72,0,0,0,0,0,92,133,190,133
        DB      FFH     ;end
;
Tb13_M284:
        DB      1               ;motor delay between steps
        DB      190,133,0,0,0,110,0,0,0,0,133
      DB    FFH   ;end
;
Tb13_M285:
      DB    10          ;motor delay between steps
        DB      150,133,0,0,0,116,0,0,0,133,190,155,0,0,0,133
      DB    FFH   ;end
;
Tb13_M286:
        DB      1               ;motor delay between steps
        DB      190,155,0,0,0,133,119,0,0,0,0,0,0,133
        DB      147,0,0,0,0,0,0,0,0,0,133
      DB    FFH   ;end
;
Tb13_M287:
        DB      1               ;motor delay between steps
        DB      190,133,110,0,0,0,0,0,133,112,0,0,0,133
      DB    FFH   ;end
;
Tb13_M288:
        DB      1               ;motor delay between steps
        DB      110,0,0,0,133,115,133,147
        DB      133,190,133   ;SEQ7 AGE4/SEQ15 AGE 4 LIGHT js
      DB    FFH   ;end
;
Tb13_M289:
        DB      1               ;motor delay between steps
        DB      145,155,133,0,0,0,0,120,115,133,150,133
        DB      160,0,0,0,0,190,0,0,0,0,0,0,0,0,0,133
        DB      0,0,0,0,0,0,0,0,0,0,133     ;SEQ8 AGE4/SEQ 16 AGE 4
INVERT js
      DB    FFH   ;end
;END GEORGE 07/08/98
;END DARK
;
;SOUND
Tb13_M290:
        DB      1               ;motor delay between steps
        DB      155,133,0,0,0,0,125
        DB      115,145,155,133         ;S1-A1/S9-A1/S1-A2 SOUND js
        DB      FFH     ;end            ;S9-A2/S1-A3/S9-A3 SOUND js
;
Tb13_M291:
        DB      1               ;motor delay between steps
        DB      100,0,0,0,10
```

```
        DB      0,0,0,0,0,0,0
        DB      0,0,0,70,0,0,0,0        ;S2-A1/S10-A1/S2-A2 SOUND js
        DB      0,0,100,0,0,0,133       ;S10-A2/S2-A3/S10-A3 SOUND js
        DB      FFH     ;end            ;S2-A4 SOUND js
;
Tbl3_M292:
        DB      1               ;motor delay between steps
        DB      110,0,0,133,0,0,0,0
        DB      0,0,155,0,0,0,0
        DB      133,120,0,112,0
        DB      148,0,0,0,0,0,133           ;S3-A1/S11-A1 SOUND js
      DB    FFH   ;end
;
Tbl3_M293:
        DB      15              ;motor delay between steps
        DB      110,0,120,0,0,0,0,0
        DB      145,0,0,0,155,115
        DB      118,0,0,0,0,133         ;S4-A1/S12-A1 SOUND js
      DB    FFH   ;end
;
Tbl3_M294:
        DB      1               ;motor delay between steps
        DB      115,0,0,0,148
        DB      115,0,0,133         ;S5-A1/S13-A1 LIGHT (with say/m2) js
      DB    FFH   ;end
;
Tbl3_M295:
        DB      1               ;motor delay between steps
        DB      155,133,122,0               ;S6-A1/S14-A1/S6-A2 SOUND js
        DB      115,145,120,0,0,133         ;S14-A2/S6-A3/S14-A3 SOUND js
      DB    FFH   ;end
;
Tbl3_M296:
        DB      1               ;motor delay between steps
        DB      14  150
        DB      125,115
        DB      0,0,0,0,133         ;S7-A1/S15-A1 SOUND (with say/m2) js
      DB    FFH   ;end
;
Tbl3_M297:
        DB      1               ;motor delay between steps
        DB      115,0,0,148,0,0,0,0
        DB      138,0,0,0,148,0,0,0
        DB      0,0,0,0,133         ;S8-A1/S16-A1/S8-A3/S16-A3 SOUND js
        DB      FFH     ;end
;
Tbl3_M298:
        DB      1               ;motor delay between steps
        DB      110,0,0,133,0,0,0,0
        DB      0,0,155,0,0,0,0
        DB      133,120,0,112,0
        DB      148,0,0,0,0,0,133       ;S3-A2/S11-A2 SOUND js
      DB    FFH   ;end
;
Tbl3_M299:
        DB      1               ;motor delay between steps
        DB      110,0,120,0,0,0,0,0
        DB      145,0,0,0,155,190
        DB      0,0,0,0,0,0,160,0,133       ;S4-A2/S12-A2 SOUND js
      DB    FFH   ;end
```

```
;
Tbl3_M300:
        DB      1               ;motor delay between steps
        DB      165,0,0,0,190,0,0       ;S5-A2/S13-A2 SOUND (with
say/m2) js
        DB      0,0,165,0,0,0,0,133     ;S5-A3/S13-A3 SOUND (with
say/m2) js
        DB      FFH     ;end            ;S5-A4 SOUND (with say/m2) js
;
Tbl3_M301:
        DB      1               ;motor delay between steps
        DB      115,0,0,0,0,145,0,0,165         ;S7-A2/S15-A2 SOUND (with
say/m2) js
        DB      0,0,190,165,0,0,0,133
      DB    FFH   ;end
;
Tbl3_M302:
        DB      1               ;motor delay between steps
        DB      115,0,0,148,0,0,0
        DB      0,0,0,0,133         ;S8-A2/S16-A2 SOUND js
      DB    FFH   ;end
;
Tbl3_M303:
        DB      1               ;motor delay between steps
        DB      110,0,0,133,0,0,0,0
        DB      0,0,155,0,0
        DB      133,0,112,0
        DB      148,0,0,0,0,0,133       ;S3-A3/S11-A3 SOUND js
      DB    FFH   ;end
;
Tbl3_M304:
        DB      1               ;motor delay between steps
        DB      110,0,120,0,0,0,0,0
        DB      160,0,0,0,190
        DB      160,0,0,0,0,133     ;S4-A3/S12-A3 SOUND js
        DB      FFH     ;end        ;S4-A4 SOUND js
;
Tbl3_M305:
        DB      1               ;motor delay between steps
        DB      115,0,0,0,0,160
        DB      0,0,190,0,0,0,0
        DB      0,165,133       ;S7-A3/S15/A3 SOUND (with say/m2) js
        DB      FFH     ;end    ;S7-A4 SOUND (with say/m2) js
;
Tbl3_M306:
        DB      1               ;motor delay between steps
        DB      157,0,0,0,133
        DB      0,0,120,0,0,0
        DB      133,150,0,0,0,0,133     ;S1-A4 SOUND js
      DB    FFH   ;end
;
Tbl3_M307:
        DB      1               ;motor delay between steps
        DB      110,0,0,133,0,0,0,0
        DB      0,0,155,0,0
        DB      133,0,112,0,0,0
        DB      148,0,0,0,0,0,0,0,0,133       ;S3-A4 SOUND js
      DB    FFH   ;end
;
Tbl3_M308:
```

```
        DB      1               ;motor delay between steps
        DB      157,0,0,0,133
        DB      0,0,120,0,0,0
        DB      133,150,0,0,0,0,0,0,133          ;S6-A4 SOUND js
      DB    FFH   ;end
;
Tbl3_M309:
        DB      1               ;motor delay between steps
        DB      115,0,0,148,0,0,0,0,0,0,0
        DB      138,0,0,0,0,0,148,0,0,0
        DB      0,0,0,0,133       ;S8-A4 SOUND js
      DB    FFH   ;end
;END GEORGE
;END SOUND
;GEORGE 07/09/98
;TILT
Tbl3_M310:
        DB      1             ;motor delay between steps
        ;DB      170,170,0,0,0
        DB      170,0,0,0,0
        DB      0,0,0,0,133        ;S1 A1/S4 A1/S2 A4 TILT js
        DB      FFh
Tbl3_M311:
        DB      1               ;motor delay between steps
        DB      125,0,0,0,133,120,145,110,133   ;S2 A1 TILT js
        DB      FFH

Tbl3_M312:
        DB      1               ;motor delay between steps
        DB      150,133,145,133,120,133     ;S3 A1 TILT js
      DB    FFh
Tbl3_M313:
      DB    1           ;motor delay between steps
        DB      100,0,0,0,0
        DB      0,0,0,0,133     ;S5 A1/S4 A2/S2 A3/S2 A4 TILT js
        DB      FFh
Tbl3_M314:
        DB      1               ;motor delay between steps
        DB      120,100,0,0,0,0,0,0,70,80,90
        DB      70,85,100,0,0,133     ;S6 A1 TILT js
      DB    FFh
Tbl3_M315:
        DB      1             ;motor delay between steps
        DB      125,133,100,133,145,0,0,160
        DB      190,0,0,175,160,133     ;S7 A1 TILT/S6 A2 TILT js
      DB    FFh
Tbl3_M316:
        DB      1               ;motor delay between steps
        DB      145,133,145,160,145,160
        DB      0,0,0,0,0,0,190,0,0,0,0,0
        DB      0,0,0,0,0,0,0,150,133         ;S8 A1 TILT (with say/m5)
js
      DB    FFh
Tbl3_M317:
      DB    10          ;motor delay between steps
        DB      160,0,0,0,0,0,0,0,190,133         ;S9 A1 TILT/S9 A2 TILT
js
      DB    FFh
Tbl3_M318:
      DB    10          ;motor delay between steps
```

```
        DB      145,165,0,0,0,0,0,0,0,0,0,0
        DB      190,0,0,180,190,133     ;S10 A1 TILT/S10 A2 TILT js
      DB    FFh
Tbl3_M319:
      DB    1           ;motor delay between steps
        DB      0,120,0,0,133,141
        DB      133,120,0,0,0,133       ;S11 A1 TILT (with say/m2) js
        DB      FFh

Tbl3_M320:
        DB      1               ;motor delay between steps
        DB      150,133,123,0,0,133,142
        DB      0,0,150,0,0,0,0,0,133         ;S12 A1 TILT js
      DB    FFh

Tbl3_M321:
        DB      1               ;motor delay between steps
        ;DB      200,170,170,0,0,0,0,133    ;S13 A1 / S15 A3 TILT js

        DB      200,170,0,0,0,0,0,133    ;S13 A1 / S15 A3 TILT js
        DB      FFh

Tbl3_M322:
        DB      1               ;motor delay between steps
        DB      170,0,0,0,0,133,126,130,118,133         ;S15 A1 TILT js
      DB    FFh
Tbl3_M323:
        DB      1             ;motor delay between steps
        DB      155,0,0,0,0,185
        DB      160,0,0,133     ;S16 A1 / S15 A2 / S13 A3 TILT js
      DB    FFh
Tbl3_M324:
        DB      1               ;motor delay between steps
        DB      170,160,0,0,0,0,0,133    ;S1 A2/S3 A2/S1 A3/S1 A4 TILT
js
      DB    FFh
Tbl3_M325:
      DB    10          ;motor delay between steps
        DB      120,145,110,133         ;S2 A2 TILT (with say/m16) js
      DB    FFh
Tbl3_M326:
      DB    10          ;motor delay between steps
        DB      120,100,0,0,0,0,0,0,133
        DB      148,133,142,115,0,0,133             ;S5 A2 TILT js
        DB      FFh
Tbl3_M327:
        DB      1               ;motor delay between steps
        DB      145,133,145,160,145,160,0,0,0,0,0,0
        DB      190,0,0,0,0,0,0,0,0
        DB      150,133                 ;S7 A2 TILT (with say/m5) js
      DB    FFh
;
Tbl3_M328:
        DB      1               ;motor delay between steps
        DB      145,0,0,160,0,0,0,0
        DB      0,0,0,0,0,0,133         ;S8 A2 TILT (with say/m5) js
      DB    FFH   ;end
;
Tbl3_M329:
        DB      1               ;motor delay between steps
```

```
        DB      0,120,133,143
        DB      118,0,0,0,133           ;S11 A2 TILT (with say/m2) js
      DB    FFH   ;end
;
Tbl3_M330:
        DB      1               ;motor delay between steps
        DB      150,133,123,0,0,133,142
        DB      0,0,150,0,0,0,0,0,133   ;S12 A2 TILT js
      DB    FFH   ;end
;
Tbl3_M331:
        DB      1               ;motor delay between steps
        DB      120,150,133             ;S13 A2 TILT (with say/m5) js
      DB    FFH   ;end
;
Tbl3_M332:
        DB      1               ;motor delay between steps
        DB      120,0,0,0,0,150,0,0,0
        DB      160,0,0,0,133,110,0,0,133       ;S14 A2 TILT js
      DB    FFH   ;end
;
Tbl3_M333:
      DB    10          ;motor delay between steps
        DB      155,0,0,0,0,190,0,0,183,0,0,0
        DB      175,0,0,0,162,0,0,0,0,0,0,0,133
        DB      0,0,120,115,110,115,105,133
        DB      145,155,165,0,0,0,0
        DB      0,0,0,0,0,133       ;S16 A2/S14 A3/S14 A4 TILT js
      DB    FFH   ;end
;
Tbl3_M334:
      DB    10          ;motor delay between steps
        DB      120,100,0,0,0,0,0,0,133       ;S3 A3 TILT js
      DB    FFH   ;end
;
Tbl3_M335:
        DB      1               ;motor delay between steps
        DB      145,133,120,117
        DB      110,0,0,133     ;S4 A3/S4 A4 TILT (with say/m26) js
      DB    FFH   ;end
;
Tbl3_M336:
        DB      1               ;motor delay between steps
        DB      145,165,0,0,0,0,0,0,0,0,0,0,133
        DB      120,133,145,155,0,0,0,133,115,0,0,0,133 ;S4 A3 TILT js
        DB      FFH     ;end

Tbl3_M337:
        DB      1               ;motor delay between steps
        DB      145,133,122,147,139,160
        DB      190,0,0,0,0
        DB      0,0,0,0,155,133         ;S6 A3 TILT (with say/m5) js
      DB    FFH   ;end
;
Tbl3_M338:
        DB      1               ;motor delay between steps
        DB      145,165,0,0,0,0,0,0,0
        DB      0,0,0,0,0,0,133   ;S7 A3/S7 A4 TILT (with say/m5) js
      DB    FFH   ;end
;
```

```
Tbl3_M339:
        DB      1               ;motor delay between steps
        DB      145, 165,0,0,0,0,0,0,0
        DB      0,0,0,0,0,0,190,133,155,133     ;S8 A3/S8 A4 TILT js
      DB    FFH   ;end
;
Tbl3_M340:
        DB      1               ;motor delay between steps
        DB      0,0,0,110,0,0
        DB      115,0,0,0,0,0,0,133     ;S9 A3/S9 A4 TILT (with say/m9)
js
      DB    FFH   ;end
;
Tbl3_M341:
        DB      10               ;motor delay between steps
        DB      165,0,0,0,0,0,0,0,0
        DB      0,0,190,180,190,133   ;S10 A3/S10 A4 TILT (with
say/m16)js
      DB    FFH   ;end
;
Tbl3_M342:
        DB      1               ;motor delay between steps
        DB      143,118,0,0,0,0,0,133   ;S11 A3/S15 A4 TILT (with
say/m2&34)js
        DB      FFH     ;end
;
Tbl3_M343:
        DB      1               ;motor delay between steps
        DB      145,150,145,160,133         ;S12 A3 TILT (with say/m5)
js
      DB    FFH   ;end
;
Tbl3_M344:
        DB      10               ;motor delay between steps
        DB      148,155,0,0,0,0,138,148,155
        DB      0,0,0,0,133,125,120,115,133    ;S16 A3 TILT (with
say/m5)js
      DB    FFH   ;end
;
Tbl3_M345:
        DB      1               ;motor delay between steps
        DB      155,0,0,120,0,0,0,0,133      ;S3 A4 TILT (with say/m26)
js
      DB    FFH   ;end
;
Tbl3_M346:
        DB      1               ;motor delay between steps
        DB      145,165,0,0,0,0,0,0,0,0,0,0,133
        DB      120,133,145,125,0,0,0
        DB      133,115,0,0,0,133       ;S5 A4 TILT js
      DB    FFH   ;end
;
Tbl3_M347:
      DB    10          ;motor delay between steps
        DB      115,133,120,160
        DB      0,0,0,0,0,190,0,0,0,0
        DB      0,0,0,0,0,0,0,0,155,133    ;S6 A4 TILT (with say/m5) js
      DB    FFH   ;end
;
Tbl3_M348:
```

```
        DB      1               ;motor delay between steps
        DB      120,133,115,133,155
        DB      0,0,0,0,0,0,133         ;S11 A4 TILT (with say/m2) js
      DB    FFH   ;end
;
Tbl3_M349:
        DB      1               ;motor delay between steps
        DB      145,155,115,133          ;S13 A4 TILT (with say/m5) js
      DB    FFH   ;end
;
Tbl3_M350:
        DB      5               ;motor delay between steps
        DB      145,158,0,0,0,0,138,147,155
        DB      0,0,0,0,0,0,133
        DB      125,120,115,133     ;S16 A4 TILT (with say/m5) js
      DB    FFH   ;end
;
;END TILT
;END GEORGE
;GEORGE
;IR 07/09/98
Tbl3_M351:
        DB      20              ;motor delay between steps SGTEST
        DB      120,100,133     ;seq1,seq2,seq3,seq4 IR age 1
        DB      FFh
Tbl3_M352:
        DB      46               ;motor delay between steps SGTEST
        DB      115,100,75,133     ;seq5  ir age 1
        DB      FFh

; DANGER
Tbl3_M353:
        DB      30               ;motor delay between steps
        DB      115,130,100,70  ;SEQ6 (DANCE,WAH) ir AGE1
      DB    FFh
;
Tbl3_M354:
        DB      1               ;motor delay between steps
        DB      133,145,155,190,133,155,175,145,133  ;SEQ6 (DO DO DO) ir
AGE1
        DB      FFh
Tbl3_M355:
        DB      8               ;motor delay between steps
        DB      145,115,145,133,145,115,145,133,0,0,0,0,0
        DB      125,110,133,0,160,0,0,0,133
      DB    FFH   ;end
Tbl3_M356:
        DB      1              ;motor delay between steps
        DB      0
        DB      FFh                ;empty space
Tbl3_M357:
        DB      1               ;motor delay between steps
        DB      120,115,110,105,100,80,100,120,115,100,45,133     ;seq8
ir age1
      DB    FFh
Tbl3_M358:
      DB    10          ;motor delay between steps
        DB      120,115,100,80,133,145,160,133       ;seq9 ir age1
      DB    FFh
Tbl3_M359:
```

```
        DB      1               ;motor delay between steps
        DB      115,133,140,145,133,160,180
        DB      173,167,160,180,173,167,160,140,145,133     ;seq10,11 ir
age1
      DB    FFh
Tbl3_M360:
      DB    1           ;motor delay between steps
        DB      120,107,122,113,100,75,90,80,88,100,0,0,133
        DB      120,107,122,113,100,75,90,80,88,100,0,133
        DB      146,140,155,133                                 ;seq12 ir
age1
        DB      FFh

Tbl3_M361:
        DB      5               ;motor delay between steps
        DB
115,125,100,10,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
        DB      0,0,0,0,0,0,0,0,0,0,75,80,85,90,95,100,115
        DB      110,118,100,0,133                               ;seq13,14 ir
age1
        DB      FFh

Tbl3_M362:
        DB      10
        DB      160,0,0,190,160,0,0,133,100,0,0,0,133           ;seq15 ir
age1
        DB      FFh

; DANGER SLEEP
Tbl3_M363:
        DB      90        ;10               ;motor delay between steps
        DB      85,40,30,85,40,30,0,85,40,30,0,85,40,30,10  ;seq16 ir
age1
      DB    FFh
Tbl3_M364:
        DB      1             ;motor delay between steps
        DB      125,113,125,118,105,133   ;seq1,2,3 ir age2
      DB    FFh
Tbl3_M365:
      DB    10          ;motor delay between steps
        DB      125,113,125,118,105,133  ;SEQ4,5 IR AGE2
        DB      FFh


Tbl3_M366:
      DB    10          ;motor delay between steps
        DB      145,155,140,145,142,150,0,0,0,0,0,0,0,133         ;seq6
ir age2
      DB    FFh
Tbl3_M367:
        DB      5               ;motor delay between steps
        DB      10,40,10,40,133,143,140,145,143,145
        ;DB      133
        DB      100,133
        DB      125,113,133                         ;seq7,8 ir age 2
        DB      125,113,133                         ;seq7,8 ir age 2
        DB      FFh

Tbl3_M368:
      DB    10          ;motor delay between steps
```

```
        DB      125,115,105,0,0,133,145,143,155,133,100,133      ;seq9
ir age2
      DB    FFh
;
Tbl3_M369:
        DB      1               ;motor delay between steps
        DB      125,120,115,113,110,105,123,108
        ;DB      123,115,110,100,100,100,100,0,0,0,0,0,0,0,0,133
;seq10 ir age2

        DB      123,115,110,100,0,0,0,0,0,0,0,0,0,0,0,133   ;seq10 ir
age2
        DB      FFH     ;end
;
Tbl3_M370:
        DB      1               ;motor delay between steps
        DB      125,119,113,120,113,140,150,133                  ;seq11
ir age2
      DB    FFH   ;end
;
Tbl3_M371:
        DB      1               ;motor delay between steps
        DB      150,0,0,0,100,0,0,10,0,0,0,0,0,0,0,0,0,0,0,0,0
        DB      0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
        DB      115,90,110,100,133                  ;seq13,14 ir age2
      DB    FFH   ;end
;
Tbl3_M372:
        DB      43               ;motor delay between steps
        DB      100,0,0,150,0,0,100,0,0,0,0,133          ;seq15 ir age2
        DB      FFH     ;end
; DANGER SLEEP
Tbl3_M373:
        DB      90               ;motor delay between steps
        DB      85,40,30,85,40,30,85,40,30,10       ;seq16 ir age2
      DB    FFH   ;end
;
Tbl3_M374:
        DB      1               ;motor delay between steps
        DB      115,145,140,160,133         ;seq1,2,3,4,5 ir age3
      DB    FFH   ;end
;
Tbl3_M375:
        DB      1              ;motor delay between steps
        DB      120,0,0,145,138,150,120,105,133      ;seq6 ir age3
      DB    FFH   ;end
;
Tbl3_M376:
        DB      1               ;motor delay between steps
        DB      115,0,145,155,0,0,136,150,145,190,151,133,150
        DB      145,190,151,0,133     ;seq7.8 ir age3
      DB    FFH   ;end
;
Tbl3_M377:
        DB      1               ;motor delay between steps
        DB      120,123,112,133,143,151,160,133   ;seq9 ir age3
      DB    FFH   ;end
;
Tbl3_M378:
        DB      1               ;motor delay between steps
```

```
        DB      120,122,115,125,112,150,0,0,0,133          ;seq11 ir age3
      DB    FFH   ;end
;
Tbl3_M379:
        DB      1               ;motor delay between steps
        DB      115,10,0,0,10,0,0,0,0,0,0,0,0,0,0,0,0
        DB      0,0,0,0,0,0,0,0,0
        DB      145,110,0,0,0,0,0,0,133                ;seq13,14 ir age3
      DB    FFH   ;end
;
Tbl3_M380:
        DB      12              ;motor delay between steps
        DB      117,0,0,0,0,0,0,133,0,0,0,0,0,0,0,100,0,0,0,0,30
        DB      100,0,0,0,0,0,0,0,0,0,0,0,0,0,0,133          ;seq15 ir age3
      DB    FFH   ;end
;
Tbl3_M381:
        DB      5               ;motor delay between steps
        DB      120,150,110,0,0,0,133                  ;seq1,2,3,4,5 ir
age4
      DB    FFH   ;end
;
Tbl3_M382:
      DB    10          ;motor delay between steps
        DB      120,110,145,155,100,133                ;seq6 ir age4
      DB    FFH   ;end
;
Tbl3_M383:
        DB      8               ;motor delay between steps
        DB      145,115,145,133,145,115,145,133,0,0,0,0,0
        DB      125,110,133,0,160,0,0,0,133
      DB    FFH   ;end
;
Tbl4_M384:
        DB      1               ;motor delay between steps
        DB      115,133,143,148,136,160,180
        DB      173,167,160,180,173,167,160,140,145,133     ;seq9 ir
age4
      DB    FFH   ;end
;
Tbl4_M385:
        DB      1               ;motor delay between steps
        DB      118,0,0,155,0,0,133,0,0,118,0,133,0,0,0,0,110
        DB      0,0,0,133,120,107,122,113,100,75,90,80,88,100,133
        DB      FFH     ;end   SAY NUMBERS MODIFIED TO MATCH CORRECT
DIALOGUE
;
Tbl4_M386:
        DB      1               ;motor delay between steps
        DB      120,123,112,133,143,151,160,133
      DB    FFH   ;end
;
Tbl4_M387:
        DB      1               ;motor delay between steps
        DB    120,0,0,145,110,145,110,0,0,0,0,0,133
      DB    FFH   ;end
;
Tbl4_M388:
        DB      1               ;motor delay between steps
        DB      120,110,133 ;OK                  ;seq14 ir age4
```

```
      DB    FFH   ;end
;
Tbl4_M389:
        DB      90              ;motor delay between steps
        DB      150,0,130,0,100,0,133          ;YAWN
      DB    FFH   ;end
; DANGER SLEEP
Tbl4_M390:
         DB      90              ;motor delay between steps
         DB      0,0,0,85,30,0,20,0,85,30,0,20,0,85,30,0,20,0,85,10
         DB      FFH     ;end
;END GEORGE 07/09/98
;END IR

; FURBY SAYS: (LIGHT) DMH
Tbl4_M391:
        DB      10                      ;motor delay between steps
        DB      110,133                 ;LIGHT (furby says)
;       DB      110,120,133             ;LIGHT (furby says)
      DB    FFH   ;end

;
Tbl4_M392:                              ; dmh no light
        DB      1                       ;motor delay between steps
        DB      150,0,0,0,115,0,0,0,0,133
      DB    FFH   ;end
;
Tbl4_M393:                              ; dmh loud sound
        DB      30                      ;motor delay between steps
        DB      150,0,0,0,115,0,0,0,0,133
      DB    FFH   ;end
;
Tbl4_M394:                              ; LISTEN DMH

        DB      10                      ;motor delay between steps
        DB      140,150,0,0,133
        DB      FFH
;
Tbl4_M395:
        DB      10              ;motor delay between steps
        DB      160,133         ;(ME)
        DB      FFH     ;end
;
Tbl4_M396:
        DB      1               ;motor delay between steps
        DB      120,130,120,133 ;ME ME
      DB    FFH   ;end
;
;
Tbl4_M397:
      DB    1           ;motor delay between steps
        DB      115,130,110,133   ;DO MOH
      DB    FFH   ;end
;
Tbl4_M398:
        DB      1               ;motor delay between steps
        DB      120,130,110,133 ;TOH LOO
      DB    FFH   ;end
;
;
```

```
Tbl4_M399:
        DB      1               ;motor delay between steps
        DB      FFH   ;end
;
;
Tbl4_M400:
        DB      1               ;motor delay between steps
        DB      FFH   ;end      ; start diagnostic
;
Tbl4_M401:
        DB      1               ;motor delay between steps
        DB      FFH   ;end      ; key press beep
;
Tbl4_M402:
        DB      1               ;motor delay between steps
        DB      FFH   ;end      ; pass beep
;
Tbl4_M403:
        DB      1               ;motor delay between steps
        DB      FFH   ;end      ; fail beep
;
Tbl4_M404:
        DB      1               ;motor delay between steps
        DB      FFH   ;end
;
Tbl4_M405:
        DB      1               ;motor delay between steps
        DB      10,200,10,134   ; motor cal
        DB      FFH   ;end
;
Tbl4_M406:
        DB      1               ;motor delay between steps
        DB      120               ; feed 1
        DB      FFH   ;end
;
Tbl4_M407:
        DB      255             ;motor delay between steps
        DB      0,134           ; feed 2
        DB  FFH     ;end
;
Tbl4_M408:
        DB      1               ;motor delay between steps
        DB      30              ; light pass
        DB      FFH   ;end
;
Tbl4_M409:
        DB      1               ;motor delay between steps
        DB      160             ; sound pass
        DB      FFH   ;end
;
Tbl4_M410:
        DB      1               ;motor delay between steps
        DB      10              ; sleep
        DB      FFH   ;end
;
Tbl4_M411:                      ; PEEK-BOO (HIDE AND SEEK) DHM
        DB      20              ;MOTOR DELAY BETWEEN STEPS
        DB      155,133,0,0,147,133
        DB      FFh
;
```

```
Tbl4_M412:                              ; feed dmh
      DB    1               ;motor delay between steps
        DB      165,0,0,0,0,0,0,150,0,0,165,0,0,0,0,0,0,150 ;(AAAAH)
        DB      0,0,165,0,0,0,0,0,0,133 ;(AAAAH)
      DB    FFH   ;end
;


;     DB    FFH   ;end
;
Tbl4_M413:
      DB    1               ;motor delay between steps
      DB    FFH   ;end
;
Tbl4_M414:
      DB    1               ;motor delay between steps
      DB    FFH   ;end
;
Tbl4_M415:
      DB    1               ;motor delay between steps
      DB    FFH   ;end
;
Tbl4_M416:
      DB    1               ;motor delay between steps
      DB    FFH   ;end
;
Tbl4_M417:
        DB      1                       ;motor delay between steps
      DB    FFH   ;end
;
Tbl4_M418:
      DB    1               ;motor delay between steps
      DB    FFH   ;end
;
Tbl4_M419:
      DB    1               ;motor delay between steps
      DB    FFH   ;end
;
Tbl4_M420:
      DB    1               ;motor delay between steps
      DB    FFH   ;end
;
Tbl4_M421:
      DB    1               ;motor delay between steps
      DB    FFH   ;end
;
Tbl4_M422:
      DB    1               ;motor delay between steps
      DB    FFH   ;end
;
Tbl4_M423:
      DB    1               ;motor delay between steps
      DB    FFH   ;end
;
Tbl4_M424:
      DB    1               ;motor delay between steps
      DB    FFH   ;end
;
Tbl4_M425:
      DB    1               ;motor delay between steps
```

```
        DB      FFH     ;end
;
Tbl4_M426:
        DB      1               ;motor delay between steps
        DB      FFH     ;end
;
Tbl4_M427:
        DB      1               ;motor delay between steps
        DB      FFH     ;end
;
Tbl4_M428:
        DB      1               ;motor delay between steps
        DB      FFH     ;end
;
Tbl4_M429:
        DB      1               ;motor delay between steps
        DB      FFH     ;end
;
Tbl4_M430:
        DB      1               ;motor delay between steps
        DB      FFH     ;end
;
Tbl4_M431:
;
Tbl4_M432:
;
Tbl4_M433:
;
Tbl4_M434:
        DB      1               ;motor delay between steps
        DB      0
        DB      FFH     ;end
;
Tbl4_M435:
        DB      1               ;motor delay between steps
        DB      0
        DB      FFH     ;end
;
Tbl4_M436:
        DB      1               ;motor delay between steps
        DB      0
        DB      FFH     ;end
;
Tbl4_M437:
        DB      1               ;motor delay between steps
        DB      0
        DB      FFH     ;end
;
Tbl4_M438:
        DB      1               ;motor delay between steps
        DB      0
        DB      FFH     ;end
;
Tbl4_M439:
        DB      1               ;motor delay between steps
        DB      0
        DB      FFH     ;end
;
Tbl4_M440:
        DB      1               ;motor delay between steps
```

```
        DB      0
        DB      FFH     ;end
;
Tbl4_M441:
;
Tbl4_M442:
;
Tbl4_M443:
;
Tbl4_M444:
;
Tbl4_M445:
;
Tbl4_M446:
;
Tbl4_M447:
;
Tbl4_M448:
;
Tbl4_M449:
;
Tbl4_M450:
;
Tbl4_M451:
;
Tbl4_M452:
;
Tbl4_M453:
;
Tbl4_M454:
;
Tbl4_M455:
;
Tbl4_M456:
;
Tbl4_M457:
;
Tbl4_M458:
;
Tbl4_M459:
;
Tbl4_M460:
;
Tbl4_M461:
;
Tbl4_M462:
;
Tbl4_M463:
;
Tbl4_M464:
;
Tbl4_M465:
;
Tbl4_M466:
;
Tbl4_M467:
;
Tbl4_M468:
;
Tbl4_M469:
```

```
;
Tbl4_M470:
;
Tbl4_M471:
;
Tbl4_M472:
;
Tbl4_M473:
;
Tbl4_M474:
;
Tbl4_M475:
;
Tbl4_M476:
;
Tbl4_M477:
;
Tbl4_M478:
;
Tbl4_M479:
;
Tbl4_M480:
;
Tbl4_M481:
;
Tbl4_M482:
;
Tbl4_M483:
;
Tbl4_M484:
;
Tbl4_M485:
;
Tbl4_M486:
;
Tbl4_M487:
;
Tbl4_M488:
;
Tbl4_M489:
;
Tbl4_M490:
;
Tbl4_M491:
;
Tbl4_M492:
;
Tbl4_M493:
;
Tbl4_M494:
;
Tbl4_M495:
;
Tbl4_M496:
;
Tbl4_M497:
;
Tbl4_M498:
;
Tbl4_M499:
```

```
;
Tbl4_M500:
;
Tbl4_M501:
;
Tbl4_M502:
;
Tbl4_M503:
;
Tbl4_M504:
;
Tbl4_M505:
;
Tbl4_M506:
;
Tbl4_M507:
;
Tbl4_M508:
;
Tbl4_M509:
;
Tbl4_M510:
        DB      10                      ;motor delay between steps
        DB      10,200,134              ;
        DB      FFH                     ;end
;
Tbl4_M511:
        DB      10                      ;motor delay between steps
        DB      10,200,10               ;
        DB      FFH                     ;end

.

.
```