

f008.txt

```
;*****
; Furby(TM)
; Version 008: Finally Version, extra movements have been added
; File Name: F008.ASM
; Author: Juanita Heidebrecht, 9308771
; Date: November 19, 2000
; Class: COMP630, Computer Engineering Technology
; School: Niagara College of Applied Arts & Technology
;
;*****
; Target: PIC16F84 MCU Assembler: MPASM 2.15
;
; Hardware:
; Port B
; RB0 MAX232 Pin ? / TxD1 (Output)
; RB1 MAX232 Pin ? / RxD1 (Input)
; RB2 Motor Forward (Output)
; RB3 Motor Reverse (Output)
; RB4 CAM (Input)
; RB5 Gear_LED_ON (Output)
; RB6 Gear_Rotation (Input)
; RB7 N/A
; Port A
; RA0 74HC165 Pin 1 PL / Shift Register enable (Output)
; RA1 74HC165 Pin 2 CP1 / Clock #1 (Output)
; RA2 N/A
; RA3 74HC165 Pin 9 / Serial Output From Last State (Input)
; RA4 N/A
; RA5 N/A
; RA6 N/A
; RA7 N/A
; * 74HC165 Pin 15 is now hooked up to ground directly instead of
; using the pic.
; * Extra Hardware layout
; 74HC165 Pin 11 Sound
; 74HC165 Pin 12 Light
; 74HC165 Pin 13 Tilt
; 74HC165 Pin 14 Upsidedown
; 74HC165 Pin 3 Tummy
; 74HC165 Pin 4 Back
; 74HC165 Pin 5 Reset
; 74HC165 Pin 6 n/a - never got accurate info from this one
; NOTES:
; This program uses 2400 buad rate without flow control. This
; program looks its best when used with the front end that was
; made for it.
;*****
; Define type of processor to use and include file of standard EQUs
```

```

;
    LIST P=16F84
    include "P16F84.INC"

;*****
; Define Registers Used
;*****

;Constants
MaxPointer    equ    10    ;3, maximum number Input Flag Reg.
Bundle        equ    11    ;20, maximum bunch of Gear Sensor
Eight         equ    12    ;8, maximum number of bits in a byte

;Delay Variables
DelayTemp     equ    13
DelayT2       equ    14
DelayTempS    equ    15
DelayTempSS   equ    16

;Database
FurbyINPUT1   equ    17    ;Input Flag Register
FurbyINPUT2   equ    18
FurbyINPUT3   equ    19
FurbyINPUT4   equ    20

;Gear Variables
EightBites    equ    21    ;Counter, just for eight bytes
Current_State equ    22    ;Hold the Current position of Furby(TM)
Gear_Counter  equ    23
Cam_Counter   equ    24
Inc_Counter   equ    25

;Temperary Variables
Temp          equ    26    ;Temperary General Register
Counter       equ    27    ;Temperary General Register/Counter
GearCycles    equ    28    ;Temperary holder for the number gear
                ;cycles
WantedPosition equ    29    ;Temperary holder for wanted position

;NOTES:
;384 cycles needed for 2400 buad rate :. 127 //417us
;95 cycless needed for 9600 buad rate :. 31 //104us
;16/18 cycles needed for 57200 buad rate :. 5
BuadRate      equ    30
SendCommandByte equ    31
SentBites     equ    32
ReceivedCommandByte equ    33

```

```

ReceiveBites      equ    34

;*****
; Beginning of the main part of the program
;*****
main
    ;PORTB::Input:1,4,6/Output:0,2,3,5
    ;Port 7 not used
    movlw    b'11010010'
    tris     PORTB

    ;PORTA::Input:3/Output:1,0
    ;Ports 7-4(not used)
    movlw    b'11111100'
    tris     PORTA

    call     SETPIC          ;Clear all Output ports

RESETPROGRAM
    call     RESET
    call     LongDelay
ContinueToCheckInputs
    clrf     EightBites
    btfsc   PORTB,1        ;if line is low, start bit is present
    goto    ContinueOn    ;received a high: No start bit yet, re- check
    btfss   PORTB,1        ;recieved a low, checks again
    btfsc   PORTB,1        ;a low was for sure received and now falls
                        ;through to the delay call for the start bit
    goto    ContinueOn
    call    StartBitDelay ;Have to waite 1.5 times the cycle for
                        ;the start
    call    RECEIVECOMMAND
            call    CHECKCOMMANDS
            nop

ContinueOn
    call    GETSRINPUT      ;Get Input from any of the
                        ;Furby(TM) Sensors
            movf    FSR,w
            sublw   FurbyINPUT4
            btfss   STATUS,Z
            goto    ContinueToCheckInputs
            call    CHECKINPUTS    ;check inputs to determine if there was
anything
            call    RESETVARIABLES
            goto    ContinueToCheckInputs

;*****

```

```

;
; FUNCTION CALLS/METHODS
; :. Below are all the function call made by the root of the
; program. Each function has its own duty which may call
; upon other function calls to complete the task. The most
; complicated function call may call an endless number of
; other function calls
;
;
;*****
;*****
; BACK
; The back sensor was touched. The command 'B' is then sent
; to the computer and the command wiggle is then called for
; execution
;*****
BACK
    movlw    0x42 ;B
            movwf  SendCommandByte
            call   SENDCOMMAND
            call   WIGGLE
            return
;*****
; Blink
; A 'b' was received from the computer serially. This funciton
; call's purpose is to memic a person blinking their eyes
;*****
BLINK
    movlw    0x10
            movwf  GearCycles    ;
            call   Move_Forward
            call   Delay
            movlw  0x05
            movwf  GearCycles    ;
            call   Move_Backwards
            call   LongerDelay
            call   RESET
            return
;*****
; Check Commands
; As serial information call in via RS-232. Each character is
; then checked against a predefined command. Once reconized,
; the command is then executed (called)
;*****
CHECKCOMMANDS
    movf     ReceivedCommandByte,w
    sublw   0x74 ;t
    btfss   STATUS,Z

```

f008.txt

```
        goto    CheckSleep
        call    Talking
        return

CheckSleep
    movf    ReceivedCommandByte,w
        sublw  0x73 ;s
        btfss  STATUS,Z
        goto    CheckScared
        call    Sleeping
        return

CheckScared
        movf    ReceivedCommandByte,w
        sublw  0x53 ;S
        btfss  STATUS,Z
        goto    CheckWingle
        call    Scared
        return

CheckWingle
        movf    ReceivedCommandByte,w
        sublw  0x77 ;w
        btfss  STATUS,Z
        goto    CheckCLOSE_MOUTH
        call    WIGGLE
        return

CheckCLOSE_MOUTH
        movf    ReceivedCommandByte,w
        sublw  0x63 ;c
        btfss  STATUS,Z
        goto    CheckQUITE
        call    CLOSE_MOUTH
        return

CheckQUITE
        movf    ReceivedCommandByte,w
        sublw  0x71 ;q
        btfss  STATUS,Z
        goto    Checkblink
        call    QUITE
        return

Checkblink
        movf    ReceivedCommandByte,w
        sublw  0x62 ;b
        btfss  STATUS,Z
        goto    CheckReset
        call    QUITE
        return

CheckReset
    movf    ReceivedCommandByte,w
        sublw  0x72 ;r
```

f008.txt

```
    btfss STATUS,Z
    goto  CheckRESET
    call  RESET
    return
```

CheckRESET

```
    movf  ReceivedCommandByte,w
    sublw 0x52 ;R
    btfss STATUS,Z
    goto  NoCommands
    call  RESET
    return
```

NoCommands

```
    return
```

```
;*****
; Check Furby(TM) Inputs
; This function call checks each sensor. I did not use
; interrupts and therefore had to be creative in how I would
; interpret if there was a sensor being used while still being
; able receive incoming RS-232 commands
;*****
```

CHECKINPUTS

;CheckReset

```
    btfsc FurbyINPUT1,0 ;looking for a 0
    goto  CheckBack
    btfsc FurbyINPUT2,0
    goto  CheckBack
    btfsc FurbyINPUT3,0
    goto  CheckBack
    call  RESET
    return
```

CheckBack

```
    btfsc FurbyINPUT1,1 ;looking for a 0
    goto  CheckTummy
    btfsc FurbyINPUT2,1
    goto  CheckTummy
    btfsc FurbyINPUT3,1
    goto  CheckTummy
    call  BACK
    return
```

CheckTummy

```
    btfsc FurbyINPUT1,2 ;looking for a 0
    goto  CheckFeed
    btfsc FurbyINPUT2,2
    goto  CheckFeed
    btfsc FurbyINPUT3,2
    goto  CheckFeed
    call  TUMMY
```

```

return
CheckFeed
    btfss    FurbyINPUT1,3 ;looking for a 1
    return
    btfss    FurbyINPUT2,3
    return
    btfss    FurbyINPUT3,3
    return
    call     FEED
    return
;*****
; Close Mouth
;   A 'c' was received serailly via RS-232. This function call
;   memics someone closing their mouth
;*****
CLOSE_MOUTH
    movlw   0x07
            movwf   GearCycles
            call    Move_Backwards
            call    LongerDelay
            call    RESET
            return
;*****
; Delay Routines
;   Below is a listing of a varity of delays, each having their
;   own unique function
;*****
LongerDelay ;A delay that the user can see
    movlw   .8
            movwf   DelayTempSS

delayler
    call    LongDelay
    decfsz DelayTempSS,f
    goto   delayler
    return

;.....
LongDelay ;Approx 125 mS delay
    movlw   .255
    movwf   DelayT2
ldelaya
    call    Delay
    decfsz DelayT2,f ;Decrement this register and
    goto   ldelaya ; keep going until it hits zero
    return
;.....
Delay ;Short delay
    movlw   .255 ;Load Temp register with constant

```

```

                                f008.txt
delaya    movwf    DelayTemp    ;for .3 ms
          decfsz  DelayTemp,f    ;Decrement until zero
          goto    delaya
          return

;.....
ShortDelay
          movlw   .100
          movwf   DelayTempS
delayS    decfsz  DelayTempS,f
          goto    delayS
          return

;.....
ShortestDelay
          movlw   .25
          movwf   DelayTempSS
delaySS   decfsz  DelayTempSS,f
          goto    delaySS
          return

;.....
SendDelay
;9600 need .25 and a nop
          movlw   .119
          movwf   BuadRate
SendLoop  decfsz  BuadRate,f
          goto    SendLoop
          nop
          nop
          return

;.....
ReceiveDelay
          movlw   .119
          movwf   BuadRate
ReceiveLoop
          decfsz  BuadRate,f
          goto    ReceiveLoop
          return

;.....
StartBitDelay
          movlw   .170
          movwf   BuadRate
StartBitLoop
          decfsz  BuadRate,f
          goto    StartBitLoop
          return

```


f008.txt

```

;*****
; FEEDME
;   One of the sensors was touched and now an 'F' is sent to the
;   computer and a little wiggle is executed
;*****
FEED
        movlw    0x46 ;f
        movwf    SendCommandByte
        call     SENDCOMMAND
        call     WIGGLE
        call     LongerDelay
        return

;*****
; Forward
;   Forward motion command function call. This was set up
;   originally so bit 3 and bit 2 are not set on at the same
;   time automatically.
;*****
FORWARD
        ;bcf     PORTB,3
        bsf     PORTB,2
        return

;*****
; GET INPUTS FROM SHIFT REGISTER
;   This function call takes in inputs from the shift register
;   serially thorough a shift register
;*****
GETSRINPUT
        bsf     PORTA,0          ;Enable the Shift Register
NextInputBit
        bsf     PORTA,1          ;Create the trigger
        bcf     PORTA,1

        btfscl PORTA,3          ;waiting for the start of the next bit
        bsf     STATUS,C        ;set the next bit (1)
        btfss  PORTA,3
        bcf     STATUS,C        ;clear the next bit (0)

        rrf     Temp,f          ;shift all the bits to the right
        incf   EightBites,f    ;increment the bit counter

        movf   EightBites,w    ;checking for the eight's bit
        sublw  .8              ;to make that byte
        btfss  STATUS,Z
        goto   NextInputBit    ;8 bits have not been received yet - agian
        bcf   PORTA,0          ;8 bits have been received now

        movf   Temp,w          ;Move the contents into the safe place

```

f008.txt

```
movwf INDF
incf FSR,f ;Increment the pointer
return
;*****
; Halloween Take, The
; I thought that eyes & mouth when open while the ears were
; straight up made a good scared or surprised position.. This
; function call is not relativent but was cute.
;*****
Scared
;I want to send a singal to the computer to play a scray noise
movlw 0x06
movwf GearCycles ;
call Move_Forward
call LongerDelay
call LongerDelay
call RESET
return
;*****
; Move Forward
; this funciton call counts the number of times the gears goes
; around so I can fake movement. this is not very accurate but
; is close enough that when making other packaged function call
; the toy looks as if it goes to the same place each time. It
; is not true. It depends on many factors and timing has a lot
; to do with it.
;*****
Move_Forward
movlw .0
movwf Gear_Counter ;need this one
movwf Inc_Counter ;need this one

KeepGoingForward
movlw .0
movwf Gear_Counter ;Clear the Gear Counter
KeepLookingForGEAR_Forward
call FORWARD
call ShortDelay
btfss PORTB,6 ;Check the Gear
goto KeepLookingForGEAR_Forward
call STOPMOTOR
incf Gear_Counter,f
movf Gear_Counter,w
sublw Bundle ;Move motors 20 pulses
btfss STATUS,Z
goto KeepLookingForGEAR_Forward
incf Inc_Counter,f
movf Inc_Counter,w
```

```

                                f008.txt
    subwf  GearCycles,w          ;The End yet?
    btfss  STATUS,Z
    goto   KeepGoingForward     ;Still have to move motors
    return
;*****
; Move Backwards
; this funciton call counts the number of times the gears goes
; around so I can fake movement. this is not very accurate but
; is close enough that when making other packaged function call
; the toy looks as if it goes to the same place each time. It
; is not true. It depends on many factors and timing has a lot
; to do with it.
;*****
Move_Backwards
    movlw  .0
    movwf  Gear_Counter        ;need this one
    movwf  Inc_Counter         ;need this one

KeepGoingBackwards
    movlw  .0
    movwf  Gear_Counter        ;Clear the Gear Counter
KeepLookingForGEAR_Backwards
    call   REVERSE
    call   ShortDelay
    btfss  PORTB,6             ;Check the Gear
    goto   KeepLookingForGEAR_Backwards
    call   STOPMOTOR
    incf   Gear_Counter,f
    movf   Gear_Counter,w
    sublw  Bundle              ;Move motors 20 pulses
    btfss  STATUS,Z
    goto   KeepLookingForGEAR_Backwards
    incf   Inc_Counter,f
    movf   Inc_Counter,w
    subwf  GearCycles,w        ;The End yet?
    btfss  STATUS,Z
    goto   KeepGoingBackwards  ;Still have to move motors
    return
;*****
; QUITE!!!
; A 'q' was received from the computer via RS-232. This is,
; if nothings else cute little function call. Not a compete
; routine package.
;*****
QUITE
    movlw  0x10
    movwf  GearCycles          ;
    call   Move_Forward

```

f008.txt

```
        call    LongerDelay
        call    RESET
        return
;*****
; HOME
; This function call is a primer. It is the most important
; function call. It is the 'home' position where the toy
; repositions itself after each function call. This function
; calls gives me the ability to fake movements : make furby
; appear to being mimicing something.
;*****
RETURNHOME
KeepLookingForCAM          ;Position Furby(TM) home
    call    REVERSE
    call    STOPMOTOR
    btfsc   PORTB,4         ;Check for CAM
    goto    KeepLookingForCAM
    call    STOPMOTOR
    movlw   .0
    movwf   Current_State  ;Hold current position
    return
;*****
; Receive a Command from the computer
; this function call was taking from my lab 3 (RS-232
; communication). It allows me to taking in information
; from the computer and interprete them correctly
;*****
RECEIVECOMMAND
    ;Just need to receive on byte(a command/option)
    ;ReceivedCommandByte
    clr     ReceiveBites
NextRXBit
    btfsc   PORTB, 1       ;waiting for the start of the next bit
    bsf     STATUS,C       ;set the next bit (1)
    btfss   PORTB, 1
    bcf     STATUS,C       ;clear the next bit (0)

    rrf     ReceivedCommandByte,f ;shift all the bits to the right
    incf    ReceiveBites,f ;increment the bit counter

    call    ReceiveDelay   ;need 104u second delay between bits

    movf    ReceiveBites,w ;checking for the eight's bit
    subwf   Eight,w        ;to make that byte
    btfss   STATUS,Z
    goto    NextRXBit      ;8 bits have not been received yet - agian
    return                  ;8 bits have been received - can return now
;*****
```

```

; Reset The Furby(TM)
;*****
RESET
    call    RESETVARIABLES
           call    RETURNHOME
    call    LongDelay
    return
;*****
; Reseting Variables
;   Addresses are reset to the beginning position and variables
;   are cleared
;*****
RESETVARIABLES
    movlw   .0
    movwf  FurbyINPUT1    ;Clear a Input Flag
    movwf  FurbyINPUT2
    movwf  FurbyINPUT3
    movwf  EightBites
           movwf  Temp
           movwf  Counter

           movlw  FurbyINPUT1    ;Making the pointer
           movwf  FSR

    return
;*****
; Reverse
;   This function call is for backward motion .This was set up
;   orginally so bit 3 and bit 2 are not set on at the same
;   time automatically.
;*****
REVERSE
    bcf    PORTB,2
    bsf    PORTB,3
    return
;*****
;Send a Command to the computer
;   this function call was taking from my lab 3 (RS-232
;   communication). It allows me to send information
;   to the computer and interprete them correctly
;*****
SENDCOMMAND
    ;Just need to send one byte (a command/option)
    clr   SentBites
           bcf    PORTB,0
           call   SendDelay    ;Start bit
NextTXBit
    btfsc SendCommandByte,0

```

```

                                f008.txt
    bsf    PORTB,0                ;set the next bit (1)
    btfss  SendCommandByte,0
    bcf    PORTB,0                ;clear the next bit (0)
    rrf    SendCommandByte,f      ;shift all the bits to the right
    incf   SentBites,f           ;increment the bit counter
    call   SendDelay
    movf   SentBites,w
    subwf  Eight,w
    btfss  STATUS,Z              ;Check if 8 bits have been sent
    goto   NextTXBit            ;8 bits have not been sent,
                                ;must continue
                                bsf    PORTB,0                ;End
    return
;*****
; Setup the Pic
; Purpose: Setup the states on the Outputs and initialize any
;          constances
;*****
SETPIC

    bsf    PORTB,0                ;RS-232 TxD1
    bcf    PORTB,2                ;Forward Control
    bcf    PORTB,3                ;Reverse Control
    bsf    PORTB,5                ;Turn the GEAR_LED_ON
                                ;and Leave it on

    bcf    PORTA,0                ;Shift Register Enable line
                                ;Active Low
        bcf    PORTA,1            ;CP1 Clock Control

    ;Constants
    movlw  .3
    movwf  MaxPointer            ;3, maximum number Input Flag Reg.
    movlw  .10
    movwf  Bundle                ;20, maximum bunch of Gear Sensor
    movlw  .8
    movwf  Eight                 ;8, maximum number of bits in a byte

    return
;*****
; Sleeping away
;   A 's' was received from the computer. The toy will now
;   memic someone sleeping but standing up :)
;*****
;Sleeping Away
Sleeping
    ;I would like to send a command to the computer to play a wave file
        movlw  0x13

```

f008.txt

```
movwf GearCycles ;
call Move_Forward
call LongerDelay
movlw 0x4
movwf GearCycles ;
call Move_Forward
call LongerDelay
movlw 0x6
movwf GearCycles ;
call Move_Backwards
call LongerDelay
movlw 0x5
movwf GearCycles ;
call Move_Forward
call LongerDelay
movlw 0x5
movwf GearCycles ;
call Move_Backwards
call LongerDelay
movlw 0x5
movwf GearCycles ;
call Move_Forward
call LongerDelay
movlw 0x5
movwf GearCycles ;
call Move_Forward
call LongerDelay
call LongerDelay
call RESET
return
```

;*****

```
; Stop the motor
; Making sure that both bits are set low, as to stop any
; movement
```

;*****

STOPMOTOR

```
bcf PORTB,2
bcf PORTB,3
return
```

;*****

```
; Talk
; A 't' was received from the computer. this function call makes
; the toy memic someone talking
```

;*****

;Talking Away

Talking

```
;I would like to send a command to the computer to play a wave file
;movlw 0x26
```

f008.txt

```
movlw    0x05
movwf    GearCycles    ;
call     Move_Forward
call     LongDelay
movlw    0x07
movwf    GearCycles    ;
call     Move_Backwards
call     LongDelay
movlw    0x04
movwf    GearCycles    ;
call     Move_Forward
call     LongDelay
movlw    0x07
movwf    GearCycles    ;
call     Move_Backwards
call     LongDelay
movlw    0x06
movwf    GearCycles    ;
call     Move_Forward
call     LongDelay
movlw    0x05
movwf    GearCycles    ;
call     Move_Backwards
call     LongDelay
movlw    0x05
movwf    GearCycles    ;
call     Move_Forward
call     LongDelay
movlw    0x05
movwf    GearCycles    ;
call     Move_Backwards
call     LongDelay
movlw    0x05
movwf    GearCycles    ;
call     Move_Forward
call     LongDelay
movlw    0x05
movwf    GearCycles    ;
call     Move_Backwards
call     LongDelay
movlw    0x05
movwf    GearCycles    ;
call     Move_Forward
call     LongDelay
movlw    0x05
movwf    GearCycles    ;
call     Move_Backwards
call     LongDelay
```



```

                                f008.txt
                                movlw 0x05
                                movwf GearCycles ;
                                call Move_Forward
                                call LongerDelay
                                call LongerDelay
                                call RESET
                                return
;*****
; Tummy was touched
; The tummy sensor was touched
;*****
TUMMY
                                movlw 0x54 ;T
                                movwf SendCommandByte
                                call SENDCOMMAND
                                call WIGGLE
                                return
;*****
; Wingle ears
; A cute and useless function call
;*****
WIGGLE
                                movlw 0x02
                                movwf GearCycles ;
                                call Move_Backwards
                                call LongDelay
                                movlw 0x03
                                movwf GearCycles ;
                                call Move_Forward
                                call LongerDelay
                                call RESET
                                return
;*****
; The End of the Program
;*****
                                END
;*****

```