```
        DATA @254,0                     ' Set record protect location to 0
                                        '      Pushbuttons:
ACTIVE  VAR GPIO.1                      ' Channel activation (input LOW), port 1
CONTROL VAR GPIO.2                      ' Control or RECORD (input LOW), port 2
TRIGR   VAR GPIO.3                      ' Trigger (input HIGH), port 3
                                        '      Indicators:
ALED    VAR GPIO.0                      ' Activity LED (output) on port 0
SLED    VAR GPIO.4                      ' Status LED (output) on port 4
RELAY   VAR GPIO.5                      ' Relay driver (output) on port 5

ADDRS   VAR WORD                        ' Address for event memory
CLOCK   VAR WORD                        ' Clock count for event timing
CYCLE   VAR BYTE                        ' Generic counter for status LED
FLASH   VAR BYTE                        ' Sets status LED flash rate
FLASH2  VAR BYTE                        ' Always set to (FLASH/2)+1
FRAME   VAR ADDRS.BYTE0                 ' Frame counter is low byte of ADDRS
PFLAG   VAR BYTE                        ' 0 or 1 to select memory protection
STATE   VAR BYTE                        ' Value from data memory & generic byte

INCLUDE "DT_INTS-14.BAS"                ' Enables timebase routine
INCLUDE "REENTER.BAS"                   ' Enables return from timebase

ASM
INT_LIST MACRO                          ; Define the interrupts
        INT_HANDLER TMR1_INT, _TIMER, PBP, YES
    ENDM
    INT_CREATE                          ; Create the interrupt processor
ENDASM

ANSEL =%00000000                        ' Set all ports digital
CMCON0=%00000111                        ' Disable the comparators
OSCCON=%01100000                        ' Configure oscillator 4MHz ($60)
TRISIO=%00001110                        ' Set GP5-4-0 outputs, 3-2-1 inputs

OPTION_REG.7=0                          ' Enable weak pull-ups
WPU=%00000110                           ' Set weak pull-ups on GP2-1

OSCTUNE=$06                             ' Set oscillator 4.8% fast
T1CON=$01                               ' Prescaler set to 16 interrupts/second

        RELAY=0 : ALED=0 : SLED=0       ' To begin, RELAY and both LEDs are off
                                        '  (NOTE: Do NOT use power-up timer!)
        CLOCK=0                         ' Initial value for CLOCK

        PAUSE 105                       ' Pause for start-up latency

@ INT_ENABLE TMR1_INT                   ; Enable interrupt for timebase

        READ 254,PFLAG                  ' Get data protect flag from memory
        IF TRIGR=1 THEN                 ' If held down at power-up, toggle flag
            CYCLE=0 : FLASH=2 : FLASH2=2' Status LED indicates "attention!"
            GOSUB PFSWAP
            ENDIF
        READ 254,PFLAG                  ' Get flag again in case it changed

        IF PFLAG=1 THEN                 ' If data protect is set, notify
```

```
            GOSUB NOTIFY                '  the user at power-up
            ENDIF

        IF CONTROL=0 THEN               ' If button held down at power-up,
            CYCLE=0 : FLASH=2 : FLASH2=2'  status LED indicates "attention!"
            GOSUB ERASE                 '  and data memory will be erased
            ENDIF

        CYCLE=0 : FLASH=16 : FLASH2=9   ' Default values for "idle" condition

MAIN:   IF TRIGR=1 THEN                 ' Selects "playback"
            PAUSE 10
            GOSUB PLAY
            ENDIF
        IF CONTROL=0 THEN               ' Selects "record"
            PAUSE 10
            GOSUB RECORD
            ENDIF
        IF ACTIVE=0 THEN                ' Manual override when idle
            PAUSE 10
            GOSUB MANUAL
            ENDIF
        GOTO MAIN

PFSWAP: IF PFLAG=0 THEN WRITE 254,1     ' If data is not protected, protect it
        IF PFLAG=1 THEN WRITE 254,0     ' If data is protected, unprotect it
        WHILE TRIGR=1                   ' Wait for the button to be released
            WEND                        '  before continuting
        RETURN

NOTIFY:                                 ' Notify user that data protect is set
@ INT_DISABLE TMR1_INT                  ; Disable the interrupt
        IF PFLAG=1 THEN SLED=1          ' Turn the status LED on continuously
        PAUSE 3150                      '  for 3 seconds,
        SLED=0                          '  then turn it off and
@ INT_ENABLE TMR1_INT                   ;  re-enable the interrupt
        RETURN

ERASE:  IF PFLAG=1 THEN RETURN          ' If data is protected, do not erase
        FOR STATE=0 TO 248              ' Erase data memory through location 248
            WRITE STATE,$FF             '  ("erasure" is loading all locations
            NEXT STATE                  '  with $FF, decimal 255)
        WHILE CONTROL=0                 ' Wait for the button to be released
            WEND                        '  before continuing
        RETURN                          ' Total erase time about 1.2 seconds

PLAY:   CYCLE=0 : FLASH=8 : FLASH2=5    ' Status LED indicates "playback"
        CLOCK=0                         ' Reset the clock for read
PLOOP:  IF FRAME=248 THEN GOTO PLEXIT   ' If top of data is reached, exit
        READ FRAME,STATE                ' Read the frame from data memory
        IF STATE=$FF THEN GOTO PLEXIT   ' If end-of-data marker is read, exit
        IF STATE=1 THEN                 ' For state 1,
            RELAY=1 : ALED=1            '  turn the relay and activity LED on,
            ENDIF                       '  otherwise,
        IF STATE=0 THEN                 '  for state 0,
            RELAY=0 : ALED=0            '  turn the relay and activity LED off
            ENDIF
        GOTO PLOOP                      ' Repeat until an exit condition occurs
PLEXIT: CYCLE=0 : FLASH=16 : FLASH2=9   ' Status LED indicates "idle"
        RETURN

RECORD: IF PFLAG=1 THEN RETURN          ' If data is protected, do not record
        CYCLE=0 : FLASH=2 : FLASH2=2    ' Status LED indicates "attention!"
        WHILE TRIGR<>1                  ' Wait for signal to begin recording
            WEND                        '  until trigger button is pressed
        CYCLE=0 : FLASH=4 : FLASH2=3    ' Status LED indicates "recording"
        CLOCK=0                         ' Reset the clock for write
RLOOP:  IF CONTROL=0 THEN GOTO RECEND   ' If control button is pressed, exit
        IF FRAME=248 THEN GOTO RANOUT   ' If top of data is reached, exit
        IF ACTIVE=0 THEN                ' If the activity button is pressed,
            RELAY=1 : ALED=1            '  turn relay and activity LED on and
            WRITE fRAME,1               '  write "on" (1) to data memory
            ELSE                        ' If the button is not pressed,
            RELAY=0 : ALED=0            '  turn relay and activity LED off and
            WRITE FRAME,0               '  write "off" (0) to data memory
            ENDIF
        GOTO RLOOP                      ' Repeat until an exit condition occurs
RECEND: CYCLE=0 : FLASH=2 : FLASH2=2    ' Status LED indicates "attention!"
        IF RELAY=1 AND ALED=1 THEN      ' If relay and activity LED were on
            RELAY=0 : ALED=0            '  when record ends, turn both off and
            WRITE FRAME,0               '  write "off" (0) in the current frame
            ENDIF
        WRITE (FRAME+1),$FF             ' Write end-of-data marker ($FF or 255)
        WHILE TRIGR<>1                  ' Wait for the signal to finish
            WEND                        '  until trigger button is pressed
        CYCLE=0 : FLASH=16 : FLASH2=9   ' Preset status LED to "idle"
        WHILE TRIGR=1                   ' Do nothing until button is released
            WEND                        ' When trigger button is released,
        PAUSE 1050                      '  pause for 1 second before
        GOTO REXIT                      '  returning to the idle state
RANOUT: CYCLE=0 : FLASH=2 : FLASH2=2    ' Status LED indicates "attention!"
        WRITE 247,0                     ' If at top of data memory, make sure
        RELAY=0 : ALED=0                '  last frame turns relay and activity
        PAUSE 3150                      '  LED off, then wait 3 seconds
REXIT:  CYCLE=0 : FLASH=16 : FLASH2=9   ' Status LED indicates "idle"
        RETURN

MANUAL: WHILE ACTIVE=0                  ' While the button is held down,
            ALED=1 : RELAY=1            '  the activity LED and the relay
            WEND                        '  will remain on and be turned
        ALED=0 : RELAY=0                '  off when the button is released
        RETURN                          '  (relay "mirrors" button action)

TIMER:  CYCLE=CYCLE+1                   ' Interrupt handler for timebase
        IF CYCLE<FLASH2 THEN            '   Manage the status indicator LED:
            SLED=1                      '     RATE  FLASH  FLASH2  INDICATES
            ELSE                        '     1 Hz    16     9     "Idle"
            SLED=0                      '     2 Hz     8     5     "Playback"
            ENDIF                       '     4 Hz     4     3     "Recording"
        IF CYCLE>FLASH THEN CYCLE=0     '     8 Hz     2     2     "Attention!"
        ADDRS=CLOCK>>1                  ' Binary shift right 1 bit for address
        CLOCK=CLOCK+1                   ' Increment the clock counter
@ INT_RETURN                            ; Return from interrupt

        END
```