WDTIMER PAGE 1

1 ;\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

2 ;

3 ; Author : ADI - Apps www.analog.com/MicroConverter

4 ;

5 ; Date : FEB 2000

6 ;

7 ; File : WDtimer.asm

8 ;

9 ; Hardware : ADuC824

10 ;

11 ; Description : Demonstrates use of the on-chip watchdog timer.

12 ; In normal operation, WD timer is refreshed by code

13 ; every 100ms, as indicated by a flashing LED (10 Hz).

14 ; In a runaway code condition (which can be simulated

15 ; here by pressing the INT0 button on the eval board)

16 ; code fails to refresh WD bits, the LED stays in the

17 ; off position, before the WD timer generates a

18 ; reset after a user selected time-out period (from

19 ; 15.6ms to 2000ms) has elapsed. The time-out period

20 ; in this routine is 2.0s for visual clarity.

21 ;

22 ; After a reset the light blinks at a slower rate 2Hz.

23 ; Future watchdog resets can be enabled in the same

24 ; way. A hard reset will clear the WDS bit and the

25 ; quicker flashes will occur again.

26 ;

27 ; note: be sure to remove the PSEN pull-down (LK2)

28 ; before allowing watchdog to time-out, or

29 ; you'll end up in serial download mode again

30 ; (the LED will stay in the on condition)

31 ; rather than recovering normal code execution.

32 ;

33 ;\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

34

35 $MOD824 ; Use 8052&ADuC824 predefined symbols

36

00B4 37 LED EQU P3.4 ; P3.4 drives red LED on eval board

00D5 38 ERROR EQU F0 ; the 'ERROR' flag is used here to

39 ; simulate an erroneous command that

40 ; sends code into an unknown state

41

42 ;\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

43 ; BEGINNING OF CODE

---- 44 CSEG

45

0000 46 ORG 0000h

47

0000 020060 48 JMP MAIN ; jump to main program

49 ;\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

50 ; EXTERNAL INTERRUPT VECTOR SPACE

0003 51 ORG 0003h ; (INT0 ISR)

52

0003 D2D5 53 SETB ERROR ; simulate an error condition..

54 ; ..when INT0 button is pressed

0005 32 55 RETI

56

57

58 ;\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

WDTIMER PAGE 2

59 ; MAIN PROGRAM

0060 60 ORG 0060h ; Start at address above interrupts

61

0060 62 MAIN:

0060 20C205 63 JB WDS, WDRESET

0063 7801 64 MOV R0, #01h ; this will blink the LED at 10Hz

0065 02006A 65 JMP START

66

0068 7805 67 WDRESET: MOV R0, #05h ; this will blink the LED at 2Hz

68

69 ; Enable external interupt to trigger simulated error condition...

70

006A D288 71 START: SETB IT0 ; make INT0 edge triggered

006C D2A8 72 SETB EX0 ; enable INT0 (button on eval board)

006E D2AF 73 SETB EA

74

75 ; Configure the Watchdog timer. It should be configured like this,

76 ; with the global interrupts turned off and setting WDWR to allow

77 ; writing to WDCON.

0070 C2AF 78 CLR EA

0072 D2C0 79 SETB WDWR

0074 75C072 80 MOV WDCON, #72h ; Enable Watchdog timer to cause

81 ; -2.0 second timeout period

82 ; -enable WDIR bit to generate

83 ; a reset and not an interrupt

0077 D2AF 84 SETB EA ; set global interrupts again

85

86 ; from this point forward, watchdog bits must be refreshed every

87 ; 2.0 seconds or less. if they are not, watchdog timer will

88 ; generate a reset.

89

0079 C2D5 90 CLR ERROR ; simulate error free operation

91

92 ; The below loop represents normal code execution...

93

007B E8 94 FLASH: MOV A, R0

007C 120090 95 CALL DELAY ; delay by 100ms x R0

007F B2B4 96 CPL LED ; blink (complement) the red LED

97

0081 C2AF 98 CLR EA ; refresh watchdog timer

0083 D2C0 99 SETB WDWR

0085 D2C1 100 SETB WDE

0087 D2AF 101 SETB EA

102

0089 30D5EF 103 JNB ERROR, FLASH ; jump if 'ERROR' flag is not set

104

105 ; The below endless loop represents run-away code execution...

106

008C C2B4 107 CLR LED ; turn LED off during runaway code

008E 80FE 108 JMP $ ; this endless loop is used to

109 ; represent an unknown state of

110 ; program execution

111

112 ; program will sit in the above endless loop until the watchdog

113 ; period (2000ms) has elapsed, at which time a reset will be

114 ; generated by the watchdog timer, thereby recovering the chip to

115 ; resume normal code execution.

116

WDTIMER PAGE 3

117

118 ;\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

119 ; 100ms DELAY

0090 120 DELAY: ; Delays by 100ms \* A

121 ; 100mSec based on 1.5728MHZ

122 ; Core Clock

123 ; i.e. default ADuC824 Clock

124

0090 FA 125 MOV R2,A ; Acc holds delay variable

0091 7B19 126 DLY0: MOV R3,#019h ; Set up delay loop0

0093 7CFE 127 DLY1: MOV R4,#0FEh ; Set up delay loop1

0095 DCFE 128 DJNZ R4,$ ; Dec R4 & Jump here until R2 is 0

0097 DBFA 129 DJNZ R3,DLY1 ; Dec R3 & Jump DLY1 until R1 is 0

0099 DAF6 130 DJNZ R2,DLY0 ; Dec R2 & Jump DLY0 until R0 is 0

009B 22 131 RET ; Return from subroutine

132

133 ;\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

134

135 END

VERSION 1.2h ASSEMBLY COMPLETE, 0 ERRORS FOUND

WDTIMER PAGE 4

DELAY. . . . . . . . . . . . . . C ADDR 0090H

DLY0 . . . . . . . . . . . . . . C ADDR 0091H

DLY1 . . . . . . . . . . . . . . C ADDR 0093H

EA . . . . . . . . . . . . . . . B ADDR 00AFH PREDEFINED

ERROR. . . . . . . . . . . . . . NUMB 00D5H

EX0. . . . . . . . . . . . . . . B ADDR 00A8H PREDEFINED

F0 . . . . . . . . . . . . . . . B ADDR 00D5H PREDEFINED

FLASH. . . . . . . . . . . . . . C ADDR 007BH

IT0. . . . . . . . . . . . . . . B ADDR 0088H PREDEFINED

LED. . . . . . . . . . . . . . . NUMB 00B4H

MAIN . . . . . . . . . . . . . . C ADDR 0060H

P3 . . . . . . . . . . . . . . . D ADDR 00B0H PREDEFINED

START. . . . . . . . . . . . . . C ADDR 006AH

WDCON. . . . . . . . . . . . . . D ADDR 00C0H PREDEFINED

WDE. . . . . . . . . . . . . . . B ADDR 00C1H PREDEFINED

WDRESET. . . . . . . . . . . . . C ADDR 0068H

WDS. . . . . . . . . . . . . . . B ADDR 00C2H PREDEFINED

WDWR . . . . . . . . . . . . . . B ADDR 00C0H PREDEFINED