DATAFLSH PAGE 1

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

2 ;

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

4 ;

5 ; Date : September 2000

6 ;

7 ; File : dataflsh.asm

8 ;

9 ; Hardware : ADuC824

10 ;

11 ; Description : Demonstrates use of the on-chip read/write 640 byte

12 ; FlashEE data memory space. Stores a sequence of

13 ; button presses (INT0 button on eval board) in data

14 ; FlashEE space. Replays sequence on LED when board

15 ; is reset or power cycled.

16 ; The ADuC824 stores the play sequece in data flash

17 ; until another is recorded with a new set of button

18 ; presses. To record a new sequence, just wait until

19 ; the current one finishes playing (LED is off) and

20 ; enter new sequence via button (INT0).

21 ;

22 ;\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

23

24 $MOD824 ; Use 8052&ADuC824 predefined symbols

25

00B4 26 LED EQU P3.4 ; P3.4 drives red LED on eval board

00B2 27 BUTTON EQU P3.2 ; button on eval board drives P3.2

00D5 28 PREVIOUS EQU F0 ; flag to hold previous button value

0001 29 READ EQU 01h ; FlashEE command: 'read page'

0002 30 WRITE EQU 02h ; FlashEE command: 'write page'

0004 31 VERIFY EQU 04h ; FlashEE command: 'verify page'

0005 32 ERASE EQU 05h ; FlashEE command: 'erase page'

0006 33 ERASEALL EQU 06h ; FlashEE command: 'erase all'

34 ;--------------------------------------------------------------------

35 ;\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

36 ; BEGINNING OF CODE

---- 37 CSEG

38

0000 39 ORG 0000h

40

0000 41 MAIN:

0000 C2B4 42 CLR LED ; turn LED off

0002 75C600 43 MOV EADRL,#0 ; set data FlashEE address to page 0

44

45 ; READ FLASH/EE DATA and indicate values via LED on and off times...

46

0005 47 READFLASH:

0005 75B901 48 MOV ECON,#READ ; read current 4byte page of FlashEE

49 ; into EDATA1,2,3,4

0008 E5BF 50 MOV A,EDATA4

000A B4010A 51 CJNE A,#1,RECORD ; if EDATA4 is 1, then page contains

52 ; a valid play sequence

53 ; => Play this sequence

54 ; otherwise jump to record mode

55

56

57 ;--------------------------------------------------------------------

000D 58 PLAYBACK:

DATAFLSH PAGE 2

000D 120046 59 CALL BLINK ; flash LED for period determined

60 ; by FlashEE data just read

0010 05C6 61 INC EADRL ; increment to next FlashEE page addr

0012 E5C6 62 MOV A,EADRL

0014 B4A0EE 63 CJNE A,#0A0h,READFLASH

64 ; if address is less than 160 then jump

65 ; to read the next page

66 ; when PLAYBACK is finished jump to RECORD mode

67

68 ;--------------------------------------------------------------------

0017 69 RECORD:

0017 C2B4 70 CLR LED

0019 20B2FD 71 JB BUTTON,$ ; wait for first button press

72

73 ; once button is pressed, erase dataflash

001C 75B906 74 MOV ECON,#ERASEALL ; clear all data FlashEE memory

001F 75C600 75 MOV EADRL,#0

76

77 ; record time of button press

78

0022 79 RECORD\_NEXT\_TIME:

0022 120063 80 CALL RECORDTIME

81

0025 8582BC 82 MOV EDATA1,DPL ; place DPTR in EDATA1,2,3

0028 8583BD 83 MOV EDATA2,DPH

002B 8584BE 84 MOV EDATA3,DPP

002E 75BF01 85 MOV EDATA4,#1 ; put 1 in EDATA4 as identifier

0031 75B902 86 MOV ECON,#WRITE ; write EDATA1-4 into pre-erased

87 ; page of FlashEE data memory

88

0034 75B904 89 MOV ECON,#VERIFY ; verify current page is same as

0037 E5B9 90 MOV A,ECON ; EDATA1-4. If same, ECON=0.

0039 70DC 91 JNZ RECORD ; if verify fails, jump to RECORD

92

003B 05C6 93 INC EADRL ; increment to next FlashEE page addr

003D E5C6 94 MOV A,EADRL

003F B4A0E0 95 CJNE A,#0A0h,RECORD\_NEXT\_TIME

96 ; record first 160 button presses only

97

98 ; when flash/EE data space is full turn off LED and wait

99 ; for a power cycle

0042 C2B4 100 CLR LED

0044 80FE 101 JMP $

102

103

104 ;====================================================================

105 ; FUNCTIONS

106 ;====================================================================

107

108

109 ;\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

110 ; SUBROUTINES

0046 111 BLINK:

112 ; Turn LED ON/OFF based on the time in EDATA3/2/1

0046 B2B4 113 CPL LED

114

0048 E4 115 CLR A

0049 F582 116 MOV DPL,A

DATAFLSH PAGE 3

004B F583 117 MOV DPH,A ; clear DPTR

004D F584 118 MOV DPP,A

119

004F 05BC 120 INC EDATA1 ; EDATA1 -> EDATA3 should be

0051 05BD 121 INC EDATA2 ; incremented for the below to work

0053 05BE 122 INC EDATA3

123

0055 124 BLINKLOOP:

125 ; the record loop takes 6 instruction cycles hence 4 NOPs are

126 ; required to make the Playback loop 6 instruction cycles also.

127 ; NOTE: the main Playback loop will jump to BLINKLOOP after

128 ; decrementing EDATA1 and hence the time required to decrement

129 ; EDATA2 (approx 1/256 time of main loop) and EDATA3 are ignored.

0055 00 130 NOP ; 1

0056 00 131 NOP ; 1

0057 00 132 NOP ; 1

0058 00 133 NOP ; 1

0059 D5BCF9 134 DJNZ EDATA1, BLINKLOOP ; 2

005C D5BDF6 135 DJNZ EDATA2, BLINKLOOP ; EDATA1 overflows back to FFh

005F D5BEF3 136 DJNZ EDATA3, BLINKLOOP ; EDATA2 overflows back to FFh

137

0062 22 138 RET

139

140 ;\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

141

0063 142 RECORDTIME:

143 ; Record how long button is pressed for and store in EDATA3/2/1

0063 E4 144 CLR A

0064 F582 145 MOV DPL,A

0066 F583 146 MOV DPH,A ; clear DPTR

0068 F584 147 MOV DPP,A

148

006A B2B4 149 CPL LED

150

151 ; measure how long the button is either pressed or released

152 ; for. If the button is pressed then the LED is on. If the

153 ; button is released then the LED is off.

006C 154 RECORDLOOP:

006C A3 155 INC DPTR ; incrementing DPTR.. ; 2

006D 30B404 156 JNB LED, CHKB ; 2

0070 30B2F9 157 JNB BUTTON,RECORDLOOP ; 2

158 ; keep recording while button is pressed

0073 22 159 RET

0074 20B2F5 160 CHKB: JB BUTTON,RECORDLOOP ; 2

161 ; keep recording while button is released

0077 22 162 RET

163

164 ; DPP,DPH,DPL now holds a number that represents the length of

165 ; time between button edges. this data will be stored in FlashEE

166 ; space for use in controlling LED on and off times in "play" mode.

167

168 ;\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

169

170

171 END

VERSION 1.2h ASSEMBLY COMPLETE, 0 ERRORS FOUND

DATAFLSH PAGE 4

BLINK. . . . . . . . . . . . . . C ADDR 0046H

BLINKLOOP. . . . . . . . . . . . C ADDR 0055H

BUTTON . . . . . . . . . . . . . NUMB 00B2H

CHKB . . . . . . . . . . . . . . C ADDR 0074H

DPH. . . . . . . . . . . . . . . D ADDR 0083H PREDEFINED

DPL. . . . . . . . . . . . . . . D ADDR 0082H PREDEFINED

DPP. . . . . . . . . . . . . . . D ADDR 0084H PREDEFINED

EADRL. . . . . . . . . . . . . . D ADDR 00C6H PREDEFINED

ECON . . . . . . . . . . . . . . D ADDR 00B9H PREDEFINED

EDATA1 . . . . . . . . . . . . . D ADDR 00BCH PREDEFINED

EDATA2 . . . . . . . . . . . . . D ADDR 00BDH PREDEFINED

EDATA3 . . . . . . . . . . . . . D ADDR 00BEH PREDEFINED

EDATA4 . . . . . . . . . . . . . D ADDR 00BFH PREDEFINED

ERASE. . . . . . . . . . . . . . NUMB 0005H NOT USED

ERASEALL . . . . . . . . . . . . NUMB 0006H

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

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

MAIN . . . . . . . . . . . . . . C ADDR 0000H NOT USED

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

PLAYBACK . . . . . . . . . . . . C ADDR 000DH NOT USED

PREVIOUS . . . . . . . . . . . . NUMB 00D5H NOT USED

READ . . . . . . . . . . . . . . NUMB 0001H

READFLASH. . . . . . . . . . . . C ADDR 0005H

RECORD . . . . . . . . . . . . . C ADDR 0017H

RECORDLOOP . . . . . . . . . . . C ADDR 006CH

RECORDTIME . . . . . . . . . . . C ADDR 0063H

RECORD\_NEXT\_TIME . . . . . . . . C ADDR 0022H

VERIFY . . . . . . . . . . . . . NUMB 0004H

WRITE. . . . . . . . . . . . . . NUMB 0002H