FLASHEE PAGE 1

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

2 ;

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

4 ;

5 ; Date : 28 May 1999

6 ;

7 ; File : FlashEE.asm

8 ;

9 ; Hardware : ADuC812

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. Will store the sequence

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

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

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

19 ; enter new sequence via button (INT0).

20 ;

21 ; NOTE: : DO NOT write to FlashEE addresses above page 159!

22 ; ----- The 640 bytes are stored in pages 0 thru 159 (9Fh)

23 ; as four bytes per page. Writing to pages above

24 ; those documented can permanently lock you out of

25 ; the chip. See ADuC812 data sheet and errata sheet

26 ; for details (www.analog.com/microconverter).

27 ;

28 ;\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

29

30 $MOD812 ; Use 8052&ADuC812 predefined symbols

31

00B4 32 LED EQU P3.4 ; P3.4 drives red LED on eval board

00B2 33 BUTTON EQU P3.2 ; button on eval board drives P3.2

00D5 34 PREVIOUS EQU F0 ; flag to hold previous button value

0001 35 READ EQU 01h ; FlashEE command: 'read page'

0002 36 WRITE EQU 02h ; FlashEE command: 'write page'

0004 37 VERIFY EQU 04h ; FlashEE command: 'verify page'

0005 38 ERASE EQU 05h ; FlashEE command: 'erase page'

0006 39 ERASEALL EQU 06h ; FlashEE command: 'erase all'

40

41 ;\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

42 ; BEGINNING OF CODE

---- 43 CSEG

44

0000 45 ORG 0000h

46

0000 D2B4 47 SETB LED ; turn LED off

0002 740F 48 MOV A,#15

0004 12009C 49 CALL DELAY ; pause 1.5 seconds

50

0007 75C600 51 MOV EADRL,#0 ; set data FlashEE address to page 0

52

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

54

000A 75B901 55 READPG: MOV ECON,#READ ; read current 4byte page of FlashEE

56 ; into EDATA1,2,3,4

000D 7401 57 MOV A,#1

000F B5BF0C 58 CJNE A,EDATA4,EMPTY ; if EDATA4 is 1, then page contains

FLASHEE PAGE 2

59 ; valid data from previous write

60 ; otherwise, jump to EMPTY

61

0012 12005A 62 CALL BLINK ; flash LED for period determined

63 ; by FlashEE data just read

64

0015 05C6 65 INC EADRL ; increment to next FlashEE page addr

0017 E5C6 66 MOV A,EADRL

0019 B4A000 67 CJNE A,#0A0h,CMPRR ; if address is less than 160..

001C 40EC 68 CMPRR: JC READPG ; ..then jump to read the next page

69 ; (DO NOT write to pages above 159!)

70

71 ; WHEN "PLAY" SEQUENCE IS FINIISHED, wait for button press...

72

001E D2B4 73 EMPTY: SETB LED

0020 20B2FD 74 JB BUTTON,$ ; wait for first button press

0023 7401 75 MOV A,#1

0025 12009C 76 CALL DELAY ; pause 100ms

0028 20B2F3 77 JB BUTTON,EMPTY ; ensure button wasn't false trigger

78

79 ; IF BUTTON PRESSED, then ERASE and go into "RECORD" mode...

80

002B 75B906 81 MOV ECON,#ERASEALL ; clear all data FlashEE memory

002E 75C600 82 MOV EADRL,#0

83

84 ; CAPTURE AND RECORD in FLASH/EE space time between button edges...

85

0031 12007E 86 GETNEW: CALL GETVAL

87

0034 8582BC 88 MOV EDATA1,DPL ; place DPTR in EDATA1,2,3

0037 8583BD 89 MOV EDATA2,DPH

003A 8584BE 90 MOV EDATA3,DPP

003D 75BF01 91 MOV EDATA4,#1 ; put 1 in EDATA4 as identifier

0040 75B905 92 MOV ECON,#ERASE

0043 75B902 93 MOV ECON,#WRITE ; write EDATA1-4 into current page of

94 ; FlashEE data memory

95

0046 75B904 96 MOV ECON,#VERIFY ; verify current page is same as..

0049 E5B9 97 MOV A,ECON ; ..EDATA1-4. if same, ECON <- 0

004B 70D1 98 JNZ EMPTY ; if verify fails, jump to EMPTY

99

004D 05C6 100 INC EADRL ; increment to next FlashEE page addr

004F E5C6 101 MOV A,EADRL

0051 B4A000 102 CJNE A,#0A0h,CMPRG ; if EADRL<A0h..

0054 40DB 103 CMPRG: JC GETNEW ; ..then jump to get the next value

104 ; (DO NOT write to pages above 159!)

105

106 ; WHEN FLASH/EE DATA SPACE IS FULL...

107

0056 D2B4 108 SETB LED ; code will end up here only after 160 button

109 ; edges. no more can be recorded, so code

0058 80FE 110 JMP $ ; just waits for a reset or power cycle.

111

112 ;\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

113 ; SUBROUTINES

114

005A 115 BLINK: ; turn LED on or off for the duration

116 ; based on the value in EDATA3/2/1

FLASHEE PAGE 3

005A B2B4 117 CPL LED

118

005C 758200 119 MOV DPL,#0

005F 758300 120 MOV DPH,#0 ; clear DPTR

0062 758400 121 MOV DPP,#0

122

0065 A3 123 AGAIN1: INC DPTR ; increment DPTR.. 2

0066 E584 124 MOV A,DPP ; 1

0068 B5BE0A 125 CJNE A,EDATA3,CMPR3 ; 2

006B E583 126 MOV A,DPH ; 1

006D B5BD08 127 CJNE A,EDATA2,CMPR2 ; 2

0070 E582 128 MOV A,DPL ; 1

0072 B5BC06 129 CJNE A,EDATA1,CMPR1 ; 2

0075 00 130 CMPR3: NOP ; 1

0076 00 131 NOP ; 1

0077 00 132 NOP ; 1

0078 00 133 CMPR2: NOP ; 1

0079 00 134 NOP ; 1

007A 00 135 NOP ; 1

007B 40E8 136 CMPR1: JC AGAIN1 ; ..until DPTR>=EDATA3/2/1 2

137

007D 22 138 RET

139

140 ; this routine directly controls LED on and off times based on data

141 ; previously stored by a similar routine (GETVAL) which measures

142 ; BUTTON on and off times.

143

144 ;\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

145

007E 146 GETVAL: ; get a value based on duration of

147 ; button press

007E 758200 148 MOV DPL,#0

0081 758300 149 MOV DPH,#0 ; clear DPTR

0084 758400 150 MOV DPP,#0

151

0087 B2B4 152 CPL LED

153

0089 A3 154 AGAIN2: INC DPTR ; keep incrementing DPTR.. 2

008A 00 155 NOP ; 1

008B 00 156 NOP ; 1

008C 00 157 NOP ; 1

008D 00 158 NOP ; 1

008E 00 159 NOP ; 1

008F 00 160 NOP ; 1

0090 00 161 NOP ; 1

0091 30B404 162 JNB LED,CHKB ; 2

0094 30B2F2 163 JNB BUTTON,AGAIN2 ; 2

0097 22 164 RET ; ..until the button changes state

0098 20B2EE 165 CHKB: JB BUTTON,AGAIN2 ; 2

009B 22 166 RET

167

168 ; DPTR (DPP,DPH,DPL) now holds a number that represents the length of

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

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

171

172 ;\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

173

009C 174 DELAY: ; delay 100ms \* A

FLASHEE PAGE 4

175

009C 7FC8 176 DLY0: MOV R7,#200 ; 200 \* 500us = 100ms

009E 7EE5 177 DLY1: MOV R6,#229 ; 229 \* 2.17us = 500us

00A0 DEFE 178 DJNZ R6,$ ; sit here for 500us

00A2 DFFA 179 DJNZ R7,DLY1 ; repeat 200 times (100ms total)

00A4 D5E0F5 180 DJNZ ACC,DELAY ; repeat 100ms delay ACC times

00A7 22 181 RET

182

183 ;\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

184

185 END

VERSION 1.2h ASSEMBLY COMPLETE, 0 ERRORS FOUND

FLASHEE PAGE 5

ACC. . . . . . . . . . . . . . . D ADDR 00E0H PREDEFINED

AGAIN1 . . . . . . . . . . . . . C ADDR 0065H

AGAIN2 . . . . . . . . . . . . . C ADDR 0089H

BLINK. . . . . . . . . . . . . . C ADDR 005AH

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

CHKB . . . . . . . . . . . . . . C ADDR 0098H

CMPR1. . . . . . . . . . . . . . C ADDR 007BH

CMPR2. . . . . . . . . . . . . . C ADDR 0078H

CMPR3. . . . . . . . . . . . . . C ADDR 0075H

CMPRG. . . . . . . . . . . . . . C ADDR 0054H

CMPRR. . . . . . . . . . . . . . C ADDR 001CH

DELAY. . . . . . . . . . . . . . C ADDR 009CH

DLY0 . . . . . . . . . . . . . . C ADDR 009CH NOT USED

DLY1 . . . . . . . . . . . . . . C ADDR 009EH

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

EMPTY. . . . . . . . . . . . . . C ADDR 001EH

ERASE. . . . . . . . . . . . . . NUMB 0005H

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

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

GETNEW . . . . . . . . . . . . . C ADDR 0031H

GETVAL . . . . . . . . . . . . . C ADDR 007EH

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

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

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

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

READPG . . . . . . . . . . . . . C ADDR 000AH

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

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