DACSYNC PAGE 1

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

2 ;

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

4 ;

5 ; Date : October 2003

6 ;

7 ; File : DACsync.asm

8 ;

9 ; Hardware : ADuC842

10 ;

11 ; Description : Outputs sine waves on DAC0 and DAC1 at 590Hz.

12 ; Output signals are in quadrature with eachother,

13 ; DAC1 leading DAC0 by 90 degrees. the SYNC bit is

14 ; used to ensure that both DAC outputs update

15 ; simultaneously thus avoiding a phase error of 0.625

16 ; degrees.

17 ;

18 ;

19 ;\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

20

21 $MOD842 ; Use 8052&ADuC842 predefined symbols

22

00B4 23 LED EQU P3.4 ; P3.4 drives red LED on eval board

24

25 ;\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

26 ; BEGINNING OF CODE

---- 27 CSEG

28

0000 29 ORG 0000h

0000 75EF80 30 MOV ADCCON1,#80H

0003 75D703 31 MOV PLLCON,#03H ;

0006 75FD1F 32 MOV DACCON,#01Fh ; both DACs on,12bit,asynchronous

0009 75FA08 33 MOV DAC0H,#008h

000C 75F900 34 MOV DAC0L,#000h ; DAC0 to mid-scale

000F 75FC0F 35 MOV DAC1H,#00Fh

0012 75FBFF 36 MOV DAC1L,#0FFh ; DAC1 to full-scale

37

0015 901000 38 MOV DPTR,#TABLE

39

0018 53FDFB 40 STEP: ANL DACCON,#0FBh ; clear SYNC bit

41

001B E4 42 CLR A ;

001C 93 43 MOVC A,@A+DPTR ; get high byte for mainDAC..

001D F5FA 44 MOV DAC0H,A ; ..and move it into DAC0 register

001F 7420 45 MOV A,#020h ; offset by 90deg for quadratureDAC

0021 93 46 MOVC A,@A+DPTR ; get high byte for quadratureDAC..

0022 F5FC 47 MOV DAC1H,A ; ..and move it into DAC1 register

0024 A3 48 INC DPTR ; move on to get low bytes

49

0025 E4 50 CLR A ;

0026 93 51 MOVC A,@A+DPTR ; get low byte for mainDAC..

0027 F5F9 52 MOV DAC0L,A ; ..and update DAC0

0029 7420 53 MOV A,#020h ; offset by 90deg for quadratureDAC

002B 93 54 MOVC A,@A+DPTR ; get low byte for quadratureDAC..

002C F5FB 55 MOV DAC1L,A ; ..and update DAC1

002E A3 56 INC DPTR ; move on for next data point

57

002F 43FD04 58 ORL DACCON,#004h ; set SYNC bit

DACSYNC PAGE 2

59

0032 53827F 60 ANL DPL,#07Fh ; wrap around at end of table

61

0035 E5FA 62 MOV A,DAC0H ;

0037 A2E3 63 MOV C,ACC.3 ; MSB of DAC0 value

0039 92B4 64 MOV LED,C ; LED = MSB of DAC0

65

003B 00 66 NOP ;

003C 00 67 NOP ;

68

003D 80D9 69 JMP STEP ;

70

71

72

73 ;\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

74 ; SINE LOOKUP TABLE

1000 75 ORG 01000h

76

1000 77 TABLE:

78

1000 07FF 79 DB 007h, 0FFh

1002 08C8 80 DB 008h, 0C8h

1004 098E 81 DB 009h, 08Eh

1006 0A51 82 DB 00Ah, 051h

1008 0B0F 83 DB 00Bh, 00Fh

100A 0BC4 84 DB 00Bh, 0C4h

100C 0C71 85 DB 00Ch, 071h

100E 0D12 86 DB 00Dh, 012h

1010 0DA7 87 DB 00Dh, 0A7h

1012 0E2E 88 DB 00Eh, 02Eh

1014 0EA5 89 DB 00Eh, 0A5h

1016 0F0D 90 DB 00Fh, 00Dh

1018 0F63 91 DB 00Fh, 063h

101A 0FA6 92 DB 00Fh, 0A6h

101C 0FD7 93 DB 00Fh, 0D7h

101E 0FF5 94 DB 00Fh, 0F5h

1020 0FFF 95 DB 00Fh, 0FFh

1022 0FF5 96 DB 00Fh, 0F5h

1024 0FD7 97 DB 00Fh, 0D7h

1026 0FA6 98 DB 00Fh, 0A6h

1028 0F63 99 DB 00Fh, 063h

102A 0F0D 100 DB 00Fh, 00Dh

102C 0EA5 101 DB 00Eh, 0A5h

102E 0E2E 102 DB 00Eh, 02Eh

1030 0DA7 103 DB 00Dh, 0A7h

1032 0D12 104 DB 00Dh, 012h

1034 0C71 105 DB 00Ch, 071h

1036 0BC4 106 DB 00Bh, 0C4h

1038 0B0F 107 DB 00Bh, 00Fh

103A 0A51 108 DB 00Ah, 051h

103C 098E 109 DB 009h, 08Eh

103E 08C8 110 DB 008h, 0C8h

1040 07FF 111 DB 007h, 0FFh

1042 0736 112 DB 007h, 036h

1044 0670 113 DB 006h, 070h

1046 05AD 114 DB 005h, 0ADh

1048 04EF 115 DB 004h, 0EFh

104A 043A 116 DB 004h, 03Ah

DACSYNC PAGE 3

104C 038D 117 DB 003h, 08Dh

104E 02EC 118 DB 002h, 0ECh

1050 0257 119 DB 002h, 057h

1052 01D0 120 DB 001h, 0D0h

1054 0159 121 DB 001h, 059h

1056 00F1 122 DB 000h, 0F1h

1058 009B 123 DB 000h, 09Bh

105A 0058 124 DB 000h, 058h

105C 0027 125 DB 000h, 027h

105E 0009 126 DB 000h, 009h

1060 0000 127 DB 000h, 000h

1062 0009 128 DB 000h, 009h

1064 0027 129 DB 000h, 027h

1066 0058 130 DB 000h, 058h

1068 009B 131 DB 000h, 09Bh

106A 00F1 132 DB 000h, 0F1h

106C 0159 133 DB 001h, 059h

106E 01D0 134 DB 001h, 0D0h

1070 0257 135 DB 002h, 057h

1072 02EC 136 DB 002h, 0ECh

1074 038D 137 DB 003h, 08Dh

1076 043A 138 DB 004h, 03Ah

1078 04EF 139 DB 004h, 0EFh

107A 05AD 140 DB 005h, 0ADh

107C 0670 141 DB 006h, 070h

107E 0736 142 DB 007h, 036h ; end of table

143

1080 07FF 144 DB 007h, 0FFh ; repeat first 90degrees for quadratureDAC

1082 08C8 145 DB 008h, 0C8h

1084 098E 146 DB 009h, 08Eh

1086 0A51 147 DB 00Ah, 051h

1088 0B0F 148 DB 00Bh, 00Fh

108A 0BC4 149 DB 00Bh, 0C4h

108C 0C71 150 DB 00Ch, 071h

108E 0D12 151 DB 00Dh, 012h

1090 0DA7 152 DB 00Dh, 0A7h

1092 0E2E 153 DB 00Eh, 02Eh

1094 0EA5 154 DB 00Eh, 0A5h

1096 0F0D 155 DB 00Fh, 00Dh

1098 0F63 156 DB 00Fh, 063h

109A 0FA6 157 DB 00Fh, 0A6h

109C 0FD7 158 DB 00Fh, 0D7h

109E 0FF5 159 DB 00Fh, 0F5h

10A0 0FFF 160 DB 00Fh, 0FFh

161

162 ;\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

163

164 END

VERSION 1.2h ASSEMBLY COMPLETE, 0 ERRORS FOUND

DACSYNC PAGE 4

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

ADCCON1. . . . . . . . . . . . . D ADDR 00EFH PREDEFINED

DAC0H. . . . . . . . . . . . . . D ADDR 00FAH PREDEFINED

DAC0L. . . . . . . . . . . . . . D ADDR 00F9H PREDEFINED

DAC1H. . . . . . . . . . . . . . D ADDR 00FCH PREDEFINED

DAC1L. . . . . . . . . . . . . . D ADDR 00FBH PREDEFINED

DACCON . . . . . . . . . . . . . D ADDR 00FDH PREDEFINED

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

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

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

PLLCON . . . . . . . . . . . . . D ADDR 00D7H PREDEFINED

STEP . . . . . . . . . . . . . . C ADDR 0018H

TABLE. . . . . . . . . . . . . . C ADDR 1000H