DACQUADR PAGE 1

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

2 ;

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

4 ;

5 ; Date : 3 April 2002

6 ;

7 ; File : DACquadr.asm

8 ;

9 ; Hardware : ADuC832

10 ;

11 ; Description : Outputs sine waves on DAC0 and DAC1 at 606Hz.

12 ; Output signals are in quadrature with eachother,

13 ; DAC1 leading DAC0 by 90 degrees. since each DAC is

14 ; updated when its DACxL register is written to, they

15 ; are not updated at the exact same moment, and a

16 ; phase error of (in this case) 0.625degrees results.

17 ; to address this problem, see code: "DACsync.asm".

18 ; Rate calculations assume a core frequency of 16.777216MHz

19 ;

20 ;\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

21

22 $MOD832 ; Use 8052&ADuC832 predefined symbols

23

00B4 24 LED EQU P3.4 ; P3.4 drives red LED on eval board

25

26 ;\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

27 ; BEGINNING OF CODE

---- 28 CSEG

29

0000 30 ORG 0000h

0000 75EF80 31 MOV ADCCON1,#80H

0003 75D700 32 MOV PLLCON,#00h ; PLL to max frequency

0006 75FD1F 33 MOV DACCON,#01Fh ; both DACs on,12bit,asynchronous

0009 75FA08 34 MOV DAC0H,#008h

000C 75F900 35 MOV DAC0L,#000h ; DAC0 to mid-scale

000F 75FC0F 36 MOV DAC1H,#00Fh

0012 75FBFF 37 MOV DAC1L,#0FFh ; DAC1 to full-scale

38

0015 901000 39 MOV DPTR,#TABLE

40

0018 E4 41 STEP: CLR A ; 1

0019 93 42 MOVC A,@A+DPTR ; get high byte for mainDAC.. 2

001A F5FA 43 MOV DAC0H,A ; ..and move it into DAC0 register 1

001C 7420 44 MOV A,#020h ; offset by 90deg for quadratureDAC 1

001E 93 45 MOVC A,@A+DPTR ; get high byte for quadratureDAC.. 2

001F F5FC 46 MOV DAC1H,A ; ..and move it into DAC1 register 1

0021 A3 47 INC DPTR ; move on to get low bytes 2

48

0022 E4 49 CLR A ; 1

0023 93 50 MOVC A,@A+DPTR ; get low byte for mainDAC.. 2

0024 F5F9 51 MOV DAC0L,A ; ..and update DAC0 1

0026 7420 52 MOV A,#020h ; offset by 90deg for quadratureDAC 1

0028 93 53 MOVC A,@A+DPTR ; get low byte for quadratureDAC.. 2

0029 F5FB 54 MOV DAC1L,A ; ..and update DAC1 1

002B A3 55 INC DPTR ; move on for next data point 2

56

002C 53827F 57 ANL DPL,#07Fh ; wrap around at end of table 2

58

DACQUADR PAGE 2

002F E5FA 59 MOV A,DAC0H ; 1

0031 A2E3 60 MOV C,ACC.3 ; MSB of DAC0 value 1

0033 92B4 61 MOV LED,C ; LED = MSB of DAC0 2

62

0035 00 63 NOP ; 1

0036 00 64 NOP ; 1

0037 00 65 NOP ; 1

0038 00 66 NOP ; 1

0039 00 67 NOP ; 1

003A 00 68 NOP ; 1

003B 00 69 NOP ; 1

003C 00 70 NOP ; 1

71

003D 80D9 72 JMP STEP ; 2

73

74 ; numbers at right in the above loop represent the number of machine

75 ; cycles for each instruction. the complete loop takes exactly 36

76 ; machine cycles. with an 16.777216MHz master clock, a machine cycle

77 ; is 715ns, so the above loop takes 25.74us to update each data

78 ; point. since there are 64 data points in the below sine lookup

79 ; table, this results in a 1.64ms period, i.e. a 606Hz frequency.

80

81 ;\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

82 ; SINE LOOKUP TABLE

1000 83 ORG 01000h

84

1000 85 TABLE:

86

1000 07FF 87 DB 007h, 0FFh

1002 08C8 88 DB 008h, 0C8h

1004 098E 89 DB 009h, 08Eh

1006 0A51 90 DB 00Ah, 051h

1008 0B0F 91 DB 00Bh, 00Fh

100A 0BC4 92 DB 00Bh, 0C4h

100C 0C71 93 DB 00Ch, 071h

100E 0D12 94 DB 00Dh, 012h

1010 0DA7 95 DB 00Dh, 0A7h

1012 0E2E 96 DB 00Eh, 02Eh

1014 0EA5 97 DB 00Eh, 0A5h

1016 0F0D 98 DB 00Fh, 00Dh

1018 0F63 99 DB 00Fh, 063h

101A 0FA6 100 DB 00Fh, 0A6h

101C 0FD7 101 DB 00Fh, 0D7h

101E 0FF5 102 DB 00Fh, 0F5h

1020 0FFF 103 DB 00Fh, 0FFh

1022 0FF5 104 DB 00Fh, 0F5h

1024 0FD7 105 DB 00Fh, 0D7h

1026 0FA6 106 DB 00Fh, 0A6h

1028 0F63 107 DB 00Fh, 063h

102A 0F0D 108 DB 00Fh, 00Dh

102C 0EA5 109 DB 00Eh, 0A5h

102E 0E2E 110 DB 00Eh, 02Eh

1030 0DA7 111 DB 00Dh, 0A7h

1032 0D12 112 DB 00Dh, 012h

1034 0C71 113 DB 00Ch, 071h

1036 0BC4 114 DB 00Bh, 0C4h

1038 0B0F 115 DB 00Bh, 00Fh

103A 0A51 116 DB 00Ah, 051h

DACQUADR PAGE 3

103C 098E 117 DB 009h, 08Eh

103E 08C8 118 DB 008h, 0C8h

1040 07FF 119 DB 007h, 0FFh

1042 0736 120 DB 007h, 036h

1044 0670 121 DB 006h, 070h

1046 05AD 122 DB 005h, 0ADh

1048 04EF 123 DB 004h, 0EFh

104A 043A 124 DB 004h, 03Ah

104C 038D 125 DB 003h, 08Dh

104E 02EC 126 DB 002h, 0ECh

1050 0257 127 DB 002h, 057h

1052 01D0 128 DB 001h, 0D0h

1054 0159 129 DB 001h, 059h

1056 00F1 130 DB 000h, 0F1h

1058 009B 131 DB 000h, 09Bh

105A 0058 132 DB 000h, 058h

105C 0027 133 DB 000h, 027h

105E 0009 134 DB 000h, 009h

1060 0000 135 DB 000h, 000h

1062 0009 136 DB 000h, 009h

1064 0027 137 DB 000h, 027h

1066 0058 138 DB 000h, 058h

1068 009B 139 DB 000h, 09Bh

106A 00F1 140 DB 000h, 0F1h

106C 0159 141 DB 001h, 059h

106E 01D0 142 DB 001h, 0D0h

1070 0257 143 DB 002h, 057h

1072 02EC 144 DB 002h, 0ECh

1074 038D 145 DB 003h, 08Dh

1076 043A 146 DB 004h, 03Ah

1078 04EF 147 DB 004h, 0EFh

107A 05AD 148 DB 005h, 0ADh

107C 0670 149 DB 006h, 070h

107E 0736 150 DB 007h, 036h ; end of table

151

1080 07FF 152 DB 007h, 0FFh ; repeat first 90degrees for quadratureDAC

1082 08C8 153 DB 008h, 0C8h

1084 098E 154 DB 009h, 08Eh

1086 0A51 155 DB 00Ah, 051h

1088 0B0F 156 DB 00Bh, 00Fh

108A 0BC4 157 DB 00Bh, 0C4h

108C 0C71 158 DB 00Ch, 071h

108E 0D12 159 DB 00Dh, 012h

1090 0DA7 160 DB 00Dh, 0A7h

1092 0E2E 161 DB 00Eh, 02Eh

1094 0EA5 162 DB 00Eh, 0A5h

1096 0F0D 163 DB 00Fh, 00Dh

1098 0F63 164 DB 00Fh, 063h

109A 0FA6 165 DB 00Fh, 0A6h

109C 0FD7 166 DB 00Fh, 0D7h

109E 0FF5 167 DB 00Fh, 0F5h

10A0 0FFF 168 DB 00Fh, 0FFh

169

170 ;\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

171

172 END

VERSION 1.2h ASSEMBLY COMPLETE, 0 ERRORS FOUND

DACQUADR 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