C51 COMPILER V5.50, URT842 17/10/03 17:49:26 PAGE 1

DOS C51 COMPILER V5.50, COMPILATION OF MODULE URT842

OBJECT MODULE PLACED IN URT842.OBJ

COMPILER INVOKED BY: C:\ADUC\BIN\C51.EXE URT842.C DB

stmt level source

1 //urt842.c

2 /\*

3 Author: Eckart Hartmann Date:16/10/2003

4 Description of Software:

5 This program demonstrates the UART functions <A HREF="/mcc/softw/842/urt/Urt842Cfg.html">UrtCfg()</A>,

6 <A HREF="/mcc/softw/842/urt/Urt842Bsy.html">UrtBsy()</A>, <A HREF="/mcc/softw/842/urt/Urt842Get.html">get

-key()</A>

7 and <A HREF="/mcc/softw/842/urt/Urt842Put.html">putchar()</A>.

8 Development progress: <A HREF="/mcc/softw/834/urt/Urt834Df.html">Urt834.df</A>

9

10 \_getkey and putchar in the standard c51s.lib are replaced by new versions.

11 The enhancements compared to Keil are that T1, T2 or the fractional divider

12 can be initialised for baudrate generation. Also odd, even or no parity

13 can be chosen. No parity and two stop bits (mark parity) is also offered.

14 Lastly an error indicator is provided.

15

16 This demonstration receives characters and when a newline is received it

17 returns the string with slight modification. (Note that for some parity

18 errors the newline character is not recognised and 22 characters must be

19 entered.) This is repeated 4 times with no, even and odd parity and 2

20 stop bits. Each one can use any of the timers at various baudrates by

21 changing the parameters for UrtCfg() manually and recompiling.

22

23 Windows Hyperterminal can be used for the test with correct operation for

24 the correct parameters and failure for the others. Note though that the

25 difference between 1 and 2 stop bits is normally not detected.

26

27 Correct operation is indicated by "Hello " then the characters sent. then

28 the value of parameter1 of UrtCfg in "pseudo hex".

29 \*/

30 #include"..\kei842.h" //;<A HREF="/mcc/softw/842/Kei842Sfr.html">SFR definition file</A>.

31 #include"..\lib842.h" //;<A HREF="/mcc/softw/842/Lib842H.html">Function and variable declaration file</A>.

32 #include<stdio.h> //"stdio.h"

33 #include<ctype.h> //"ctype.h"

34

35 void main(void)

36 {

37 1 char pcHW[30] = {"Hello World"};

38 1 char c1;

39 1

40 1 PllWcd(0); //<A HREF="/~ehartman/824/pll/Pll824Wcd.html">PllWcd</A> writes CD bits.

41 1 UrtCfg(0x07,0x862d);

42 1 printf("\nEnter strings terminated by Enter.\n");

43 1 while(UrtBsy()!=2); //Ensure printf complete.

44 1 PllWcd(1);

45 1 UrtCfg(0x12,-54);

46 1 scanf("%22s",&pcHW[6]);

47 1 if(cUrtVar&0x80)

48 1 {

49 2 PllWcd(0);

50 2 while(!TI);

51 2 PllDly(5); //<A HREF="/~ehartman/834/pll/Pll834Dly.html">PllDly</A> to settle false start.

52 2 UrtCfg(0x06,-108);

53 2 puts("Parity off error");

54 2 }

55 1 c1 = 7;

56 1 while(c1<22 && (pcHW[c1]!=0)) c1++;

57 1 pcHW[c1++] = ' ';

58 1 pcHW[c1++] = ((cUrtVar>>4)&0x0f)+'0';

C51 COMPILER V5.50, URT842 17/10/03 17:49:26 PAGE 2

59 1 pcHW[c1++] = (cUrtVar&0x0f)+'0';

60 1 pcHW[c1++] = '\n';

61 1 pcHW[c1++] = 0;

62 1 puts(pcHW);

63 1

64 1 PllWcd(0);

65 1 UrtCfg(0x0e,-108);

66 1 scanf("%22s",&pcHW[6]);

67 1 if(UrtBsy()&0x80)

68 1 {

69 2 while(!TI);

70 2 PllDly(5); // False start settles.

71 2 PllWcd(1);

72 2 UrtCfg(0x16,-54);

73 2 puts("Parity even error");

74 2 }

75 1 c1 = 7;

76 1 while(c1<22 && (pcHW[c1]!=0)) c1++;

77 1 pcHW[c1++] = ' ';

78 1 pcHW[c1++] = ((cUrtVar>>4)&0x0f)+'0';

79 1 pcHW[c1++] = (cUrtVar&0x0f)+'0';

80 1 pcHW[c1++] = '\n';

81 1 pcHW[c1++] = 0;

82 1 puts(pcHW);

83 1

84 1 PllWcd(1);

85 1 UrtCfg(0x19,-54);

86 1 scanf("%22s",&pcHW[6]);

87 1 if(UrtBsy()&0x80)

88 1 {

89 2 while(!TI);

90 2 PllDly(5); // False start settles.

91 2 PllWcd(3);

92 2 UrtCfg(0x36,-13);

93 2 puts("Parity odd error");

94 2 }

95 1 c1 = 7;

96 1 while(c1<22 && (pcHW[c1]!=0)) c1++;

97 1 pcHW[c1++] = ' ';

98 1 pcHW[c1++] = ((cUrtVar>>4)&0x0f)+'0';

99 1 pcHW[c1++] = (cUrtVar&0x0f)+'0';

100 1 pcHW[c1++] = '\n';

101 1 pcHW[c1++] = 0;

102 1 puts(pcHW);

103 1

104 1 PllWcd(2);

105 1 UrtCfg(0x026,-27);

106 1 scanf("%22s",&pcHW[6]);

107 1 if(cUrtVar&0x80)

108 1 puts("2 Stop bit error");

109 1 c1 = 7;

110 1 while(c1<22 && (pcHW[c1]!=0)) c1++;

111 1 pcHW[c1++] = ' ';

112 1 pcHW[c1++] = ((cUrtVar>>4)&0x0f)+'0';

113 1 pcHW[c1++] = (cUrtVar&0x0f)+'0';

114 1 pcHW[c1++] = '\n';

115 1 pcHW[c1++] = 0;

116 1 puts(pcHW);

117 1 printf("\nEnd. Press reset to restart.\n");

118 1 while(1);

119 1 }

MODULE INFORMATION: STATIC OVERLAYABLE

CODE SIZE = 729 ----

CONSTANT SIZE = 172 ----

C51 COMPILER V5.50, URT842 17/10/03 17:49:26 PAGE 3

XDATA SIZE = ---- ----

PDATA SIZE = ---- ----

DATA SIZE = 2 31

IDATA SIZE = ---- ----

BIT SIZE = ---- ----

END OF MODULE INFORMATION.

C51 COMPILATION COMPLETE. 0 WARNING(S), 0 ERROR(S)