

UART_project_design_with_LCD - ISIS Professional (Animating)

File View Edit Tools Design Graph Source Debug Library Template System Help

Virtual Terminal - RECEIVE FROM USART TERMINAL

Virtual Terminal - TRANSMIT TO USART TERMINAL

INSTRUMENTS

- OSCILLOSCOPE
- LOGIC ANALYSER
- COUNTER TIMER
- VIRTUAL TERMINAL
- SPI DEBUGGER
- I2C DEBUGGER
- SIGNAL GENERATOR
- PATTERN GENERATOR

Digital Oscilloscope

Trigger

Channel A Channel C

Channel B Channel D

Horizontal

U1

1	REXWLR/VPP	RC0T10S0TICM	= 15
2	RADAN1DU1RNUC12IND	RC1T10S0C P2	= 17
3	RA1AN1C12N1-	RC2P1ACC P1	= 18
4	RA2AN2VREF-CVREFC2IN+	RC3SC1KSL	= 23
5	RA3AN3VREF-AC1IN+	RC4SD1S0A	= 24
6	RA4TDC1AC1OUT	RC5SD0	= 25
7	RA5AN4SSC2OUT	RC6TCK	= 26
8	RA6OSC2C1LHOUT	RC7RVDT	= 26
9	RA7OSC1C1LHIN		
10		RD0	= 19
11	RD1AN12INT	RD1	= 20
12	RD2AN12C12IN3-	RD2	= 21
13	RD3AN3	RD3	= 22
14	RD4AN3PG1C12IN2-	RD4	= 27
15	RD5AN11	RD5P18	= 28
16	RD6AN12TG	RD6P1C	= 29
17	RD7OSC1LH	RD7P1D	= 30
18	RD8ACSPDAT	RD8AN5	= 8
19		RD9AN6	= 9
20		RD10AN7	= 10

PC16F887

TRANSMIT TO USART T

RECEIVE FROM U

7 Message(s) ANIMATING: 00:00:30.539583 (CPU load 13%)

Data entering
PIC

Data read by
PIC

```
program UART_project
```

```
dim uart_rd as byte
```

```
dim LCD_RS as sbit at RB4_bit
```

```
dim LCD_EN as sbit at RB5_bit
```

```
dim LCD_D7 as sbit at RB3_bit
```

```
dim LCD_D6 as sbit at RB2_bit
```

```
dim LCD_D5 as sbit at RB1_bit
```

```
dim LCD_D4 as sbit at RB0_bit
```

```
dim LCD_RS_Direction as sbit at TRISB4_bit
```

```
dim LCD_EN_Direction as sbit at TRISB5_bit
```

```
dim LCD_D7_Direction as sbit at TRISB3_bit
```

```
dim LCD_D6_Direction as sbit at TRISB2_bit
```

```
dim LCD_D5_Direction as sbit at TRISB1_bit
```

```
dim LCD_D4_Direction as sbit at TRISB0_bit
```

```
dim output as char[1]
```

```
main:
```

```
TRISB = 0
```

```
PORTB = %11111111
```

```
ANSEL = 0
```

```
ANSELH = 0
```

```
Lcd_init()
```

```
Lcd_Cmd(_LCD_CLEAR)
```

```
UART1_Init(9600)          ' Initialize UART module at 9600 bps
```

```
Delay_ms(100)            ' Wait for UART module to stabilize
```

```
while (TRUE)             ' Endless loop
```

```
  if (UART1_Data_Ready() <> 0) then ' If data is received,
```

```
    uart_rd = UART1_Read()          ' read the received data,
```

```
    UART1_Write(uart_rd)
```

```
    ByteToStr(uart_rd,output)
```

```
    Lcd_Out_Cp(output)              ' and send data via UART
```

```
  end if
```

```
wend
```

```
end.
```