

Brief Device Description

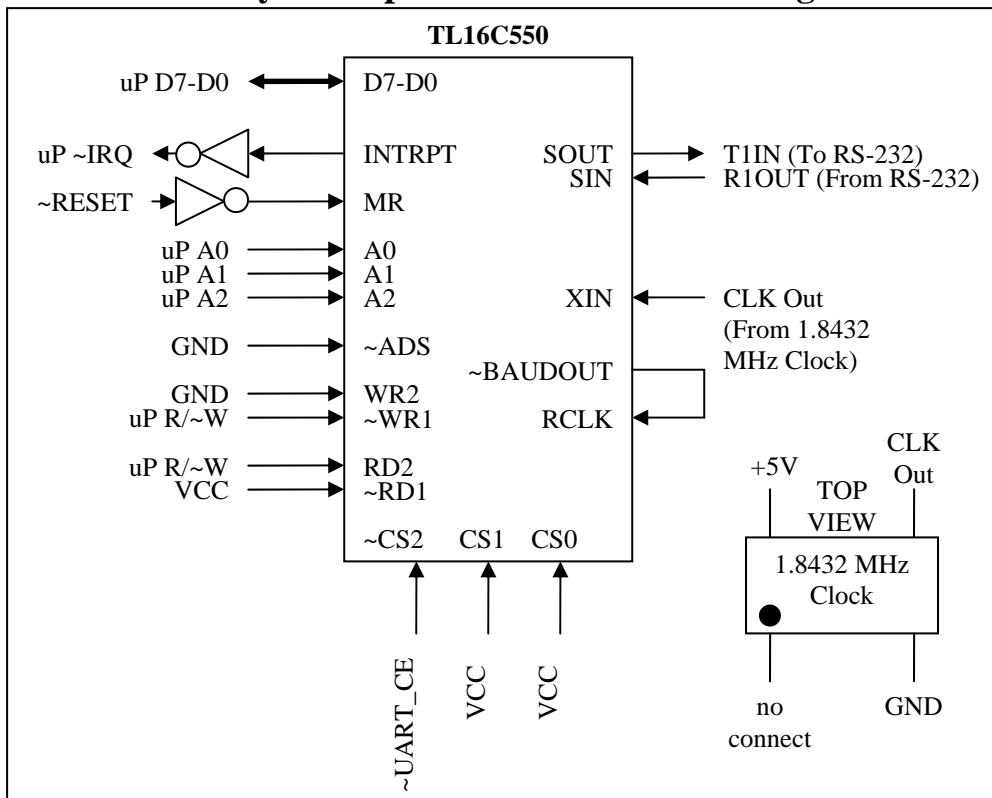
The TL16C550 is an Asynchronous Communications IC that supports one channel of full duplex RS232 communication. It features double buffering of both transmit and receive characters. It also offers programmable data lengths (5-8 bits), even/odd/no parity, 1/2 bit stop bit generation and a wide variety of programmable baud rates. See the datasheet available at Texas Instrument's web site (www.ti.com) for more information and the data sheet from TI at <http://www-s.ti.com/sc/ds/tl16c550c.pdf> (also available on our website).

Pin Definitions

See the suggested wiring diagram on the following page.

Pin Name	Description
A2:0	Address line inputs for selecting internal registers
~ADS	Address strobe input that can be used to internally latch a dynamic address bus value. Note: If your μ P supports static address, this input can be simply grounded.
D7:0	Bi-directional TTL level data bus used to R/W from internal registers.
~BAUDOUT	Output that is system clock divided by 16 and then divided by BAUD register contents.
CS0, CS1, ~CS2	Chip select inputs, where all must be true to allow reads/writes to internal registers.
~TXREADY, ~RXREADY	Transmitter/Receive DMA signaling ready.
~CTS, ~DCD, ~DSR, ~DTR, ~RI, ~RTS, ~OUT1, ~OUT2	Modem communication pins not required in basic RS232 communication.
DDIS	Output, that is high when the UART is not being read. Can be used to control external transceiver.
RD2, ~RD1	Data input strobes used to read data from the UART. Attach one to the read signal of your uP and then always DISABLE the other.
WR2, ~WR1	Data output strobes used to write data to the UART. Attach one to the ~write signal of uP and then always DISABLE the other.
INTRPT	Interrupt pin (high true) that can be used to initiate an interrupt on the uP.
MR	Master reset input that resets the UART when high.
RCLK	Receiver clock input that is tied to -BAUDOUT in most cases.
SIN	Serial input (RX line)
SOUT	Serial output (TX line)
Vcc, Vss	+5V, Ground
XIN, XOUT	System clock or crystal inputs.

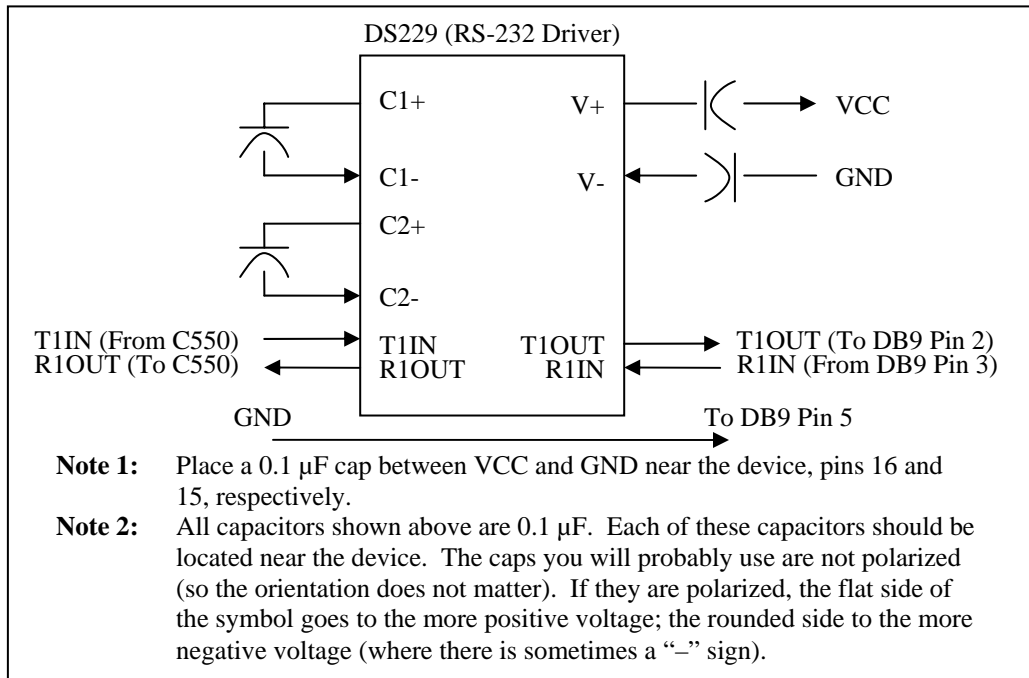
Suggested Elementary Microprocessor Interface Wiring



Notes:

1. See the UART spec sheet for pinout of this device.
2. Connect a 0.1 μF capacitor between power and ground pins (not shown above). (VCC is pin 40 and GND (VSS) is pin 20.)
3. All other pins not shown above are left open (i.e., no connection).
4. ~UART_CE is a decoded signal in the microprocessor's memory map. You will generate this signal using your CPLD.
5. ~Reset is the microcontroller system's active-low reset.
6. All inverters shown in the schematic can be placed in the CPLD used for address decoding.

RS-232 Line Driver/Receiver Interface



Be careful! Incorrect wiring of the RS-232 driver may destroy your serial port.

Typical Steps in Initializing the 16C550 (or 16C450) UART

UART BAUD Rate Initialization

1. Set D7 in the Line Control Register (LCR) high. This is the Divisor Latch Access Bit (DLAB) and when set high it allows us to write/read the Divisor Latches (BAUD rate register) at addresses 0 and 1 in the UART.
2. Write the upper divisor value to the upper divisor latch at base address + 1. Where base address is the lowest address that the UART has been mapped to in the microprocessor's memory map.
3. Write the lower divisor value to the lower divisor latch at base address + 0.
4. Disable the Divisor Latch Access Bit in the Line Control Register by setting D7 in the LCR low. This now allows us to write/read to the Receive & Transmit buffers (RBR & THR) at base address +0 and Interrupt Enable Register (IER) at base address +1.

Note: With a 1.8432 MHz system clock, 9600 BAUD has an upper divisor = 0 and lower divisor = 12 decimal.

Data Length & Stop Bit Initialization

5. Set D1 & D0 to the appropriate values in the Line Control Register (LCR). Note: For data word lengths = 8 bits, D1 and D0 in LCR are both set high. See data page 28 for additional word length bit field definitions if necessary.
6. Set D2 in the LCR low if only one stop bit is desired, else set high if 2 stop bits are desired.

Enabling a TX or RX Interrupt

7. Set D0 high in the Interrupt Enable Register (IER) if you desire to generate an interrupt when new characters are received.
8. Set D1 high in the IER if you desire to generate an interrupt when the transmit buffer is empty.

Suggested Early Functional Testing

1. Verify that you can write and read to registers in the UART without error.
2. Try sending out characters by polling the Transmitter Holding Register (THRE) bit and Transmitter Empty (TEMT) bit in the Line Status Register (LSR). If THRE is high, you can write a new character to the transmit buffer.
3. Try reading in a character by sending characters from hyper terminal and writing code to poll the Data Ready (DR) bit in the Line Status Register (LSR). If Data Ready is high, a character can be read from the receive buffer.
4. Once polling works, move on to generating an interrupt with the UART when a character has been received. Your ISR should read the Interrupt Identification Register (IIR) and check bits D3:0. You should see '0100' on these bits indicating a character has been received. Read the character and transmit using polling.
5. Enable both RX and TX interrupts in the UART. Modify your interrupt handler to read/test D3:0 in the IIR such that if 0'100' is detected jump to code to read the character in and if '0010' is detected then write a character out. See "Table 5. Interrupt Control Functions" in the 16C550 data sheet for more details on these bit field definitions.