How to do RS232/UART with CTS/RTS?

system **Dec '13** **post #1**

I want to connect my RS232 device (modem or even something else) to Arduino Mega 2560 R3. The device uses a standard serial communication with CTS/RTS handshake (or whatever it's properly called).

I plan to purchase RS232 - TTL converter, which contains all the required signals (see figure).

AVR / Arduino Serial class uses TX and RX pins (so it is USART only) and I'm interested in how to "add" CTS/RTS handshaking (wiring and programming).

Serielle Kommunikation mit Ofen scheitert (RS232)

system **Dec '13** **post #2**

Depending on the exact application, you may not even have to write a single piece of code and can just connect RTS to CTS to fool the device into thinking that your Arduino is always ready to accept new data.
The problem you may have is that if the device does not leave the minimum required inter byte delay, you may get some errors.
Also in some obscure devices, you may have to invert either CTS or RTS before connecting them together.
Worth a try.

system **Dec '13** **post #3**

I do not want to fool the devices. I wand to
:LOOP
receive some batch of data, to disable receiving, processing data, to enable receiving
GOTO :LOOP

robtillaart **Dec '13** **post #4**

The Arduino needs to use 2 extra pins one in and one out

The code below is not engineered but should give you the idea

```
uint8_t CTS = 4;  // CLEAR TO SEND
                  / REQUEST TO SEND
```

Skip to main content

```
....
if (digitalRead(RTS) == HIGH) // the PC want to send data
{
   if (buffer is empty)
   {
      digitalWrite(CTS, HIGH);  // allow sending of data
      while (Serial.available() == 0) ;  // wait for data to come in.
      digitalWrite(CTS, LOW);  // signal to stop sending data
      process data from serial
    }
}
...
```

a variation sometimes seen, (depends on the sending hardware

```
uint8_t CTS = 4;  // CLEAR TO SEND
uint8_t RTS = 5;  // REQUEST TO SEND

....
if (digitalRead(RTS) == HIGH) // the PC want to send data
{
   if (buffer is empty)
   {
      // give a pulse to signal send allow
      digitalWrite(CTS, HIGH);
      delay(n);  // n typical 1 or 2 milliseconds.
      digitalWrite(CTS, LOW);
      while (digitalRead(RTS) == HIGH); // wait for LOW. ==> CTS is recieved.
      while (Serial.available() == 0) ;  // wait for data to come in.
      x = Serial.read();
      ...
    }
}
...
```

should give you the idea.

more about CTS/RTS

- **RS232 Voltage Levels - Signals DTR CTS RTS » Electronics Notes** -

To make RTS/CTS really work it should be embedded in the hardware serial core code.

---

system **Dec '13 post #5**

> K5CZ:
> I do not want to fool the devices. I wand to
> :LOOP
> receive some batch of data, to disable receiving, processing data, to enable receiving
> GOTO :LOOP

OK, does this help?

RTS/CTS handshaking and waveforms

I decided to learn about RS-232 signalling and handshaking. I had a heck of a time finding a single page which summed up everything I would...

---

closed on May 6, '21

Skip to main content

Back to top