

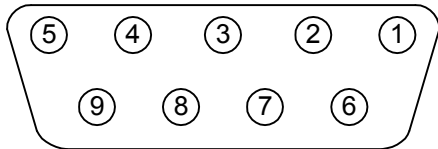
# Figuring out RS-232/DB9 connectors

RS-232 can be a bit intimidating at first. However, once you understand the basics, you'll find that it's really pretty straightforward. There are even some good reasons for the way that some aspects of it ended up (of course, there are other aspects that don't seem to make any sense at all and must have involved heavy drinking).

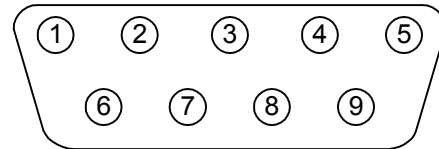
You've probably heard the terms "DCE" and "DTE". I won't delve into all the details of the nomenclature since it probably doesn't apply to your PDK implementation anyway and it just gets confusing. Here's what you need to know:

DCE (data communications equipment)- transmits on pin 2; receives on pin 3  
DTE (data terminal equipment)- transmits on pin 3; receives on pin 2

First, decide whether you want a male or female DB-9 on your PDK. If you're going to connect to a PC serial port, you'll probably need a female DB-9 on your PDK since most PCs had a male DB-9. If you're connecting to your own equipment you can select either. Here are the most common pin outs for the DB-9s but don't be surprised if you find something different!



Female pinout



Male pinout

Pin number	Function
1	Rx signal detect
2	Transmitted data
3	Received data
4	Data terminal ready
5	ground
6	Data set ready
7	Request to send
8	Clear to send
9	Ring indicate

Pin number	Function
1	Rx signal detect
2	Received data
3	Transmitted data
4	Data terminal ready
5	ground
6	Data set ready
7	Request to send
8	Clear to send
9	Ring indicate

Both of the above drawings are shown looking at the mating end of the connector. (I hope I don't have to explain the male/female designations!) I won't bother addressing most of the pins since you probably don't need them. If you do need them, then you probably already know about them and don't need to read this.

## Essentials:

Here are the ones you really need:

Ground: better have this connected if you want any chance of anything working.

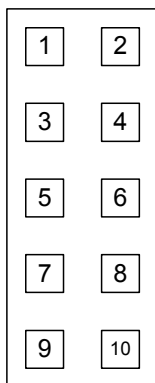
Received data: if you're trying to get data *into* your PDK you'll need this one.

Transmitted data: if you want data from the PDK to get somewhere else, you'll need to connect this pin.

Note: I've often connected the TX data line just for debugging purposes. It's easy to setup a printf statement in your code to output status or error messages, etc to a terminal.

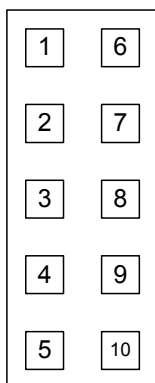
## The Other End

Now things take a twist. While all DB-9s have the same pin out, the other end of the cable offers two choices. There are two common ways to number it:



The "standard" way to number a header on a PCB is shown to the left. Note that this is the view of the male header on the pcb and not the connector on the end of the DB-9 cable (it's a mirror image).

This works fine for cables build from discrete wire but causes problems if you're using ribbon cables. Why? Manufacturers using ribbon cables want to use IDC (Insulation Displacement Connectors) that can be easily crimped on in one step.



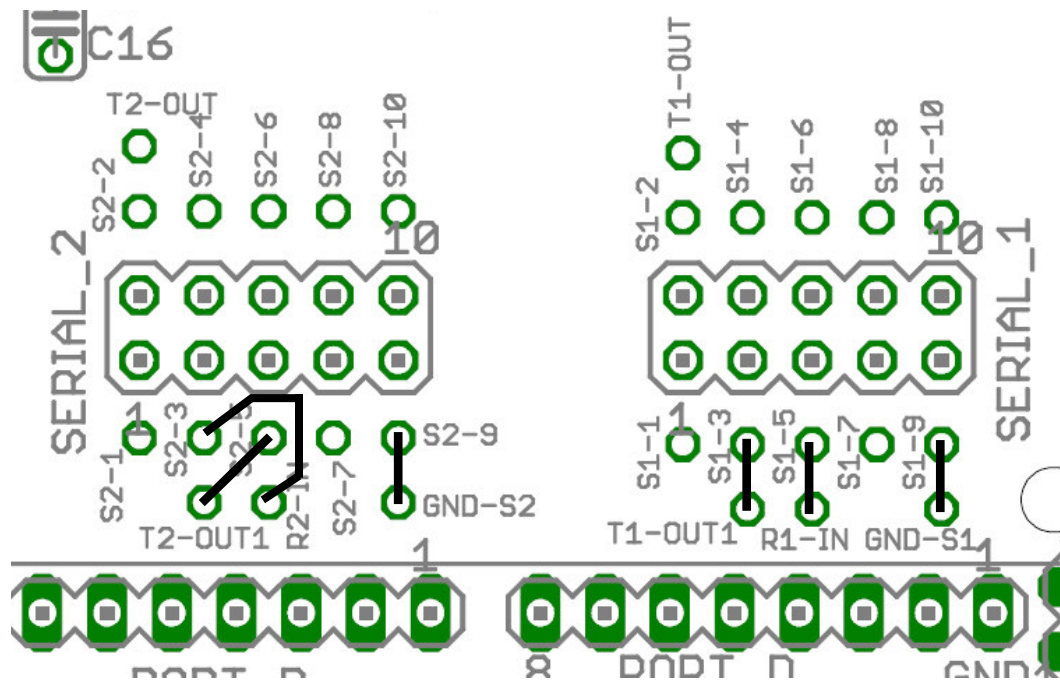
To use IDC connectors on each end you need to adopt this pin out. Note also that pin 10 isn't used since the DB-9 is a DB-9 (not a DB-10).

**If you ordered a DB-9 ribbon cable (male or female) with your PDK, it will use this pin out!**

We didn't connect any of the signals to this header on your PDK since we didn't know which connector you were going to use. You'll have to run a jumper from the TX or RX side to the appropriate pins based on which DB-9 cable assembly you use. **Don't forget to hook up ground!**

## Making it work.

Enough theory; here's how to make it work. Below is the layout from the PDK-I showing the Serial port connectors. Note, our CAD package automatically labeled the pin headers using the industry standard; not the IDC version (we'll fix that in the next spin).



### Example explained:

First, both ports must be using an IDC cable, since S2-9 is grounded on each port (thus being treated as pin 5).

Serial port 1 is transmitting on DB-9 pin 2; receiving on DB-9 pin 3. Probably a female connector.

Serial port 2 is transmitting on DB-9 pin 3; receiving on DB-9 pin 2. Probably a male connector.

### Next:

Just because you've got it wired up right doesn't mean it will work! You still need to select your bit rate; parity, stop bits and number of data bits. These all vary based on what you're connecting to. We've found that 8 data bits, no parity and 1 stop bit is pretty common. Also, 57,600bps is fast enough for most applications.

You'll also need some kind of handshaking. Most systems use a software based flow control. However, you can connect unused I/O from the Atmega128 and implement your own hardware based flow control in software.