

#### **UART Protocol**

Electronics > Communication Protocols > UART Protocol

# Contents

- 1. Overview
- 2. Terminology
- 3. Protocol
- 4. Transmission Speeds
- 5. Flow Control

#### 1. RTS/CTS

- 6. Error Checking/Noise Immunity
- 7. Break Signal
- 8. Higher-Level Protocols
- 9. <u>Terminal Programs</u>
  - 1. <u>RealTerm (3.5/5)</u>
  - 2. <u>Terminal by Br@y (3.5/5)</u>
  - 3. <u>PuTTy (4/5)</u>
- 10. 9-Bit Addressing
- 11. Radiation Hardening
- 12. <u>RS-232</u>
- 13. RS-485
- 14. Cables
- 15. Powerline Transceivers
- 16. Creating A Serial Port Bridge

#### **Overview**

UART (*Universal Asynchronous Receiver/Transmitter*) is a digital data transmission protocol with origins dating back to the 1960's. It was designed as a communication protocol to talk between *DTE* (data terminal equipment) and *DCE* (data communication equipment). It is universal in the sense the timing, voltages, flow control and error checking can be configured.

Drive Type	Single-ended
Num. Wires (excl. GND)	2 (TX/RX) or 4 (TX/RX and RTS/CTS)
Duplexity	Full
Connection Topology	Point-to-point
OSI Layers	Layers 1 (physical) and 2 (data link)

It is commonly used today as a simple, two-way node-to-node serial communications protocol between devices on a circuit board or possibly over a cable. Because of it's low voltage and single-ended nature, it is not very noise resilient, and is usually replaced with a more robust protocol such as <u>RS-232</u> or <u>R</u>:

 $\underline{485}$  when communication occurs over any significant cable length or in a noisy environment.

# Terminology

Sorted in alphabetical order.

Term	Description
CTS	CTS is an initialism for " <b>Clear To Send</b> ". See the <u>Flow Control</u> <u>section</u> for more information.
DCD	DCD is an initialism for " <b>Data Carrier Detect</b> ". See the <u>Flow Control</u> <u>section</u> for more information.
DCE	DTE is an initialism for " <b>Data Communication Equipment</b> ". It was a term created when UART was first developed to describe electronic devices which transmitted/received data and connected to terminals (which were termed DTE's). See the <u>Flow Control section</u> for more information.
DTE	DTE is an initialism for " <b>Data Terminal Equipment</b> ". It was a term created when UART was first developed to describe electronic devices which displayed data and connected to modems (which were termed DCE's). See the <u>Flow Control section</u> for more information.
DTR	DTR is an initialism for " <b>Data Terminal Ready</b> ". See the <u>Flow</u> <u>Control section</u> for more information.
RI	RI is an initialism for " <b>Ring Indicator</b> ". See the <u>Flow Control section</u> for more information.
RTS	RTS is an initialism for " <b>Request To Send</b> ". See the <u>Flow Control</u> <u>section</u> for more information.
RxD	RxD is an acronym for " <b>Receive Data</b> ". See the <u>Flow Control</u> <u>section</u> for more information.
TxD	TxD is an acronym for " <b>Transmit Data</b> ". See the <u>Flow Control</u> <u>section</u> for more information.
UART	UART is an initialism for " <b>Universal Asynchronous</b> <b>Receiver/Transmitter</b> ".
USART	USART is an initialism for " <b>Universal Asynchronous</b> / <b>Synchronous</b>

	Receiver/Transmitter". ATMEL uses this term to describe the
	peripherals on its ATmega range of microcontrollers that support the
	standard asynchronous protocol as well as a synchronous (clocked)
	protocol.
	Voting describes a error-checking process in which the same bit of
Voting	UART data is sampled multiple times and then a vote occurs to
	determine it's state.

# Protocol

The long history of UART has resulted in many physical and protocol layer variations such as:

- CMOS UART
- RS-232
- RS-423 (differential)
- RS-485 (differential)
- DMX512 (commonly used to control stage lighting and effects)
- MIDI
- LIN Bus
- IrDa

### **Transmission Speeds**

UART, by today's standards, is a slow transmission protocol. However, it is still fast enough for tons of applications. The commonly supported baud rate speeds are:

600, 1200, 3400, 4800, 9600, 14400, 19200, 28800, 38400, 56000, 57600, 115200, 128000, 256000, 460800, 921600

I have had basically no issues using speeds up to 460800 baud in embedded systems (either talking to another embedded system, or to a computer through a virtual COM port). However, 921600 baud has not worked for me in some situations.

Some devices also support custom baud rates.

# **Flow Control**

The flow control is a way of detecting when the receiver or transmitter is ready to accept or send new data. The UART protocol provides a few (all optional) methods for flow control:

- Hardware RTS/CTS lines
- Software XON/XOFF flow control

#### **RTS/CTS**

This is done by various additional connections to the standard transmit, receive and ground wires. The two most common are CTS and RTS. Typically, a micro-controller may have one or two UART peripherals that support this feature, while the rest are just basic non-flow control UART's.

A small amount of power can be extracted from the RTS and CTS lines for powering low-power devices.

The following tables lists all of the flow control signals (as well as the data signals), with respect to the device in question. Matching signals are grouped together.

Signal	Port Type	Description	DB- 25 Pin	DE-9 Pin
DTR (Data Terminal Ready)	Output	DTE drives this to indicate to the DCE that it is present.	5	4
DCD (Data Carrier Detect)	Input	DCE drives this when it is connected to the telephone line.	8	1
RI (Ring Indicator)	Input	The DCE drives this when it has detected a phone call. Note that there is no matching signal going the other way.	22	9
RTS (Request To Send)	Output	DTE drives this to tell the DCE to get ready to receive data.	4	7
CTS (Clear To Send)	Input	Driven by the DCE when it is ready to accept data.	5	o
TxD (Data	Output	The DTE sends data to the DCE over this line.	2	3

http://blog.mbedded.ninja/electronics/communication-protocols/uart-protocol

Transmit)				
RxD (Data Receive)	Input	The DCE sends data to the DTE over this line.	3	2
Common Ground (GND)	n/a	The common ground for all signals.	7	5
Protective Ground (PG)	n/a	The protective ground. This is usually just connected up to the common ground on the PCB.	1	n/a

Note that confusion of how to connect two UART devices together arises when it is not terminal equipment (DTE) connected to modem equipment (DCE). In the embedded world, microcontrollers and other devices which support UART can act either as a DTE or a DCE. Take particular care when connecting UART ports together on embedded devices!

Rule-of-thumb: Most RS-232 serial interfaces with a male 9 or 25 pin connector are DTE's, most with a female 9 or 25 pin connector are DCE's.

Most often, manufacturers label the UART pins as DTE's. In this case, you have to swap all connections with their matching line. So TxD of device 1 is connected to RxD of device 2, RxD of device 1 is connected to TxD of device 2, RTS of device 1 is connected to CTS of device 2, e.tc.

### **Error Checking/Noise Immunity**

The only error checking a UART has by specification is parity checking (additional error features may exist).

You may notice when sending lots of characters across a UART that some appear to be corrupted. These can be a real bummer if you are using UART to transmit lots of data (for say a data logging application). The best ways to improve noise immunity are:

- Slow down the transmission rate to the slowest acceptable speed. Far less errors occurs at 9600 baud than say, 57600 baud.
- Enable parity checking (does not completely fix the problem!)
- Enable voting algorithms if the transmitter or receiver support it.

- Similarly, enable oversampling if the transmitter or receiver support it (very similar to voting).
- Make the UART transmission lines as short as possible and with as little capacitance as possible.
- Shield the UART cable (not so important)
- Implement a checksum algorithm into the receiver and transmitter, such as a CRC. The UART protocol does not support this natively, you will have to use a 3rd party library/write the code to do this yourself. Even when using a simple checksum algorithm such as exclusive or (XOR), this is probably one of the most fool proof methods for error checking.

#### **Break Signal**

The *break signal* is not a character, but a **special signal which can be sent from transmitter to receiver to indicate an event**.

The transmitter sends a break signal by **driving it's TX line low for a period longer than one frame**. There are two types of breaks, short breaks and long breaks. A short break is when the TX line is driven low for a period of between 1 and 2 frame lengths, and a long break is when it is driven low for a period of more than 2 frame lengths.

#### **Higher-Level Protocols**

Do you need a higher-level communication protocol that works over a UART connection? See the <u>SerialFiller</u> library on GitHub (written in C++). SerialFiller uses a publish/subscribe mechanism and works well on point-to-point serial connections such as UART.

#### **Terminal Programs**

#### **RealTerm (3.5/5)**

Website: http://realterm.sourceforge.net/

A easy to use and powerful terminal program for Windows. Stolen from the website, it's description is:

UART | mbedded.ninja

a terminal program specially designed for capturing, controlling and debugging binary and other difficult data streams. It is far better for debugging comms than Hyperterminal. It has no support for dialing modems, BBS etc – that is what hyperterminal does.

It can view and send binary, hex, ASCII, ANSI, integers (both signed and unsigned, 8 or 16-bit), floats and more. Support for half-duplex communication as well as I2C! Does not lag/hang at all (including when you disable the COM port while it is still running). You can run multiple RealTerm apps at the same time, to get data from multiple UART ports simultaneously. It can add timestamps to received UART messages, which is useful for data logging.

minute       production		Capitile Auguan (2005)						x
absolution       image: constraints in a constraint in	UT -0808083221 UT -0808083269	, 11-0000001120, U2-0 , 11-0000004064, U2-0	3263036329. 12-080808086348.	03	13=-000000568	04-0808012742. 04-0808012741	14-888888843 14-8888881169	
High Part       Control	JI 0808083268	. 11-0000001578. 02-0	3808030328, 12-08080808031,	U3 808080812. U3 808080812.	1.3	04-0808012741	14-0208080111	
Number Of Processor 1       1	11-00000003274	. 11-0000002712. U2-0	12-02020327.12-020202022.	03	13-808080696.	04-0808012742	14-0808080558	
NUMPORT       Control Type       UP       Section Type       Section Type       UP </td <td>1-0808083268</td> <td>. 11-0808081106, 02-0</td> <td>3808080320, 12-0808080819,</td> <td>03=-808080812.</td> <td>13-808080898</td> <td>04-0202012742</td> <td>14 - 2000200043</td> <td></td>	1-0808083268	. 11-0808081106, 02-0	3808080320, 12-0808080819,	03=-808080812.	13-808080898	04-0202012742	14 - 2000200043	
NUMPORT       Control Type       UP       Section Type       Section Type       UP </td <td>1~0808083268</td> <td>11-0000001419, U2-0 11-0000003751, U2-0</td> <td>3808088319, 12-0808080814, 3808080289, 12-0808080839</td> <td>U3-808080812 U3-808080812</td> <td></td> <td>04-0808012742 04-0808012742</td> <td>14-0808080896 14-0808081805</td> <td></td>	1~0808083268	11-0000001419, U2-0 11-0000003751, U2-0	3808088319, 12-0808080814, 3808080289, 12-0808080839	U3-808080812 U3-808080812		04-0808012742 04-0808012742	14-0808080896 14-0808081805	
Description 2.22       1       Description 2.22	1-0808083269	. 11-0000001107. U2-0	ARRING 280 12 - DREAD 00444	U3=-808080812.	13-808080737.	14-00000012242	14	
Description 2.22       1       Description 2.22	1-0808083269 1-0808083278	11-0000001075, 02-0	3808080275, 12-0808080842,	U3-808080812	13	04-0808012743	14-8808081137	
Bit Peri Cantas Peri Sand Edu Par DC 1002 102 20000000000000000000000000000			3202020271 12 02020000338	03=-808080811.	13-808080716.	04-0808012743	14-202020263	
Bit Decision 17:27, 1       1       Bit Decision 17:27, 1       1       Bit Decision 17:27, 1       -Bit Dec	1-0808083268	11-0000002734. U2-0	<b>12-080808263</b> . 12-0808080832.	V3-808080812.	13-808080586		14-0808080505	
Nonscription       1	1-0808083259	, 11-0000001100, U2-0	38038380263, 12-0808030328, 3803888263, 12-0808088822,	V3=-808080812,	13 -808088781,	U4-0808012244	14	
Nonscription       1	1-0808083272	11-0000001447, U2-0 11-0000003645, U2-0	3808088262, 12-0808088819, 3908080256, 12-0808080829	U3 808080812. U3 808080812	13-808080758,	U4-0808012744 U4-0808012742	14-0808080899 14-0808080931	
Nonscription       1	1 0808083267	, 11-0000001095, U2-0	3202020255, 12-0202020218,		13			
	1-0808983261 1-0808083271		3803030255 12-0803000011	U3 808080812	13-808088533,	U4-0808012742	14-0808081106	
Production 2, r. 1,	1 0808083268		3008080223, 12-080808080335, 3008080216, 12-08080808339,	U3 808080812, U3 808080812	1.3	04-0808012743. 04-0808012744	14=-000080133 14=-000080126	
Province 1277       11       Province 1277       12       Province 1277       14       Province 12777       14	1-0808083264 1-0808083266	I1-0000002766. U2-0	3808080212. 12-0808080844.	U3 808080812	13-808080647.	04-0808012743	14-0808080575	
Personalization       1	1-00000003267	, 11-0303031085, U2-0	32020209, 12-020202034,	13	13 -808080729	11.4 - 040 040 041 0 11.4 0	1	
Reference       11       -00000001170       02       -000000001260       13       -00000001274       14	1-0808083267 1-0808083278	. 11-0808081471, U2-0 . 11-0808083742, U2-0	1808080206. 12-0808080834.	03-808080812. 03-808080812.	13 808080511	04-0808012745. 04-0808012743.	14-8888888987	
Reference       11       -00000001170       02       -000000001260       13       -00000001274       14	1-0202003278	. 11-0000001840, U2-0	3202030198, 12=0202030331,	03=-202020212	13 202020703.			
Bibliotoci 226 1 1 Bibliotoci 126 0 2 Bibliotoci 126 12 Bibliotoci 12 13 Bibliotoci 12 13 Bibliotoci 1274 14 Bibli	1-0808083267 1-0808883271	11-0000004113, U2-0	3808080198, 12-08080808227,	V3-808080812.	13 808080533.	U4-0808012745	14-0000001122	
Internet 27.42, 11 = Mathematic 27.43, 12 = Mathematic 27, 13 = Mathematic 27, 14 = Mathematic 27, 13 = Mathematic 27, 13 = Mathematic 27, 14 = Mathematic 27, 13 = Mathematic 27, 13 = Mathematic 27, 14 = Mathematic 27, 13 = Mathematic 27, 13 = Mathematic 27, 14 = Mathematic 27, 13 = Mathematic 27, 14 = Mathematic 27, 13 = Mathematic 27, 14 = Mathematic 27, 13 = Mathematic 27, 14 = Mathamatic 27, 14 = Mathematic 27, 14 = Mathematic	1-0808083276 1-0808083268	. 11-0000001268, U2-0 11-0000001101 U2-0			13	U4-0202012744, U4-0202012246	14-0202030867 14-0202080899	
additional 27.00         11         additional 27.00         12         additional 27.00         additional 27	1-0000003271	11-0000002779 02-6	3808080194. 12-0808080828.	V3-808080812.	13-808080696.	04-0808012745	14-00000000586	
Home House 2:55       11       House 2:55       11<				113 - 20202030312	57 S000008-14	114 00000012244	T	
	1-0808083269	. I1-0808081485, 02-0 . I1-0808083633, 02-0	3208080190, 12-0808080810, 3208080189, 12-080808080810,	V3-808080812. V3-808080812.	13-808080768	14 HRHHH12744	14-00000000101	
Hereinseld 22 / 1       Hereinseld 15 / 2       0.3       Hereinseld 22 / 3       1.3       Hereinseld 22 / 4       1.4       Hereinseld 22 / 4       <	1=88888883266	. 11-0808081868. 02-0		03-202020312.		04-0202012746.	14	
Hereinseld 22 / 1       Hereinseld 15 / 2       0.3       Hereinseld 22 / 3       1.3       Hereinseld 22 / 4       1.4       Hereinseld 22 / 4       <	1-0808083268	. 11-0000004132. 02-0	3808080158, 12-0808080836,	V3-808080812.	13 000000564.	V4-0808012745	14-08080801219	
Televised227*, 11 - Babbiolo221, 12 - Babbiolo221, 13 - Boldbabb 73, U1 - Babbiolo221, 13 - Babbiolo222, 14 - Babbiolo222, 14 - Babbiolo222, 14 - Babbiolo222, 13 - Babbiolo222, 14 - Babbiolo22, 14 - Babbiolo2, 15	1-0000003273	, 11-0000001165, U2-0 11-0000001086, U2-0	3808080150, 12-0808080834, 3808080146, 12-08080808334,	0.3		04-0808012746	14-000000150 14-0000000129	
Barbardoursey       11       Barbardoursey       12       Barbardoursey       03       Barbardoursey       04       Barbardoursey       14       Barbardoursey       15       15       15       15 <td>1-0808083275 1-0808083269</td> <td></td> <td>4808080145 12-0808080838</td> <td>U3=-808080812.</td> <td>13-000000693.</td> <td>04-0000012746</td> <td>14-0808080577</td> <td></td>	1-0808083275 1-0808083269		4808080145 12-0808080838	U3=-808080812.	13-000000693.	04-0000012746	14-0808080577	
Barbardoursey       11       Barbardoursey       12       Barbardoursey       03       Barbardoursey       04       Barbardoursey       14       Barbardoursey       15       15       15       15 <td>1 030303272</td> <td>, 11-0303031090, 02-0</td> <td>3808080141, 12-080808080827,</td> <td>V3 R08080812,</td> <td>13 -88888889.</td> <td>04-0808012746</td> <td>14-208080892</td> <td></td>	1 030303272	, 11-0303031090, 02-0	3808080141, 12-080808080827,	V3 R08080812,	13 -88888889.	04-0808012746	14-208080892	
aplay Port Capture Pric Sand Echo Port U2C 12C-2 U2DMac Mac Status Acci Bylas 2 2 2 Acci Acci Bylas 2 2 2 Acci Bylas 2 2 2 Acci Brasy Acci	1-0000003270	. 11=0808081545. 02=0	3808080137, 12=08080808828,	03-808080812. 03-808080812.	13-808080795.	04-0808012747.		
Bit dig Ar       Image: Status       Image: Status       Image: Status       Image: Status         Area       Image: Status       Image: Status       Image: Status       Image: Status       Image: Status         Hear Acads       Image: Status								
Hed state Hes state und Hes state und Hes state Und Hes state Hes statee Hes statee Hes statee Hes statee Hes statee H	Display Ar	Hall Duplex newLine mode	#   12C   12C-2   12CMae   Mae				<u>)n Cle</u>	Status Disconnec
mbedded.nini	Hes(space) Hes + Asci ant8 Hes Hes M16 Asci Binary Binary	Privet Data 7 Do Lindon Data Frances Bytess 2 1 Signet Gulo 1 Rows Cale	n Scrollback					TOD (3) CTS (8) DCD (1) DSR (6) Ring (9) BREAK
				and Brown			mberlier	Inne

A screenshot of RealTerm in action.

I have noticed a few bugs with RealTerm, especially when it comes to changing the number of rows and columns, and scrolling back through received data (th scrollback variable is buggy also).

#### Terminal by Br@y (3.5/5)

#### Website: https://sites.google.com/site/terminalbpp/

A simple and tidy Windows terminal program. Personally, it doesn't get the same amount of respect as RealTerm because of it's simplicity and slightly buggy nature. When decoding into hex, the program can hang if your receiving large amounts of data. It can also hang if you disable the COM port while it is still connected.

-		
Disconnect BeScan Help About	CDM Port         Baudinate         Data bits         Party         Stop bits         Handshaking           CDM17         C         600         C14400         C 57600         C         Frome         C1         C none           CDM17         C         1200         C 115200         C         6         Codd         Frome         C1         C RTS/CTS           CDM6         C 2400         C 28800         C 128000         C         6         C odd         C         T S C/TS / MON/A0FF           C0M6         C 9800         C 26000         C         C mark         C 2         C RTS/CTS / MON/A0FF	
Quit Settings		
	T AutoDix/Connect T Time T Stream log <u>custom BR Rx Clear</u> <u>ASCIItable</u> <u>Scripting</u> T AutoStart Script T CR-LF T Stay on Top 9600 27 <b>‡</b> Graph Remote	CTS CD DSR CR
Receive		
CLEAR	J Recet Counter   13  Counter = 256 CHEX Conter = 256 CHEX CHEX StartLog StopLog Request/Response	
TIME: Cap mes	e is: 706s e is: 707s suurement event triggered suurement event triggered.	
CAP Cap dat CAP Data OK CAP Step Ch CAP Average TIME Time is <0>TIME Time <0>TIME Time TIME Batt met TIME Cap met CAP Starting	e is: 709s e is: 710s	1
CAP Cap data CAP Data OK CAP Step Ch CAP Average TIME Time 8 <0>TIME Time <0>TIME Time <0>TIME Time TIME Batt me TIME Cap mer	ia ready, going to read from cap sensor. K 25 25 C PF 00 00 00 81 01 13 26 C 00 80 00 65 46 59 EC hange State = State. 81 = 5468757. Average 2 = 5480009, Difference = 342, Trigger Percentage = 1% eta: T01s eta: T01s eta: T10s eta: T10s eta: T10s suorement event triggered.	- - -
CAP Cap dat CAP Data OK CAP Step Ch CAP Average TIME Time 8 <0>TIME Time 8 <0>TIME Time (0>TIME Time CAP ME Time TIME Batt me TIME Cap me CAP Starting Transmit	is ready, going to read from cap sensor to 25 stc PF do 00 00 80 10 11 30 20 00 00 04 54 59 EC hange State = Statek. # 1 = S488757, Average 2 = S480009, Difference = 342, Trigger Percentage = 1% e is 705a e is 7105 e is 7105	- DIR RIS
CAP. Cap dat CAP. Data OK CAP. Step Ch CAP. Average TIME. Time is «0>TIME. Time «0>TIME. Time TIME. Bat me TIME. Bat me CAP. Starting Transmit <u>CLEAR</u> Macros	is read, yoing to read from cap sensor. kto 25 kt F /f oo 00 00 10 11 H 30 CF 00 80 00 65 46 59 EC hange State = States 705s e is: 705s e is: 705s e is: 716s is: 716s is: 71fis is: 71fis	DDTR RTS

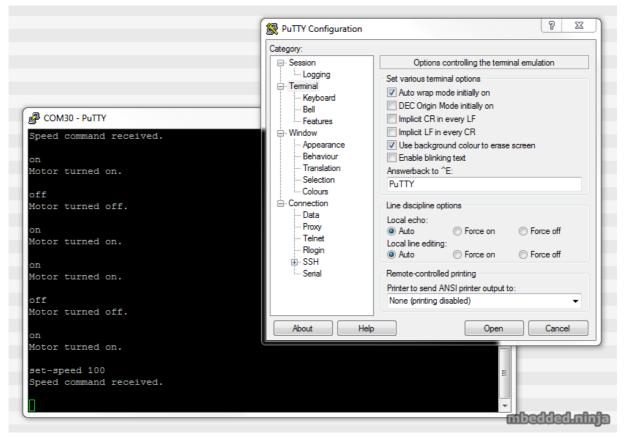
A screenshot of "Terminal by Br@y" in action.

#### PuTTy (4/5)

Website: http://www.chiark.greenend.org.uk/~sgtatham/putty/

PuTTY is a free implementation of Telnet and SSH for Windows and Unix platforms, along with an xterm terminal emulator.

If your running windows, PuTTY is a very handy application to have if you want to emulate the command-line style interface of a UNIX-like system. Although the debugging and capturing features are not as good as say, RealTerm, it offers character-by-character input and proper response to pressing 'special' keys such as enter (which RealTerm doesn't allow, instead you have to enter a string and then press send). This may sound like a very small difference, but this feature does com^ in useful! I find it very handy when using FreeRTOS and the CLI (command-lin interface) extension, which allows you to communicate from a pc to a embedded system using a command-line style interface (as in the picture to the right).



A screenshot of the PuTTy application in action, along with the settings window.

I have discovered one bug in PuTTY...if it receives a large number of characters all at once (which is common when printing debug messages from an embedded system, and for some reason, the string is not null-terminated, and starts printing gobble-degooch from random memory locations), PuTTY can freeze, and needs to be restarted. In this situation, it can also print the message "PuTTyPuTTyPuTTy" many times over across the COM port you are debugging. Weird.



PuTTY can freeze when printing a large number of random characters to it across a COM port.

## 9-Bit Addressing

9-bit addressing was employed when using a multi-drop configuration to prevent slaves from wasting processor time in decoding every byte on the bus to see if it was addressed to them. A 9th bit is sent out after every byte, and is used to signal if the previous 8-bits where an address (which the slaves have to listen to), or just data (which can be ignored).

### **Radiation Hardening**

Some UART protocols have radiation tolerant devices, such as the <u>DRS4485</u>, an Dual RS-485 Interface Transceiver made by Aeroflex.

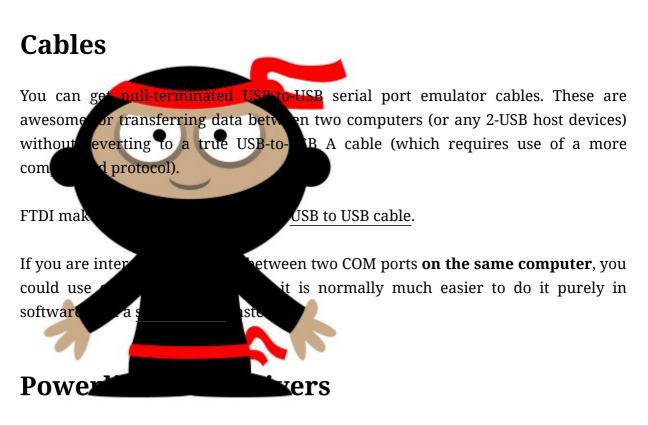
#### **RS-232**

RS-232 is a very similar protocol to UART, and a UART to RS-232 converter is one of the most popular communication protocol converters you will see in an embedded system.

For more information, see the <u>RS-232 page</u>.

#### **RS-485**

RS-482 is another very common protocol that UART is converted to and from. It is usually chosen over RS-232 when longer distances and/or larger noise immunity is needed. For more information, see the RS-485 page.



# The SIG60 is an example of a powerline transceiver.

The embedded engineering website that's got your back.

# **Creating A Serial Port Bridge**

There are occasions when you want or need to send serial data between two pieces of software on the computer, or between two hardware devices both connected to the computer. An example would be to unit test a PC-based serial communications protocol you have written without writing the unit-test code on the microcontroller. There are software programs that emulate a serial port bridge, but in my experience I found these are every buggy or cost money. UART | mbedded.ninja

	<b>-13</b>		al Capture Program	n 2.0.0.57	×
	Testing a phys www.cladlab.co	ical serial port bridge. መርቅም	CRLF		
	RealTerm: Seria	I Capture Program 2.0.0.57	-		
Display Port Capture F			Datara -	efore fter	ear Freeze Status Status Disconnec RXD (2) TXD (3) CTS (8) DCD (1) DSR (6) Ring (9) BREAK Error
0 ^C LF Repea	ts 1 🛨 🗆 Litera	mbers Send ASCII +CR +CR	US 8 -	TXD (3) CTS (8)	t: 11 57600 8N1 Non
Dump File to Port		a i oup opaces (i toto) i		DCD (1)   DSR (6)	
c:\temp\capture.txt	▼] S	end <u>File</u> X Stop Delays 0 🔹 <u>R</u> epeats 1 🔹 0		Ring (9)   BREAK   Error	
2		Char Count:8 CPS:	:0 Port: 4 57600		mbedded.ni

Testing a physical serial-port bridge, made by connecting two FTDI cable together.

You can create a rudimentary serial bridge to connect to pieces of software together by connecting two USB-to-UART (or USB-to-RS232) converters together, crossing the RX and TX lines over. Although not a very permanent solution, this is good for simple tests. The following image shows a hardware-based serial port bridge with a terminal on each end.

SHAR	E THIS:						
f	G+	in	t	P	y	8	More
LIKE	THIS:						
Loadi	ng						

```
Posted: September 12th, 2011 at 5:52 pm
Last Updated on: August 31st, 2017 at 8:04 am
```

Ο Recommend 🛛 🔁 Share	Sort by Best
Start the discussion	n
LOG IN WITH	OR SIGN UP WITH DISQUS ?
	Name
Be the	first to comment.
ALSO ON CLADLAB	
The Idle Task	NinjaCalc v1.3.0 Released
	NinjaCalc v1.3.0 Released 3 comments • a year ago
	3 comments • a year ago re ado — Done :-)
2 comments • 3 years ago <b>gbmhunter</b> — How are you sur about the real percentages bei	3 comments • a year ago re ado — Done :-)
<ul> <li>2 comments • 3 years ago</li> <li>gbmhunter — How are you sur about the real percentages bei 70% and 30%? Also, try creati</li> <li>Designators</li> </ul>	3 comments • a year ago re ado — Done :-) ing
<ul> <li>2 comments • 3 years ago</li> <li>gbmhunter — How are you sur about the real percentages bei 70% and 30%? Also, try creati</li> <li>Designators</li> </ul>	3 comments • a year ago re ing ing <b>TO-220AC Component Package</b> 2 comments • 3 years ago <b>gbmhunter</b> — Thanks for pointing that out, I have updated
<ul> <li>about the real percentages bei 70% and 30%? Also, try creati</li> <li>Designators</li> <li>2 comments • 3 years ago</li> <li>gbmhunter — Thanks for the info, I have encountered the</li> </ul>	3 comments • a year ago re ing ing <b>TO-220AC Component Package</b> 2 comments • 3 years ago <b>gbmhunter</b> — Thanks for pointing that out, I have updated

Proudly powered by WordPress