[\(/id/Fusion/\) Fusion 360 \(/id/Fusion/\)](#)

Connect the Raspberry Pi to network using UART by ludeks (/member/ludeks/)

[Download](#)[\(/id/Connect-the-Raspberry-Pi-to-network-using-UART/\)](#)

7 Steps

[+ Collection](#)[I Made it!](#)[♥ Favorite](#)[🗨 Share ▾](#)

About This Instructable

22,992 views **32** favorites**Posted:**
Dec 31, 2014**License:**
**ludeks**
[\(/member/ludeks/\)](#)[Follow](#)

1

[\(/member/ludeks/\)](#)

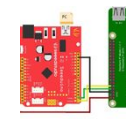
More by ludeks:

[\(/id/Connect-the-](#)[Raspberry-Pi-to-network-using-UART/\)](#)**Tags:** [rasberry pi \(/howto/rasberry%20pi/\)](#)[uart \(/howto/uart/\)](#)[serial \(/howto/serial/\)](#)[ppp \(/howto/ppp/\)](#)

Add instructable to:

[Group](#)

Related

**JTAGulating the Raspberry Pi 2**
[\(/id/JTAGulating-the-Raspberry-Pi-2/\)](#)by Tymkrs [\(/member/Tymkrs/\)](#)**RASPBERRY PI 3 (and RASPBIAN JESSIE) GSM 3G SERIAL HOW TO**
[\(/id/RASPBERRY-PI-3-and-RASPBIAN-JESSIE-GSM-3G-SERIAL-HOW-TO/\)](#)**Hello world to Raspberry Pi through Sseeeduino (Uart port)**
[\(/id/Hello-world-to-Raspberry-Pi-through-Sseeeduino-Uart-port/\)](#)**RFID Based Attendance System Using Raspberry Pi**
[\(/id/RFID-Based-Attendance-System-Using-Raspberry-Pi/\)](#)

In this tutorial I will show you how to connect your Raspberry Pi to the Internet just by using its serial line (http://elinux.org/RPi_Serial_Connection) which is accessible on GPIO (<http://pi.gadgetoid.com/pinout/uart>). This can be very helpful if there is no other means of connection available like ethernet or wifi dongle. The UART serial port is not directly suitable for TCP/IP traffic so we are going to use a PPP protocol that will encapsulate the serial link and give us an IP interface. Please note that the UART is designed to work in kilobits speed range (115.2kbit by default), that's sufficient for ssh and small files transfers, but it will take ages to transfer large files.

The operating principle is very simple, but it may be tricky to setup things properly, especially if your communication is limited to UART only. What you'll need:

- Raspberry PI (any model) with Raspbian
- USB <-> UART convertor + cable with 3.3v logic levels
- A linux computer that will act as a second serial link endpoint and a network gateway
- Optionally a monitor / keyboard / ethernet / wifi that you can plug to your Raspberry

This method is particularly suitable for the model A as it lacks an ethernet port, but it will work with any Raspberry model. I have the model B rev1.0 :-)

The convertor can be of any type, I'm using a cheap convertor with a CP2102 chip. As a second endpoint, I'm using my Ubuntu laptop. You don't need anything

else to plug in to your Raspberry if you setup things correctly, but having a monitor + keyboard and ethernet connection will make the setup easier and can greatly help in case something doesn't work as expected. I needed to connect my Raspberry to the network and I had only a monitor, so it was a lot of trial & error for me to get it working.



Raspberry Pi Dumb Terminal
(/id/Raspberry-Pi-Dumb-Terminal/)
by Tymkrs (/member/Tymkrs/)

Step 1: Identify USB UART convertor RX / TX lines

You'll need to be sure which pins on the converter are the RX / TX serial lines. Don't rely on markings on the converter PCB as they may not be correct! You can skip this step if you know which lines are TX and RX.

If you are not sure which line is TX, you can attach a resistor and LED in series to the suspected pin, the LED should blink when you write something to the UART.

Assuming the /dev/ttyUSB0 is the convertor:

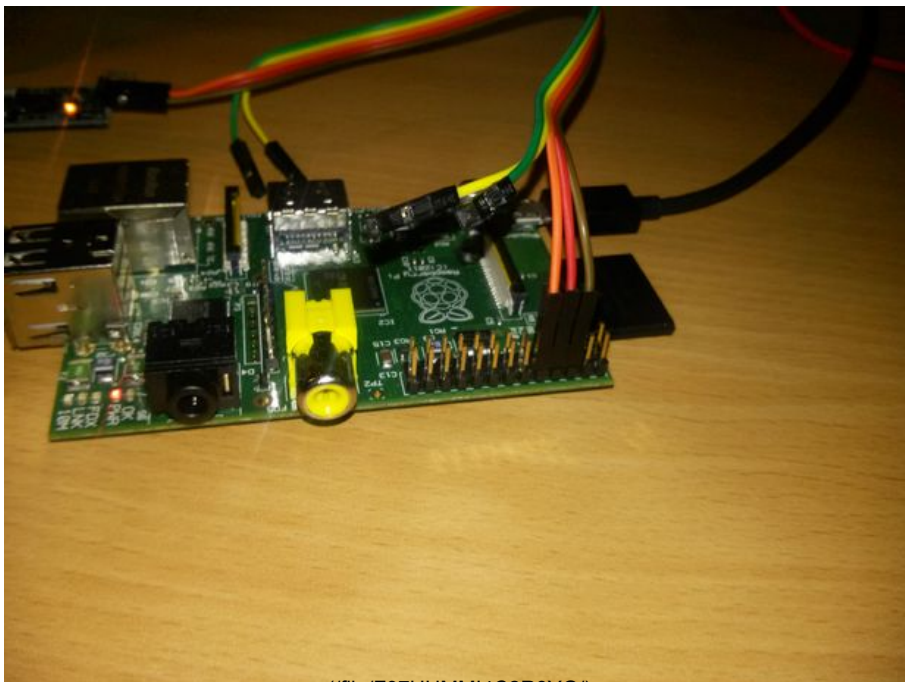
```
# This will make a LED attached to the TX blink once
stty -F /dev/ttyUSB0 9600 raw
dd if=/dev/zero of=/dev/ttyUSB0 bs=1000 count=1
```

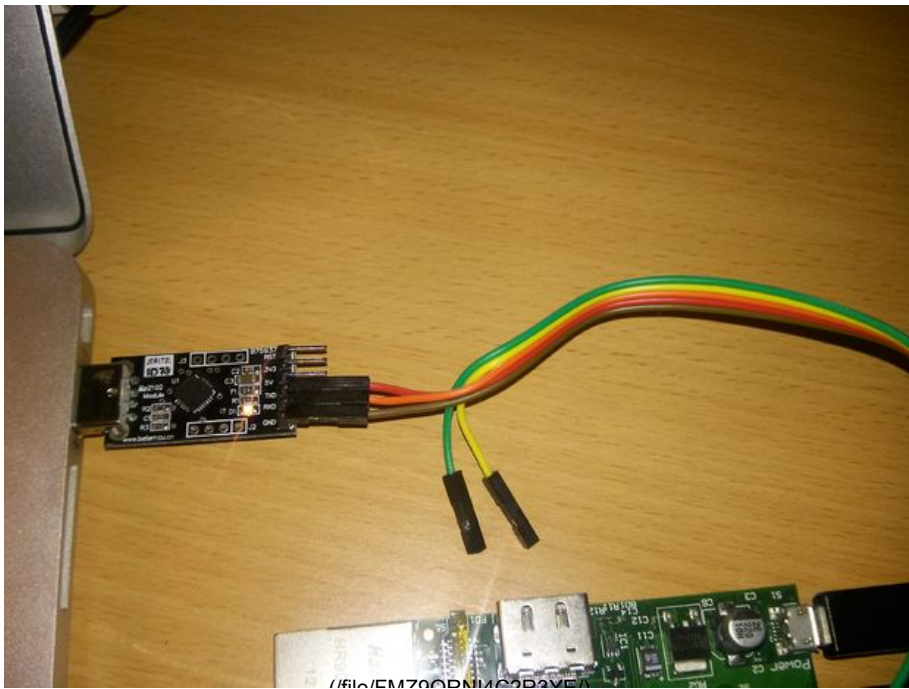
Once you have identified the TX pin, you can connect it to the suspected RX pin on the converter to make an echo loop. You should be able to receive what you write to it.

```
screen /dev/ttyUSB0 115200
```

When you type characters to the screen session, the screen should echo them when TX-RX are connected.

Step 2: Wire your computer to the Raspberry Pi





Connect your USB <-> UART convertor to the Raspberry Pi. You need to connect TX, RX and GND lines. TX stands for "transmitter" and RX stands for "receiver". When connecting two UART devices, you must connect them in a way that TX and RX form a crossover. That means that the output from a TX pin on one device goes to an RX input pin on the other device and vice versa. This is sometimes called a null-modem.

UART Convertor	Raspberry PI GPIO
GND -----	GND
TX output pin -----	RXD pin 10 on GPIO header
RX input pin -----	TXD pin 8 on GPIO header

Make sure that your convertor uses 3.3v logic levels output! **The Raspberry uses 3.3v logic** on its GPIO and higher voltage may damage the GPIO! If your convertor uses 5v logic levels you must use some logic level convertor.

Step 3: Test the connection using a console

By default, the Raspbian uses the built-in UART for linux console access and kernel debugging. Now, we will take advantage of this to test our wiring. Later we will disable this feature and use the UART for networking exclusively.

The Raspi's UART is accessible via /dev/ttyAMA0 special device.

First, make sure that the serial console is enabled. In cmdline.txt (http://elinux.org/RPi_cmdline.txt), you should see something like:

```
dwc_otg.lpm_enable=0 console=ttyAMA0,115200 kgdboc=ttyAMA0,115200 console=tty1 root=/dev/mmcblk0p2 rootfstype=ext4 elevator=deadline rootwait
```

The most important part is **console=ttyAMA0,115200** this will tell the kernel to assume the UART as a system console.

Edit the file manually or use a raspi-config (Advanced options) to enable the console on the serial port.

Second, on your linux computer, type:

```
screen /dev/ttyUSB0 115200
```

This will connect to the Raspberry Pi console. You should see a login prompt and be able to login. If you don't see anything, try pressing Enter. Also you should see kernel messages while the Raspi boots.

You can use the the serial console to connect to the Raspberry Pi if you don't have a monitor and a keyboard. Just make sure that the console is enabled in the cmdline.txt

Once you've checked that the serial connection works, you can proceed to the next step.

Step 4: Install necessary tools

You will need the 'pppd' tool to manage the connection. It is a part of the 'ppp' package.

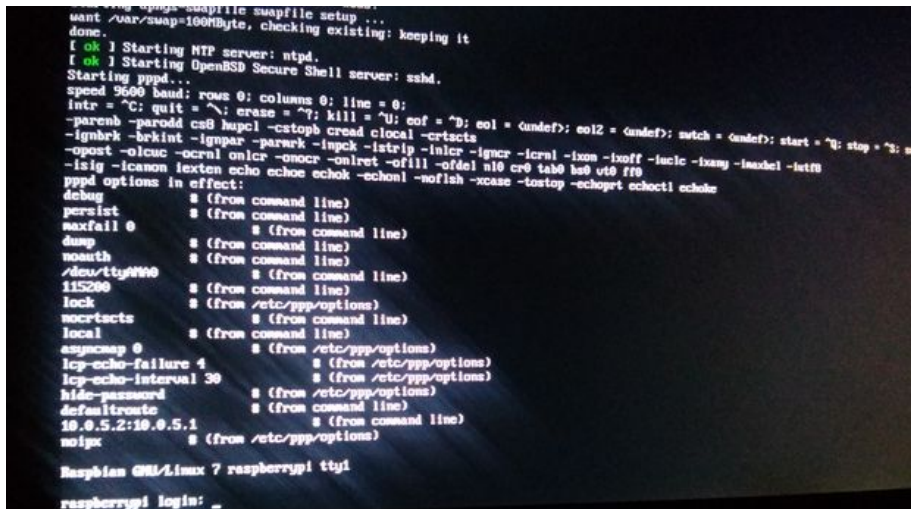
Install the ppp package both to your linux host and the Raspberry Pi:

```
sudo apt-get install ppp
```

If you don't have an access to the Internet already from your Raspberry Pi (a chicken-egg problem) you can manually download the .deb packages, place them on the sd card or a flash drive and manually install them. Just make sure you download and install all the dependencies.

```
sudo dpkg -i xyz.deb
```

Step 5: Take over the Raspberry Pi's UART



```
want to open swapfile swapfile setup ...
done.
[ ok ] Starting NTP server: ntpd.
[ ok ] Starting OpenSSH Secure Shell server: sshd.
Starting pppd...
speed 9600 baud; rows 0; columns 0; line = 0;
intr = ^C; quit = ^\; crts = ^T; kill = ^U; eof = ^D; sol = <undef>; sol2 = <undef>; swch = <undef>; start = ^Q; stop = ^Z; m
-parenb -parodd cs8 hupcl -ctstpb -crtcts
-ignbrk -bkint -ignpar -parbrk -lncp -lstrip -lnlcr -lgnr -lcrnl -lrm -lrmf -lrmc -lrmg -lrmh -lrmf -lrmf
-opost -olcuc -ocrnl onlcr -onocr -onlret -ofill -ofdel nl0 cr0 tab0 h00 vt0 ffs
-lsigs -lcanon lxtenc echo echok echok -echonl -nofish -xcase -tostty -echopt echocti echok
pppd options in effect:
debug          # (from command line)
persist        # (from command line)
maxfail 0      # (from command line)
dmg            # (from command line)
noauth         # (from command line)
/dcu/ttyAMA0   # (from command line)
115200         # (from command line)
lock           # (from /etc/ppp/options)
noctracts      # (from command line)
local         # (from command line)
asyncreap 0    # (from /etc/ppp/options)
lcp-echo-failure 4 # (from /etc/ppp/options)
lcp-echo-interval 30 # (from /etc/ppp/options)
hide-password  # (from /etc/ppp/options)
defaultroute  # (from command line)
10.0.5.2:10.0.5.1 # (from command line)
noipx         # (from /etc/ppp/options)

Raspbian GNU/Linux 7 raspberrypi tty1
raspberrypi login: _
```


Now, when the serial link reliably works. We can disable the console feature and use the UART exclusively for networking.

First, remove all ttyAMA0 references from cmdline.txt, refer to the previous step if you don't know how:

```
dwc_otg.lpm_enable=0 console=tty1 root=/dev/mmcblk0p2 rootfstype=ext4 elevator=deadline rootwait
```

On next reboot, you'll be able to use ttyAMA0 for whatever you like.

In our case we will configure the UART for some reasonable behavior and start PPP Daemon to establish a connection over the wire.

These commands will basically do what we need:

```
stty -F /dev/ttyAMA0 raw
pppd /dev/ttyAMA0 115200 10.0.5.2:10.0.5.1 noauth local debug dump default
route noctscts
```

The IP addresses 10.0.5.2 and 10.0.5.1 denote local:remote addresses being used for the connection. You can choose any IP addresses, just make sure the network range is different from what you are already using. In our case the 10.0.5.2 will be the address of our Pi, the 10.0.5.1 will be address of our linux host.

As I don't have any access to the RaspberryPi apart from the serial port, I tweaked the pppd parameters a little bit and added them to the /etc/rc.local (<http://www.raspberrypi.org/documentation/linux/usage/rc-local.md>) script so they execute every time the Raspberry boots and the pppd keeps running all the time so I can connect & disconnect the Raspberry at will.

My rc.local:

```
echo "Starting pppd..."
stty -F /dev/ttyAMA0 raw
stty -F /dev/ttyAMA0 -a
pppd /dev/ttyAMA0 115200 10.0.5.2:10.0.5.1 noauth local debug dump default
route noctscts persist maxfail 0 holdoff 1
```

The 'persist maxfail 0 holdoff 1' parameters will make the pppd attempt connect forever.

When you plug a monitor to your Raspberry, you should see an output of the startup script on the screen.

Step 6: Run pppd on host

Start the ppp daemon on your computer:

```
sudo stty -F /dev/ttyUSB0 raw
sudo pppd /dev/ttyUSB0 115200 10.0.5.1:10.0.5.2 proxyarp local noauth debug
nodetach dump noctscts passive persist maxfail 0 holdoff 1
```

When pppd is running on both devices, they should handshake and establish a link. The output should look like this:

```

using channel 10
Using interface ppp0
Connect: ppp0 <--> /dev/ttyUSB0
sent [LCP ConfReq id=0x5 <asynctest 0x0> <magic 0xb4bad9a9> <pcomp> <accomp>]
rcvd [LCP ConfReq id=0x2 <asynctest 0x0> <magic 0x263f80e1> <pcomp> <accomp>]
sent [LCP ConfAck id=0x2 <asynctest 0x0> <magic 0x263f80e1> <pcomp> <accomp>]
rcvd [LCP ConfAck id=0x5 <asynctest 0x0> <magic 0xb4bad9a9> <pcomp> <accomp>]
sent [LCP EchoReq id=0x0 magic=0xb4bad9a9]
sent [CCP ConfReq id=0x4 <deflate 15> <deflate(old#) 15> <bsd v1 15>]
sent [IPCP ConfReq id=0x4 <compress VJ 0f 01> <addr 10.0.5.1>]
rcvd [LCP EchoReq id=0x0 magic=0x263f80e1]
sent [LCP EchoRep id=0x0 magic=0xb4bad9a9]
rcvd [LCP EchoRep id=0x0 magic=0x263f80e1]
rcvd [CCP ConfReq id=0x2 <deflate 15> <deflate(old#) 15> <bsd v1 15>]
sent [CCP ConfAck id=0x2 <deflate 15> <deflate(old#) 15> <bsd v1 15>]
rcvd [IPCP ConfReq id=0x2 <compress VJ 0f 01> <addr 10.0.5.2>]
sent [IPCP ConfAck id=0x2 <compress VJ 0f 01> <addr 10.0.5.2>]
rcvd [CCP ConfAck id=0x4 <deflate 15> <deflate(old#) 15> <bsd v1 15>]
Deflate (15) compression enabled
rcvd [IPCP ConfAck id=0x4 <compress VJ 0f 01> <addr 10.0.5.1>]
Cannot determine ethernet address for proxy ARP
local IP address 10.0.5.1

```

Now, from your linux host, you should be able to ping or ssh the RaspberryPi

```
ping 10.0.5.2
```

If the ping works, you can proceed to the next step.

Step 7: Setup routing and NAT

Once we've established an IP link between our computer and the Raspberry Pi, we can configure the computer to route traffic from the Raspi to the Internet.

First, enable the traffic forwarding (<http://www.ducea.com/2006/08/01/how-to-enable-ip-forwarding-in-linux/>):

```
sudo sysctl -w net.ipv4.ip_forward=1
```

Next, enable the NAT masquerade to translate Raspi's internal address to the computer's address. Assuming the 'eth0' is an interface on your computer that you use to connect to the outside world:

```
sudo iptables -t nat -A POSTROUTING -o eth0 -j MASQUERADE
```

Now, you should be able to ssh to the Raspberry and ping some internet server :-)

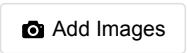
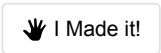
```
ssh pi@10.0.5.2
```

Finished!





We have a be nice comment policy.
Please be positive and constructive.



holla2040 (/member/holla2040)

an hour ago



Thanks for putting this together. This is exactly what I need for an off-grid hydro automation project.

Flag ☐ Feature Comment [delete]



MaykeS1 (/member/MaykeS1)

9 months ago



There's no log in prompt after the connection is established. Any ideas on that?

Flag



ludeks (/member/ludeks) (author) ▶ **MaykeS1 (/member/MaykeS1)**



9 months ago

Hi, you either use the UART for a console OR ppp network link. To be able to log in over UART, make sure the console is enabled ("console=ttyAMA0,115200" in cmdline.txt). If you use UART for networking, you can then log in over ssh.

Flag



seamster (/member/seamster)

2 years ago



Nicely done. Thanks for sharing this!

Flag

FEATURED CHANNELS

Woodworking
(/tag/type-id/category-workshop/channel-woodworking/)

Paper
(/tag/type-id/category-craft/channel-paper/)

Kitchen Hacks
(/tag/type-id/keyword-kitchen%20hacks/?play/channel-sort=FAVORITES)

Puzzles
(/tag/type-id/category-puzzles/)

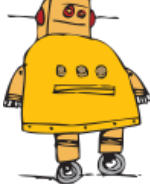
Space
(/tag/type-id/keyword-astronomy/?sort=FAVORITES)

Ice Cream
(/tag/type-id/keyword-ice%20cream/?sort=FAVORITES)

Homesteading
(/tag/type-id/category-home/channel-homesteading/)

3D Printing
(/tag/type-id/category-technology/channel-3D-Printing/)

Sewing
(/tag/type-id/category-craft/channel-sewing/)



Newsletter

Join 2 million + to receive instant
DIY inspiration in your inbox.

enter email

I'm in!

Mobile

Download our apps!

Android » (<https://play.google.com/store/apps/details?id=com.adsk.instructables>)

iOS » (<https://itunes.apple.com/app/instructables/id586765571>)

Windows » (<http://apps.microsoft.com/windows/en-us/app/7afc8194-c771-441a-9590-54250d6a8300>)

About Us

Who We Are (/about/)

Advertise (/advertise/)

Contact (/about/contact.jsp)

Jobs (/community/Positions-available-at-Instructables/)

Help (/id/how-to-write-a-great-instructable/)

Find Us

Facebook (<http://www.facebook.com/instructables>)

Youtube (<http://www.youtube.com/user/instructablestv>)

Twitter (<http://www.twitter.com/instructables>)

Pinterest (<http://www.pinterest.com/instructables>)

Google+ (<https://plus.google.com/+instructables>)

Resources

For Teachers (/teachers/)

Artists in Residence (/air)

Gift Premium Account (/account/give?sourcea=footer)

Forums (/community/)

Answers (/tag/type-question/?sort=RECENT)

Sitemap (/sitemap/)

Terms of Service (<http://usa.autodesk.com/adsk/servlet/item?siteID=123112&id=21959721>) |

Privacy Statement (<http://usa.autodesk.com/adsk/servlet/item?siteID=123112&id=21292079>) |

Legal Notices & Trademarks (<http://usa.autodesk.com/legal-notices-trademarks/>) | Mobile Site (<http://www.instructables.com>)



(<http://usa.autodesk.com/adsk/servlet/pc/index?id=20781545&siteID=123112>)

© 2016 Autodesk, Inc.