

Serial Communication

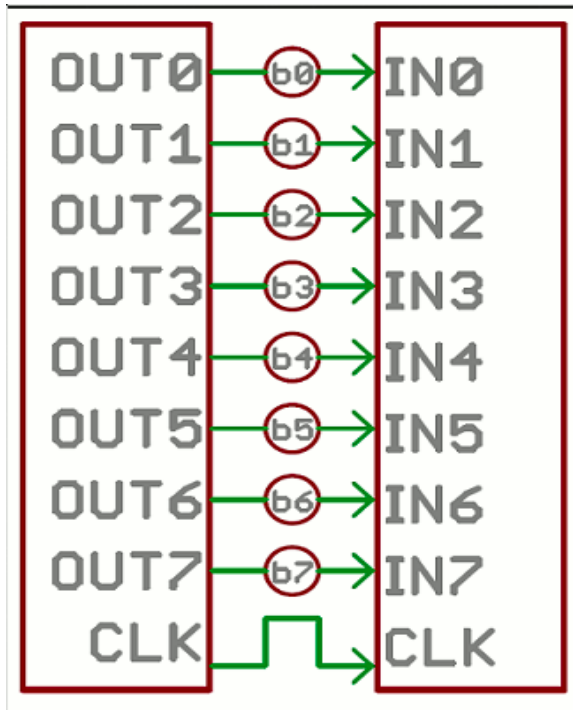
Serial protocols

Arduino

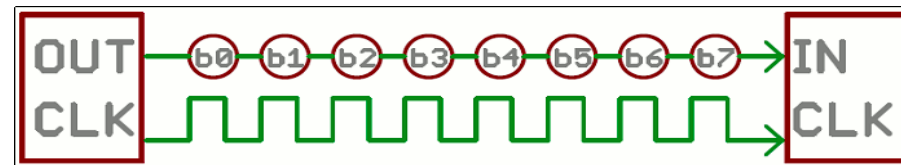
Arduino with Raspberry Pi

What is Serial Communication?

Parallel



Serial



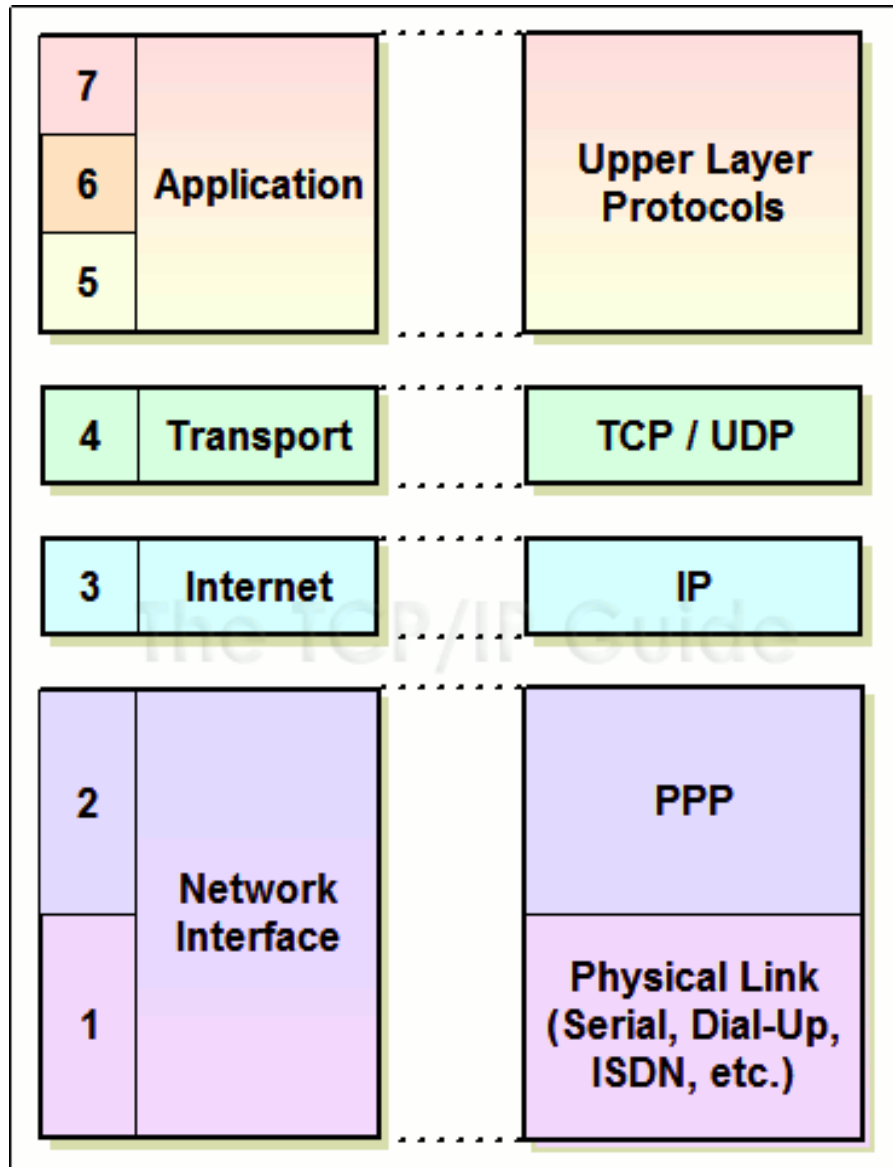
Examples:

Synchronous: I2C, SPI

Asynchronous: Bluetooth,
Xbee, USB, Ethernet

The rest of the story

Serial and the OSI model

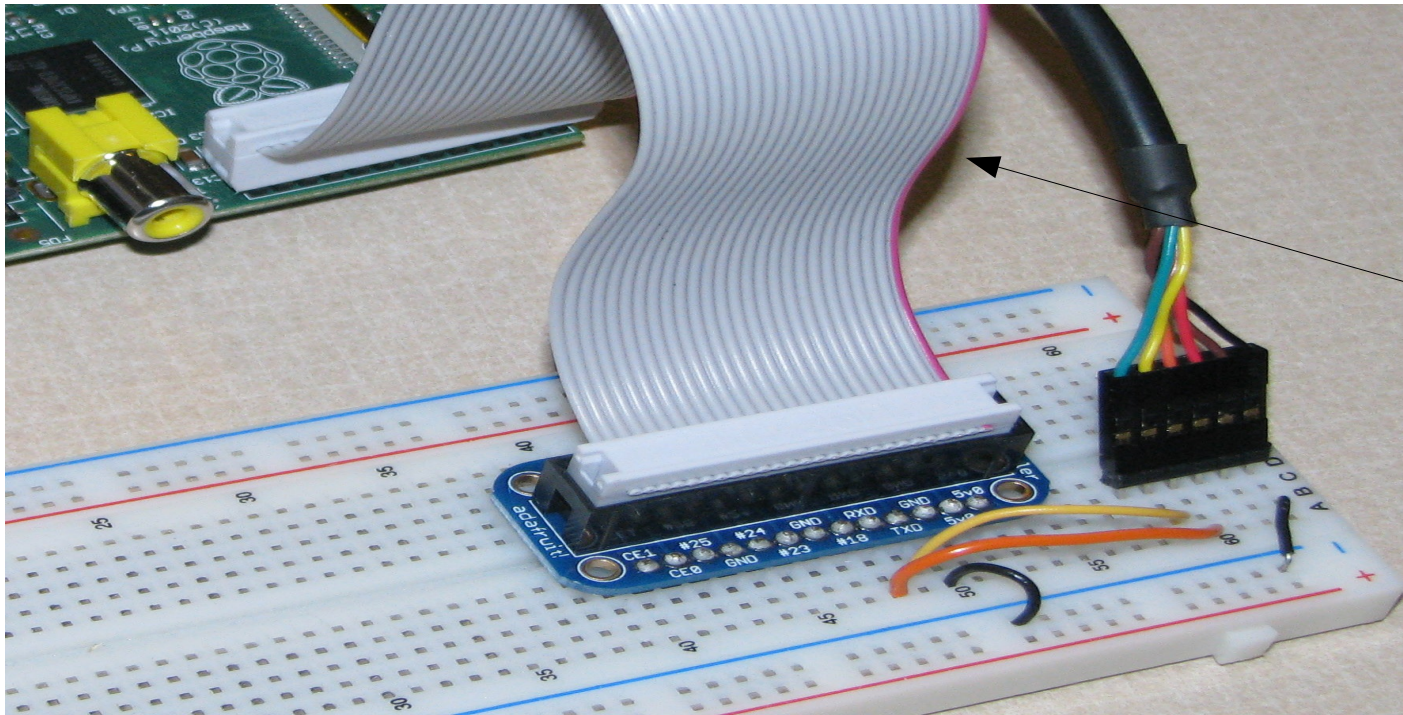


Older than the OSI model

Supports PPP

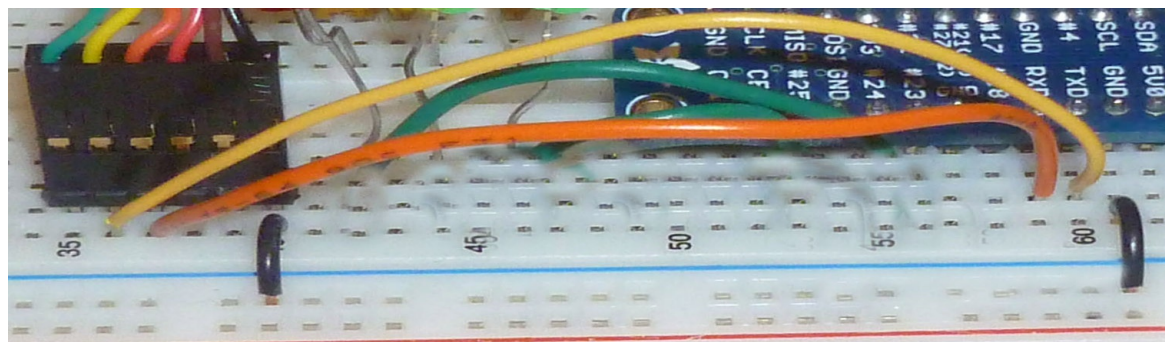
On-Line **guide**

Physical Serial Connection



The colored wire-
must connect to
pin 1 on the Pi

Detail:



Connect ground to ground and
RX to TX in both directions:

1. Connect FTDI yellow to Pi TX
2. Connect FTDI orange to Pi RX
3. Connect Gnd to Gnd via the power rail

A Linux Serial Connection

- First, determine the name of the serial device that is connected to the Pi. It is probably something like **`/dev/ttyUSBN`**.
- Connect to it with the following command:
`screen /dev/ttyUSBN 115200`
- Hit **Enter** a couple of times and hopefully the Pi will give you a login prompt.
- Use the two-character sequence Ctrl-A k to exit from **screen**.

Serial Port Programming using Python and pyserial

- Follow the steps provided, beginning at Step 2, in the [elinux guide to serial port programming](#) to write your first two programs using the serial port.
 - As mentioned in the above guide, be sure to disable the Linux default use of the serial port as described [here](#).
- Show your work before moving on.

Serial Communication between the Arduino and the RPi

- A quick look at the **Arduino Uno**
- At least 3 options for serial communication between Uno and RPi:
 - Using PySerial
 - Example **programs**
 - Using Nanpy
 - Examples **1**, **2**, and **3**
 - Using Webiopi ← today's effort
 - Also provides a web-based interface!

Using Webiopi for communication between Uno & RPi: Part 1

- Follow the setup description provided in the [Webiopi serial tutorial](#) with the following caveats/additions:
 - The Arduino IDE is installed on your desktop; no need to install it on your RPi
 - Move the USB cable connected to the Uno as necessary between the Pi and the desktop: connect to the desktop to program the Uno, connect to the Pi to communicate using WeblOPi
 - Use the Chrome browser on your desktop to interface with Webiopi; do not use Midori
 - No need to update Webiopi using SVN; the version we downloaded was created after the fix
 - Python serial still did **not** function properly when I tested it
 - The javascript implementation does work; we will use it
 - When running the code in the tutorial, adding the referenced hardware components to the breadboard is optional

Using Webiopi for communication between Uno & RPi: Part 2

- Try the following exercises provided in the **Webiopi serial tutorial**:
 - Serial Loopback trick with the Serial Monitor
 - Optional: Serial Loopback trick with REST mapping
 - Use this python script: wget <http://www.cs.unca.edu/~bruce/Fall14/loopScript.py>
 - Do not try Serial Loopback trick without REST mapping
 - Arduino Streaming Serial Monitor Test only
 - Do not try the python and javascript code (in the subdirectory: tutorials/4.serial-streaming)
 - Most important: try the Arduino Command Example
 - The code is in the subdirectory: tutorials/5.serial-command

In-Class Exercise

- Modify the code in `WebIOPi-0.7.0/tutorials/5.serial-command` to create a web interface that turns on and off 1 LED controlled by the Uno
 - You must use serial communication between the Pi and the Uno to accomplish this task
 - You will need to write an Uno program (modify the one provided in the directory *5.serial-command*)
 - You will also need to write a javascript program in the `index.html` file for WebIOPi (again, modify the one provided in the directory *5.serial-command*)
 - Hint: Look at the button functionalities in the *macros* example studied earlier this semester
 - Wire an LED to the Uno's breadboard to test your code.
 - Demonstrate your interface before leaving class