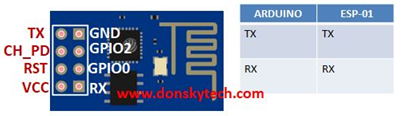
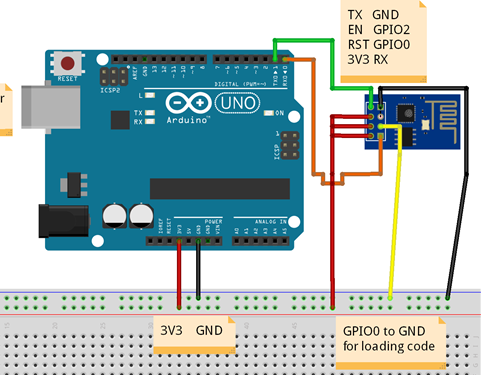
**Interfacing Computer to UNO to ISP-1S**

Connect ESP-01S with UNO as shown in the following fritzing diagram.





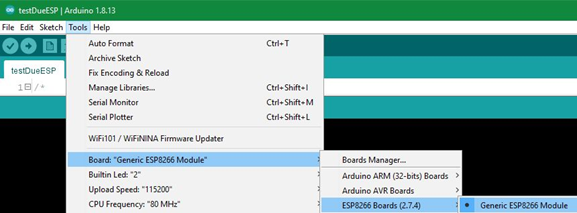
**Notes**

* ESP-01S **Tx** pin is connected to UNO **Tx** pin and ESP-01S **Rx** pin is connected to UNO **Rx** pin. This is not a mistake!
* ESP-01S **GPIO0** pin is connected to **GND**. This put the ESP8266 into flash override mode, allowing to download sketches into it.
* ESP-01S **3V3**, **EN** and **RST** pins are connected to UNO **3V3** pin
* ESP **GPIO2** pin is left floating
* Do not use the *5V* pin on UNO, this will destroy the ESP-01S!

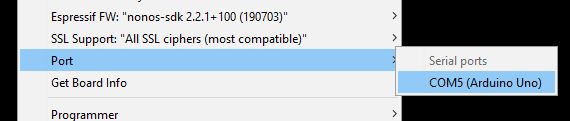
Connect UNO to PC with the standard USB cable and ESP8266 is ready to be programmed.

**Example**

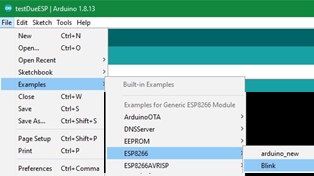
Select "Generic ESP8266 Module" as the board to use, see following figure.



Then select the port UNO is connected to. Here it is COM5:



Then open the blink example:



/\*

**Blink the blue LED on the ESP-01 module**

The blue LED on the ESP-01 module is connected to GPIO1

(which is also the TXD pin; so we cannot use Serial.print() at the same time)

Note that this sketch uses LED\_BUILTIN to find the pin with the internal LED

\*/

void setup() {

pinMode(LED\_BUILTIN, OUTPUT);

}

void loop() {

digitalWrite(LED\_BUILTIN, LOW);

delay(1000);

digitalWrite(LED\_BUILTIN, HIGH);

delay(2000);

}

Then compile compile button and finally load load button the sketch into the ESP8266 as you would load a sketch into the UNO.

|  |  |
| --- | --- |
|  | **To run the program**: first   * Disconnect the USB cable * Then disconnect ESP-01S GPIO0 pin.      * Reconnect USB cable to power ESP-01S * And watch the blue LED blink! |

**Troubleshooting**

Try loading the sketch again and again, disconnecting USB cable and reconnecting it straight after will help. See also [here](https://randomnerdtutorials.com/how-to-install-esp8266-board-arduino-ide/) for some more advice.