Internet of Things

Shabir Ali

Outline

- Communication Technologies
- WiFi
- ESP 8266
- Connecting ESP8266 with Arduino
- Install ESP8266 library
- Flashing ESP8266 from Arduino
- Connecting ESP8266 to AP
- Serial Data from ESP to Arduino
- Output at Serial Monitor

Communication Technologies

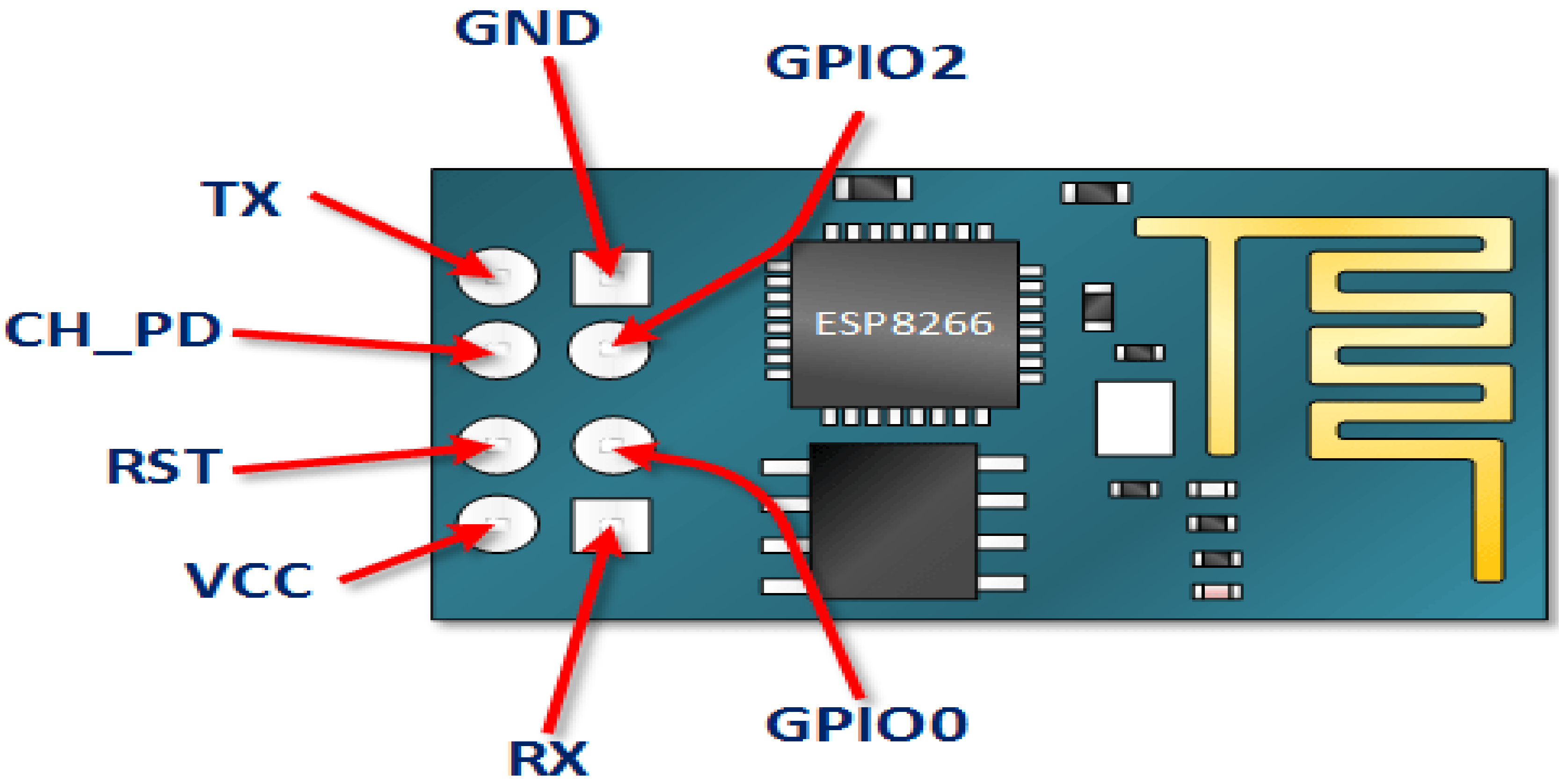
- Bluetooth
- Cellular (GSM SIM 3G/4G/5G)
- **802.11 (WiFi)**
- ZigBee
- Z-wave
- 6LowPAN
- NFC

802.11 (WiFi)

- Sending sensor data to internet for further processing.
- Need a wireless interface (hardware support).
- ESP8266 MCU.
- The ESP8266 is a low-cost Wi-Fi microchip with full TCP/IP stack and microcontroller capability produced by manufacturer Espressif Systems in Shanghai, China.
- It supports Wi-Fi 802.11 b/g/n around 2.4 GHz
- Supports ad-hoc, infrastructure,

  access point, mesh

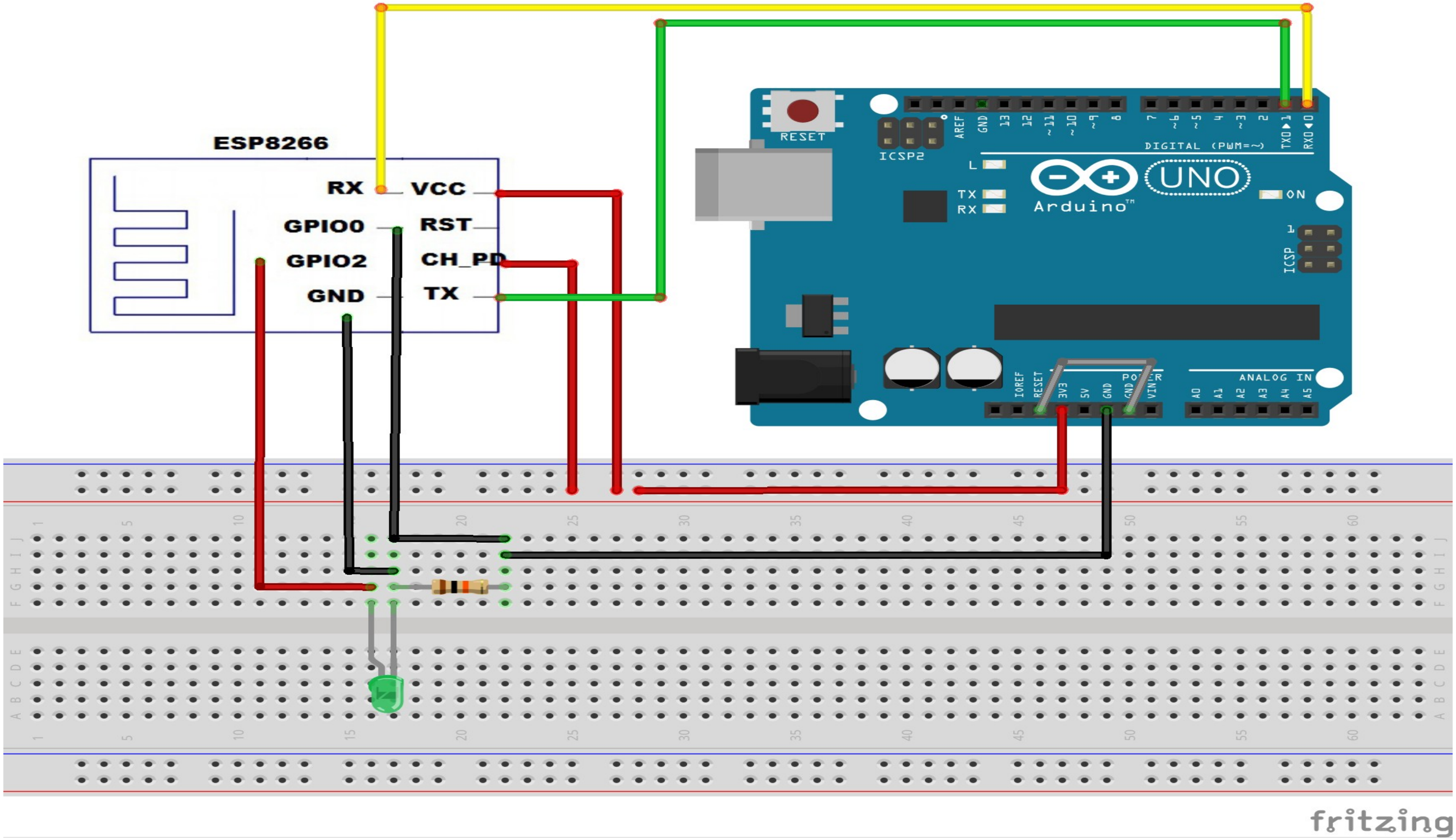
ESP8266



Pin out of ESP8266

- The pinout is as follows for the common ESP-01 module:
  - VCC, Voltage **(+3.3 V; can handle up to 3.6 V)**
  - GND, Ground (0 V)
  - RX, Receive data bit X (General-purpose input/output No. 3)
  - TX, Transmit data bit X (General-purpose input/output No. 1)
  - CH_PD, Chip power-down
  - RST, Reset
  - GPIO 0, General-purpose input/output No. 0
  - GPIO 2, General-purpose input/output No. 2

Connecting ESP8266 with Arduino



Install ESP8266 library

- First prepare arduino IDE to support ESP8266 board.
- Install esp8266 libraries from github
- Exit from arduino IDE and run below commands

```
sudo apt-get install git-core
sudo git config --global http.proxy http://edcguest:edcguest@172.31.100.14:3128
sudo git config --global https.proxy http://edcguest:edcguest@172.31.100.14:3128
```

Install ESP8266 library

- Go to Arduino downloaded directory from terminal then run commands in sudo mode:

```
cd arduino-1.8.9
cd hardware
mkdir esp8266com
cd esp8266com
sudo git clone https://github.com/esp8266/Arduino.git esp8266
cd esp8266
sudo git submodule update --init
```

Install ESP8266 library

- export http_proxy=’http://edcguest:edcguest@172.31.100.14:3128’
- export https_proxy=’http://edcguest:edcguest@172.31.100.14:3128’
- sudo apt-get install make unrar-free autoconf automake libtool gcc g++ gperf flex
- sudo apt-get install bison texinfo gawk ncurses-dev libexpat-dev python-dev python
- sudo apt-get install python-serial sed git unzip bash help2man wget bzip2
- sudo apt-get install libtool-bin
- cd tools/
- python3 get.py

- Restart Arduino IDE and verify it (Tools-->Boards-->ESP8266)

Flashing ESP8266 from Arduino

```
#include <ESP8266WiFi.h>
void setup()
{
  Serial.begin(115200);
  Serial.println();
  WiFi.begin("MNNIT");
  Serial.print("Connecting");
  while (WiFi.status() != WL_CONNECTED)
  {
    delay(500);
    Serial.println("trying");
  }
  Serial.println("WIFI Status:" + WiFi.status());
  Serial.println("Connected IP Address is:" + WiFi.localIP().toString());
}
```

```
void loop() {
  String ip="";
  delay(5000);
  if(WiFi.status() == WL_CONNECTED)
  {
    ip = WiFi.localIP().toString();
    Serial.println("Connected IP Address is:" + ip);
  }
  }
```

Flashing ESP8266 from Arduino

- Verify Port (/dev/ttyUSB0 or /dev/ttyUSB1)
- Select board: Generic ESP8266 Module
- Connect RESET line of Arduino to GND (GPIO0+GND) of ESP8266
- TX → TX
- RX → RX
- Disconnect GPIO2 Line (if connected to any pin)
- Verify and upload sketch

Connecting ESP8266 to AP

- After successful flashing the sketch to ESP8266.
- Disconnect USB cable from computer
- Disconnect RESET and GPIO0 line
- Reverse the TX and RX line (if BUILTIN pin)
- Connect TX → pin2
- Connect Rx → pin3

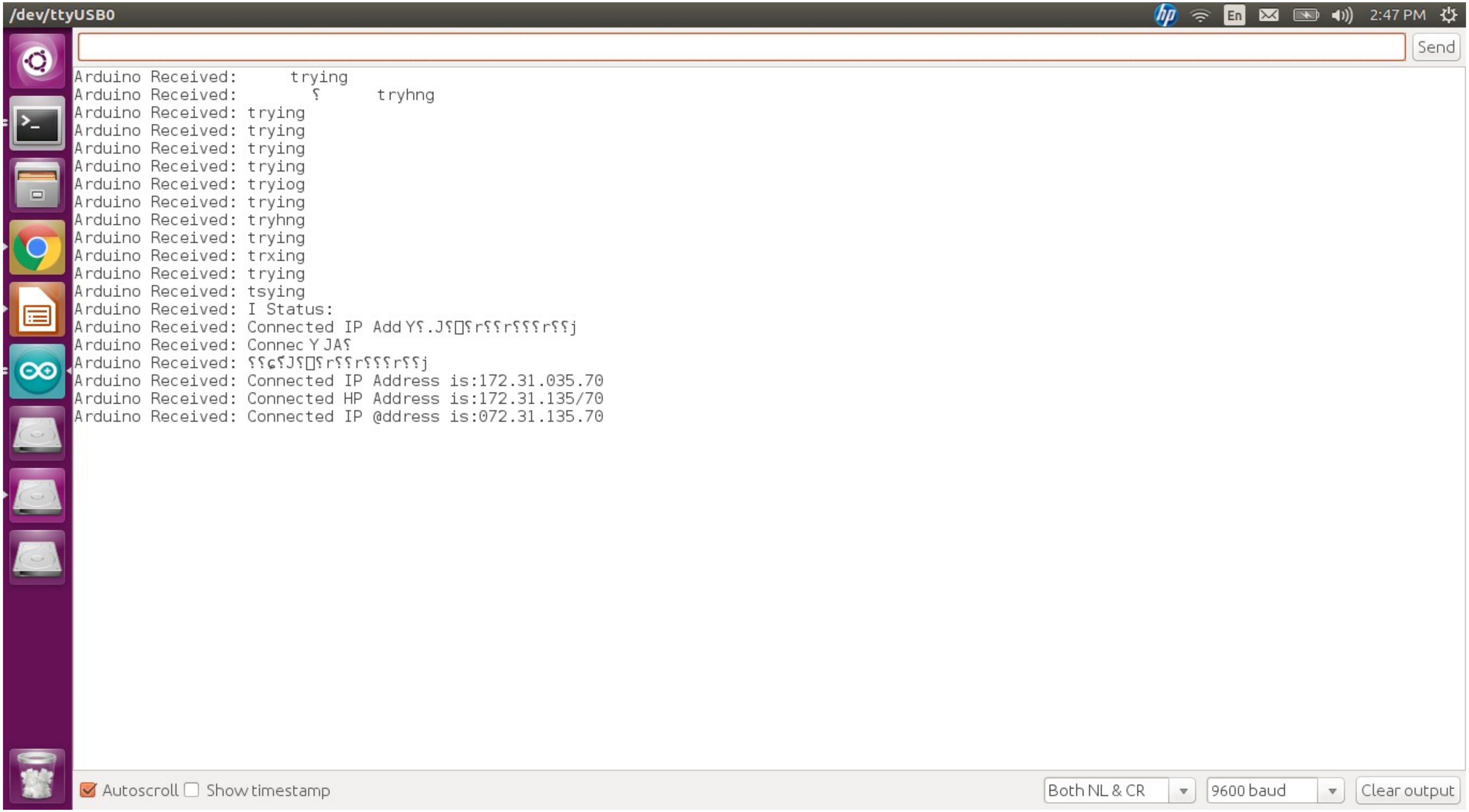
Serial Data from ESP to Arduino

- #include<SoftwareSerial.h>
- SoftwareSerial esp8266(2, 3); // RX, TX
- void setup() {
-   Serial.begin(9600);
-   esp8266.begin(115200);
-   delay(5000);
- }

```
void loop() {
  String inData="";
while (esp8266.available() > 0)
    {
        char received = esp8266.read();
        inData += received;

        // Process message when new line
character is recieved
        if (received == '\n')
        {
            Serial.print("Arduino Received: ");
            Serial.print(inData);
            inData = "";  // Clear received buffer
        }
    }
}
```

Output at Serial Port



```
Arduino Received:      trying
Arduino Received:        ʕ      tryhng
Arduino Received: trying
Arduino Received: trying
Arduino Received: trying
Arduino Received: trying
Arduino Received: tryiog
Arduino Received: trying
Arduino Received: tryhng
Arduino Received: trying
Arduino Received: trxing
Arduino Received: trying
Arduino Received: tsying
Arduino Received: I Status:
Arduino Received: Connected IP Add Yʕ.Jʕ□ʕrʕʕrʕʕʕrʕʕj
Arduino Received: Connec Y JAʕ
Arduino Received: ʕʕɕʕJʕ□ʕrʕʕrʕʕʕrʕʕj
Arduino Received: Connected IP Address is:172.31.035.70
Arduino Received: Connected HP Address is:172.31.135/70
Arduino Received: Connected IP @ddress is:072.31.135.70
```

☑ Autoscroll □ Show timestamp

Both NL & CR ▾ 9600 baud ▾

Verify reachability of ESP

- Ping to obtained IP address from any machine of lab

KEEP
CALM
AND
TRY IT
YOURSELF