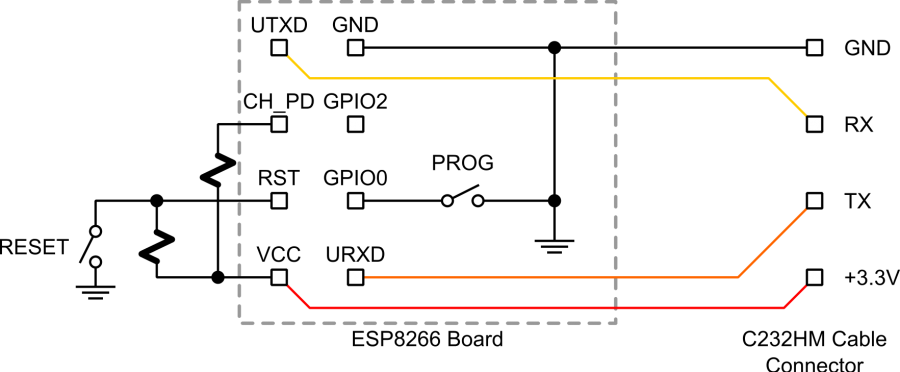
ESP8266 ESP-01



Wiring Board: <http://nfriedly.com/techblog/2015/07/build-a-diy-esp8266ex-esp-01-dev-test-programming-board/>

## ESP-01 Board:

CH\_PD ALWAYS HIGH (Enabled) via pull up resistor

## USB Cable Connector:

GND to Breadboard GND

3.3V to Breadboard VCC

TX -> ESP RX **// Once the program is uploaded, it can be repurpose to some other serial use**

RX -> ESP TX **// Once the program is uploaded, it can be repurpose to some other serial use**

p.s. Using CH340G Connector – Not enough Current!

## To Flash Program:

GPIO0 -> GND via Switch

RST -> Float and when ready to flash, GND for a moment

## To auto run (i.e. every time power up)

GPIO0 -> HIGH / Float via Switch

Builtin LED: GPIO PIN 1

GPIO2: GPIO PIN 2

const int ESP01\_BUILTINLED = 1;

const int ESP01\_GPIO2\_LED = 2; **// Using GPIO2**

void setup() {

pinMode(ESP01\_BUILTINLED, OUTPUT);

pinMode(ESP01\_GPIO2\_LED, OUTPUT);

Serial.begin(115200);

}

int count = 0;

void loop() {

count++;

Serial.printf("-->: %d\n", count);

digitalWrite(ESP01\_BUILTINLED, LOW);

digitalWrite(ESP01\_GPIO2\_LED, LOW);

delay(1000);

digitalWrite(ESP01\_BUILTINLED, HIGH);

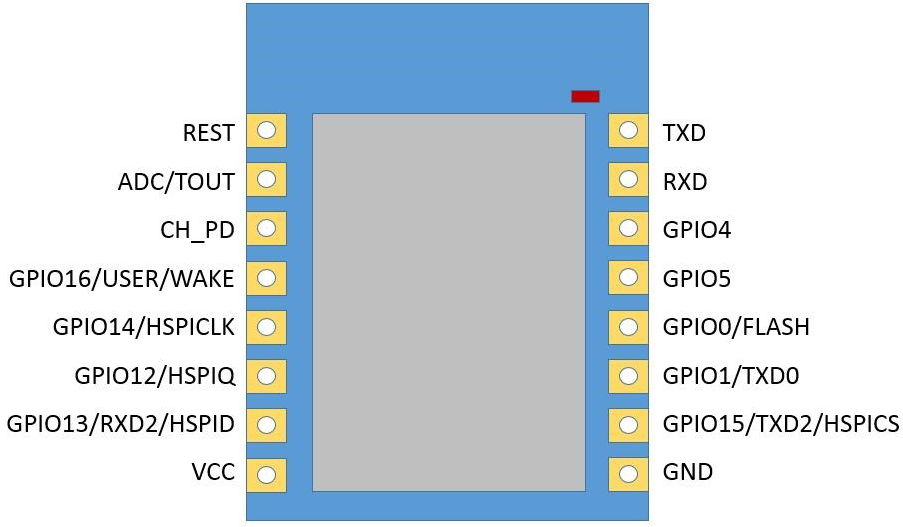
digitalWrite(ESP01\_GPIO2\_LED, HIGH);

delay(2000);

}

Examples: <https://github.com/esp8266/Arduino/tree/master/libraries/esp8266/examples>

ESP-12



Arduino IDE, Board NodeMCU 1.0 ESP-12E Module

const int ESP12\_GPIO0 = 0;

const int ESP12\_GPIO1 = 1;

const int ESP12\_GPIO2 = 2; // builtin LED

const int ESP12\_GPIO4 = 4;

const int ESP12\_GPIO5 = 5;

const int ESP12\_GPIO12 = 12;

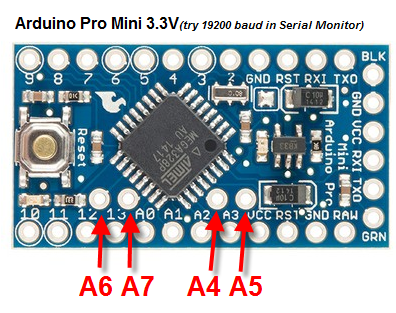
const int ESP12\_GPIO13 = 13;

const int ESP12\_GPIO14 = 14;

const int ESP12\_GPIO15 = 15;

const int ESP12\_GPIO16 = 16;

Arduino Pro Mini





<https://cdn.sparkfun.com/datasheets/Dev/Arduino/Boards/ProMini8MHzv1.pdf>

ILI9340 2.2” TFT SPI 320X240 3.3V

Mine is 9340, NOT 9341!

Install Libraries:

<https://github.com/adafruit/Adafruit-GFX-Library>

<https://github.com/adafruit/Adafruit_ILI9340>

I can use my Arduino Pro Mini which is **3.3V,**

**BUT use external 3.3V to power LED/VCC to avoid too much current drawn through Arduino Pro Mini**

SCK - Arduino Digital Pin 13  
SDO(MISO) - Arduino Digital Pin 12  
SDI(MOSI) - Arduino Digital Pin 11  
CS - Arduino Digital Pin 10  
D/C - Arduino Digital Pin 9   
RESET - Arduino Digital Pin 8  
LED - 3.3V external  
VCC - 3.3V external  
GND – GND

Ref: <http://www.instructables.com/id/Cheap-TFT-22-inch-Display-on-Arduino-ILI9340C-or-I/?ALLSTEPS>

Pulse Sensor

Install Arduino Library:

<https://github.com/WorldFamousElectronics/PulseSensor_Amped_Arduino>

PulseSensor – Arduino Pin

+ (GREY) - 3.3V

- (WHITE) - GND

Data (PURPLE) - Analog 0

Install Processing Library:

<https://github.com/WorldFamousElectronics/PulseSensor_Amped_Processing_Visualizer>

Open file “PulseSensorAmpd\_Processing\_1dot1.pde” with Processing App

Change the code:

String portName = "COM5"; // specify the right port!

port = new Serial(this, portName, 115200);

Run Processing.

Tutorial:

* <http://pulsesensor.com/pages/code-and-guide>
* <http://pulsesensor.com/pages/pulse-sensor-speaker-tutorial>
* <http://pulsesensor.com/pages/pulse-sensor-amped-arduino-v1dot1>

Forum

<http://pulsesensor.proboards.com/>