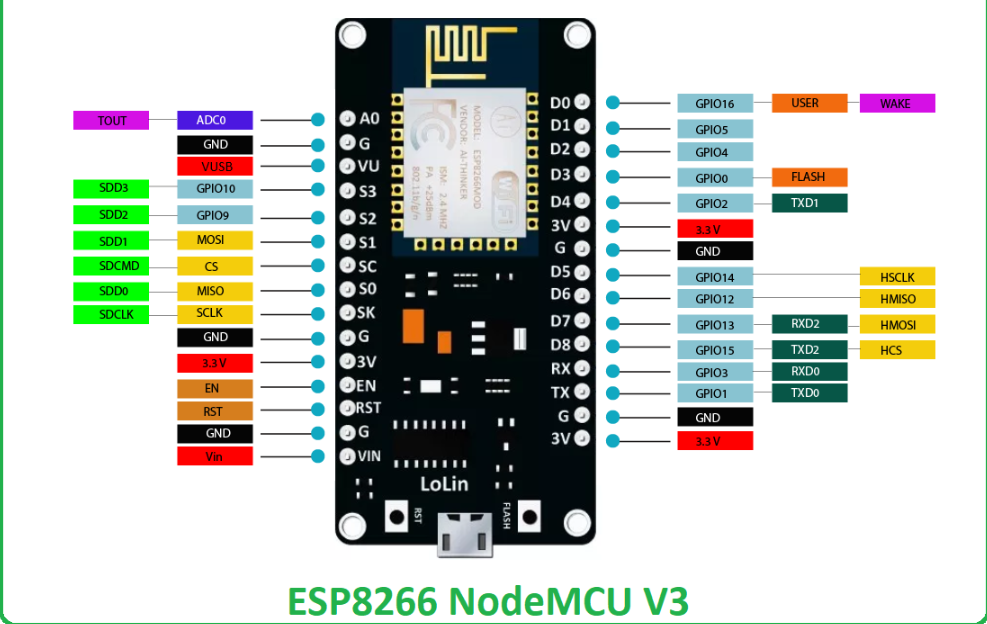
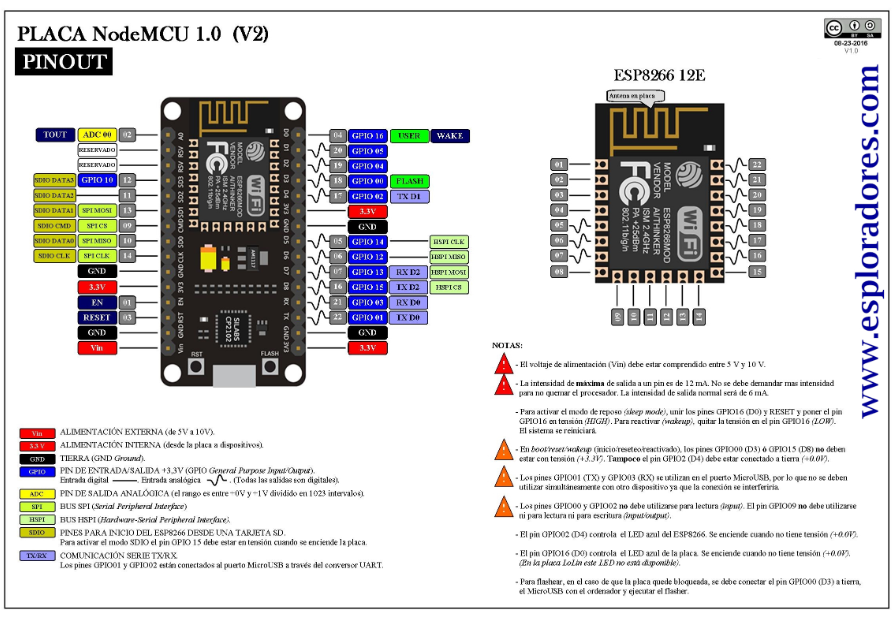
zModulo



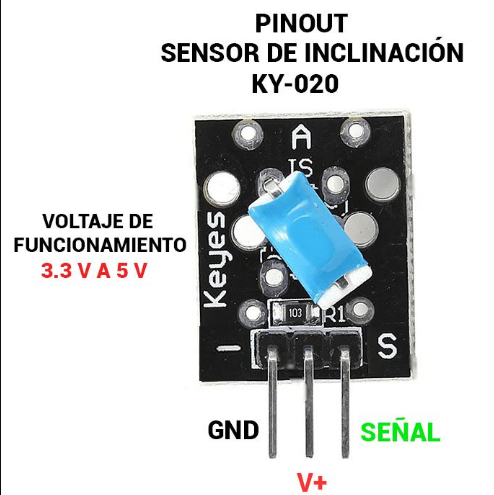


<https://www.esploradores.com/wp-content/uploads/2019/11/PINOUT-NodeMCU_1.0-V2-2_2.jpg>

<https://www.theengineeringprojects.com/2018/08/esp8266-pinout-datasheet-features-applications.html>

<https://espressif.com/sites/default/files/documentation/0a-esp8266ex_datasheet_en.pdf>

Sensor de mov



<https://sensorkit.joy-it.net/en/sensors/ky-020>

<https://uelectronics.com/producto/modulo-ky-020-sensor-de-inclinacion/>



Modulo

Ejemplos

https://hetpro-store.com/esp8266ex/

<http://www.icstation.com/mobile/nodemcu-wifi-internet-network-development-board-module-esp8266-p-8395.html>

repositorio oficial del modulo

<https://github.com/esp8266/arduino>

Librería

<https://www.instructables.com/Instalar-driver-para-CH340G/>

Videos de apoyo

<https://rogerbit.com/wprb/2021/02/como-encender-luces-con-telegram-y-esp32-desde-cualquier-parte-del-mundo/>

programación

<https://rogerbit.com/wprb/2021/02/como-encender-luces-con-telegram-y-esp32-desde-cualquier-parte-del-mundo/>

<https://elprofemichely.wordpress.com/2021/04/09/programacion-de-alarma-con-motion-sensor-que-envia-mensaje-a-telegram-2021/>

<https://www.youtube.com/watch?v=1jcMAtgw2EE&t=636s>

while(var < 200){

  // repite algo 200 veces

  var++;

}

}

///modulo wifi

#include <ESP8266WiFi.h>

#include <WiFiClientSecure.h>

#include <UniversalTelegramBot.h>

#define WIFI\_SSID "INFINITUMECA4\_2.4"

#define WIFI\_PASSWORD "v68AHWq6yV"

#define BOT\_TOKEN "5404446720:AAFzaBNBKoiXjphguIGFbYvY-FW7oOGhqaw"

const unsigned long BOT\_MTBS = 1000; // mean time between scan messages

X509List cert(TELEGRAM\_CERTIFICATE\_ROOT);

WiFiClientSecure secured\_client;

UniversalTelegramBot bot(BOT\_TOKEN, secured\_client);

unsigned long bot\_lasttime; // last time messages' scan has been done

const int ledPin = LED\_BUILTIN;

int ledStatus = 0;

const int pinservo = 5;

void handleNewMessages(int numNewMessages)

{

Serial.print("handleNewMessages ");

Serial.println(numNewMessages);

for (int i = 0; i < numNewMessages; i++)

{

String chat\_id = bot.messages[i].chat\_id;

String text = bot.messages[i].text;

String from\_name = bot.messages[i].from\_name;

if (from\_name == "")

from\_name = "Guest";

if (text == "/Bloqueo")

{

digitalWrite(ledPin, LOW); // turn the LED on (HIGH is the voltage level)

digitalWrite(pinservo, HIGH); //

ledStatus = 1;

bot.sendMessage(chat\_id, "Bloqueo activo", "");

}

if (text == "/Desbloqueo")

{

ledStatus = 0;

digitalWrite(ledPin, HIGH); // turn the LED off (LOW is the voltage level)

digitalWrite(pinservo, LOW); //

bot.sendMessage(chat\_id, "Desbloqueando sistema", "");

}

if (text == "/status")

{

if (ledStatus)

{

bot.sendMessage(chat\_id, "Bloqueado", "");

}

else

{

bot.sendMessage(chat\_id, "Desbloqueado", "");

}

}

if (text == "/start")

{

String welcome = "Soy el Bot del sistema estas son las opciones, " + from\_name + ".\n";

welcome += "Siguientes comandos.\n\n";

welcome += "/Bloqueo :El sistema se encuentra bloqueado\n";

welcome += "/Desbloqueo: Sistema desbloqueado\n";

welcome += "/status\n";

bot.sendMessage(chat\_id, welcome, "Markdown");

}

}

}

void setup()

{

Serial.begin(115200);

Serial.println();

pinMode(pinservo, OUTPUT);

pinMode(ledPin, OUTPUT); // initialize digital ledPin as an output.

delay(10);

digitalWrite(ledPin, HIGH); // initialize pin as off (active LOW)

// attempt to connect to Wifi network:

configTime(0, 0, "pool.ntp.org"); // get UTC time via NTP

secured\_client.setTrustAnchors(&cert); // Add root certificate for api.telegram.org

Serial.print("Connecting to Wifi SSID ");

Serial.print(WIFI\_SSID);

WiFi.begin(WIFI\_SSID, WIFI\_PASSWORD);

while (WiFi.status() != WL\_CONNECTED)

{

Serial.print(".");

delay(500);

}

Serial.print("\nWiFi connected. IP address: ");

Serial.println(WiFi.localIP());

// Check NTP/Time, usually it is instantaneous and you can delete the code below.

Serial.print("Retrieving time: ");

time\_t now = time(nullptr);

while (now < 24 \* 3600)

{

Serial.print(".");

delay(100);

now = time(nullptr);

}

Serial.println(now);

}

void loop()

{

if (millis() - bot\_lasttime > BOT\_MTBS)

{

int numNewMessages = bot.getUpdates(bot.last\_message\_received + 1);

while (numNewMessages)

{

Serial.println("got response");

handleNewMessages(numNewMessages);

numNewMessages = bot.getUpdates(bot.last\_message\_received + 1);

}

bot\_lasttime = millis();

}

}

// Arduino one

#include <Servo.h>

#define PUSH 3

int pushvalue = 0;

Servo servoMotor;

void setup() {

servoMotor.attach(2);

pinMode(PUSH, INPUT);

}

void loop() {

pushvalue = digitalRead(PUSH);

if(pushvalue==1){

servoMotor.write(105); // cerrado

delay(1000);

}

else {

servoMotor.write(90); // abierto

delay(3000);

}

}

///programa funcionando bloquea el griper pero con el Arduino en paralelox|

#include <ESP8266WiFi.h>

#include <WiFiClientSecure.h>

#include <UniversalTelegramBot.h>

#define WIFI\_SSID "INFINITUMECA4\_2.4"

#define WIFI\_PASSWORD "v68AHWq6yV"

#define BOT\_TOKEN "5404446720:AAFzaBNBKoiXjphguIGFbYvY-FW7oOGhqaw"

const unsigned long BOT\_MTBS = 1000; // mean time between scan messages

X509List cert(TELEGRAM\_CERTIFICATE\_ROOT);

WiFiClientSecure secured\_client;

UniversalTelegramBot bot(BOT\_TOKEN, secured\_client);

unsigned long bot\_lasttime; // last time messages' scan has been done

const int ledPin = LED\_BUILTIN;

int ledStatus = 0;

const int pinservo = 5; // PIN D1 DEL MODULO WIFI

void handleNewMessages(int numNewMessages)

{

Serial.print("handleNewMessages ");

Serial.println(numNewMessages);

for (int i = 0; i < numNewMessages; i++)

{

String chat\_id = bot.messages[i].chat\_id;

String text = bot.messages[i].text;

String from\_name = bot.messages[i].from\_name;

if (from\_name == "")

from\_name = "Guest";

if (text == "/Bloqueo")

{

digitalWrite(ledPin, LOW); // turn the LED on (HIGH is the voltage level)

digitalWrite(pinservo, HIGH); //

ledStatus = 1;

bot.sendMessage(chat\_id, "Bloqueo activo", "");

}

if (text == "/Desbloqueo")

{

ledStatus = 0;

digitalWrite(ledPin, HIGH); // turn the LED off (LOW is the voltage level)

digitalWrite(pinservo, LOW); //

bot.sendMessage(chat\_id, "Desbloqueando sistema", "");

}

if (text == "/status")

{

if (ledStatus)

{

bot.sendMessage(chat\_id, "Bloqueado", "");

}

else

{

bot.sendMessage(chat\_id, "Desbloqueado", "");

}

}

if (text == "/start")

{

String welcome = "Soy el Bot del sistema estas son las opciones, " + from\_name + ".\n";

welcome += "Siguientes comandos.\n\n";

welcome += "/Bloqueo :El sistema se encuentra bloqueado\n";

welcome += "/Desbloqueo: Sistema desbloqueado\n";

welcome += "/status\n";

bot.sendMessage(chat\_id, welcome, "Markdown");

}

}

}

void setup()

{

Serial.begin(115200);

Serial.println();

pinMode(pinservo, OUTPUT);

pinMode(ledPin, OUTPUT); // initialize digital ledPin as an output.

delay(10);

digitalWrite(ledPin, HIGH); // initialize pin as off (active LOW)

// attempt to connect to Wifi network:

configTime(0, 0, "pool.ntp.org"); // get UTC time via NTP

secured\_client.setTrustAnchors(&cert); // Add root certificate for api.telegram.org

Serial.print("Connecting to Wifi SSID ");

Serial.print(WIFI\_SSID);

WiFi.begin(WIFI\_SSID, WIFI\_PASSWORD);

while (WiFi.status() != WL\_CONNECTED)

{

Serial.print(".");

delay(500);

}

Serial.print("\nWiFi connected. IP address: ");

Serial.println(WiFi.localIP());

// Check NTP/Time, usually it is instantaneous and you can delete the code below.

Serial.print("Retrieving time: ");

time\_t now = time(nullptr);

while (now < 24 \* 3600)

{

Serial.print(".");

delay(100);

now = time(nullptr);

}

Serial.println(now);

}

void loop()

{

if (millis() - bot\_lasttime > BOT\_MTBS)

{

int numNewMessages = bot.getUpdates(bot.last\_message\_received + 1);

while (numNewMessages)

{

Serial.println("got response");

handleNewMessages(numNewMessages);

numNewMessages = bot.getUpdates(bot.last\_message\_received + 1);

}

bot\_lasttime = millis();

}

}