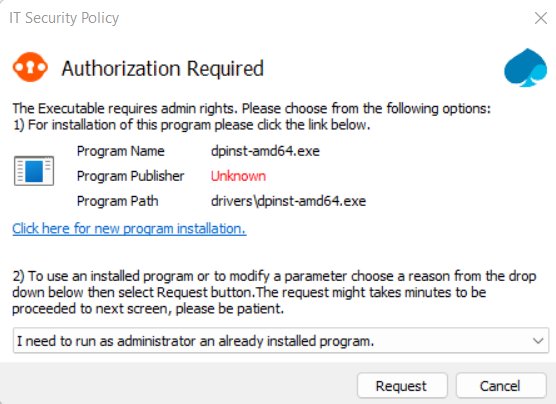
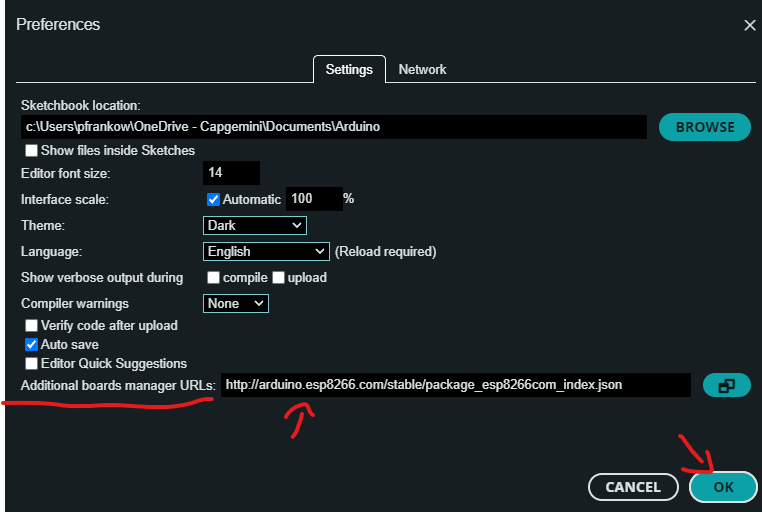
# Przygotowanie przed szkoleniem

1. Zainstaluj *Arduino IDE* - > <https://www.arduino.cc/en/software>
2. Otwórz *Arduino IDE*.
3. Od tej pory, w którymś momencie może wyskoczyć pierwszy request z zezwoleniem na instalacje *dpinst-amd64.exe*. Wystarczy wtedy zignorować klikająć: *Cancel*

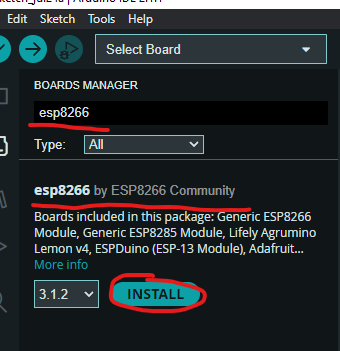


1. Wejdź w *File-> Preferences* i w polu *Additional Board Manager URLs* wpisz:

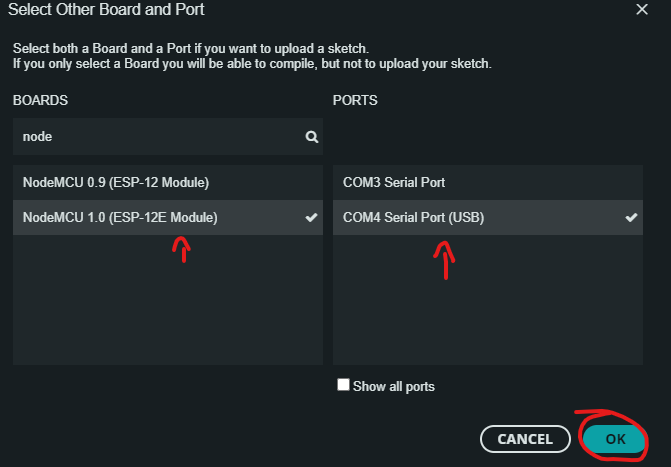
<http://arduino.esp8266.com/stable/package_esp8266com_index.json> oraz kliknij *OK*.  


1. Ściągnij bibliotekę dla płytki ESP8266. Przejdź do:  
   *- Tools – Board – Boards Manager...*

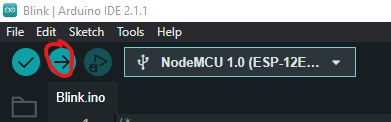
Wyszukaj *esp8266* i zainstaluj (zwróć uwagę przez kogo została wydana!)



1. Podłącz płytkę poprzez USB. Sprawdź port, pod którym masz podłączony ESP8266.  
   Na Windowsie wejdź w *Device Manager* i po rozwinięciu pola *Ports* znajdź urządzenie opisane *USB-SERIAL CH340 (COMX)*.
2. W Arduino IDE kliknij w *Select Board – Select Other Board and Port.* wybierz port, do którego jesteś podłączony (*COMX*)oraz rodzaj płytki (***NodeMCU 1.0 ESP-12E Module***)



1. Aby się upewnić, że płytka jest poprawnie skonfigurowana wejdź w *File – Examples – 01.Basics – Blink*  
   Po załadowaniu kodu podmień wszędzie w kodzie *LED\_BUILTIN* na *2*   
   Wgraj go na płytkę klikając *Upload*



Jeżeli kod wgrał się bezbłędnie, niebieska lampka na płytce powinna migać.  
Gratuluję! Oto pierwszy program został przez Ciebie wgrany! 😉

# . LED Blink

**Cel**: LED ma świecić się przez 1 sek., a następnie zgasnąć na 1 sek. Funkcjonalność ma być zapętlona.

## 1. 1 Schemat podłączenia

## 

## 1. 2 Kod

const int d5Led = 14;

void setup() {

  pinMode(d5Led, OUTPUT);

}

void loop() {

  digitalWrite(d5Led, HIGH);

  delay(1000);

  digitalWrite(d5Led, LOW);

  delay(500);

}

# . PWM LED

**Cel**: LED ma się stopniowo rozjaśniać a następnie stopniowo przyciemniać.

## 2. 1 Schemat podłączenia

Taki sam jak w poprzednim ćwiczeniu.

## 2. 2 Kod

const int d5Led = 14;

void fadeIn() {

  for (int i = 0; i <= 255; i++) {

    Serial.println(i);

    analogWrite(d5Led, i);

    delay(10);

  }

}

void fadeOut() {

  for (int i = 255; i >= 0; i--) {

    Serial.println(i);

    analogWrite(d5Led, i);

    delay(10);

  }

}

void setup() {

  pinMode(d5Led, OUTPUT);

  Serial.begin(9600);

}

void loop() {

  analogWrite(d5Led, 0);

  delay(1000);

  analogWrite(d5Led, 127);

  delay(1000);

  analogWrite(d5Led, 255);

  delay(1000);

  fadeIn();

  fadeOut();

}

# . HTML LED

**Cel**: Wystaw prostą stronę HTML, dzięki której będziesz mógł bezprzewodowo sterować swoim LEDem.

## 3. 1 Schemat podłączenia

Tak jak w poprzednim ćwiczeniu.

## 3. 2 Kod

#include <ESP8266WiFi.h>

#include "ESP8266WebServer.h"

const char\* myHTML = „ //Tutaj wklej gotowy HTML// ”;

const char\* ssid = "YOUR\_SSID";

const char\* password = "YOUR\_PASSW";

ESP8266WebServer server;

const int d5Led = 14;

void ledOn() {

  digitalWrite(d5Led, HIGH);

  server.send(200);

}

void ledOff() {

  digitalWrite(d5Led, LOW);

  server.send(200);

}

void pwmLed() {

  String duty = server.arg("duty");

  Serial.println(duty);

  analogWrite(d5Led, duty.toInt());

  server.send(200);

}

void setup() {

  Serial.begin(9600);

  pinMode(d5Led, OUTPUT);

  WiFi.begin(ssid, password);

  Serial.print("Connecting to ");

  Serial.println(ssid);

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

    delay(500);

    Serial.print(".");

  }

  Serial.println(WiFi.localIP());

  // root

  server.on("/", []() {

    server.send(200, "text/html", myHTML);

  });

  // simple http request route

  server.on("/hello", []() {

    server.send(200, "text/plain", "Hello, this is ESP8266");

  });

  // led routes

  server.on("/led/true", ledOn);

  server.on("/led/false", ledOff);

  server.on("/params", pwmLed);

  server.begin();

}

void loop() {

  server.handleClient();

}

## 3. 3 HTML

<!DOCTYPE html>\

<html>\

<head>\

    <meta name='viewport' content='width=device-width, initial-scale=1' />\

    <script src='https://cdn.tailwindcss.com'></script>\

</head>\

<body class='flex flex-wrap flex-col gap-4 p-8 max-w-xs justify-center m-auto bg-slate-900 text-slate-100'>\

    <input class='border-b border-black block bg-slate-700' placeholder='IP' type='text'\

        onchange='handleIPChange(this.value)' />\

    LED: <input class='w-10 h-10' type='checkbox' onchange='handleRequest(`led/${this.checked}`)' />\

    <input class='bg-slate-500' placeholder='Velocity PWM' type='range' min='0' max='255'\

        onchange='handleRequest(`params?duty=${this.value}`)' />\

    <div class='grid grid-cols-3 gap-4 text-7xl'>\

        <button class='col-start-2 col-end-3' onclick='handleRequest(\"forward\")'>&#x2191;</button>\

        <button class='col-start-1 col-end-2' onclick='handleRequest(\"left\")'>&#x21BA;</button>\

        <button class='col-start-2 col-end-3' onclick='handleRequest(\"stop\")'>&#9982;</button>\

        <button class='col-start-3 col-end-4' onclick='handleRequest(\"right\")'>&#x21BB;</button>\

        <button class='col-start-2 col-end-3' onclick='handleRequest(\"back\")'>&#x2193;</button>\

    </div>\

    <script>\

        var ip = ''; const handleIPChange = (newIP) => {ip = newIP;}; const handleRequest = (order) => fetch(`http://${ip}/${order}`)\

    </script>\

</body>\

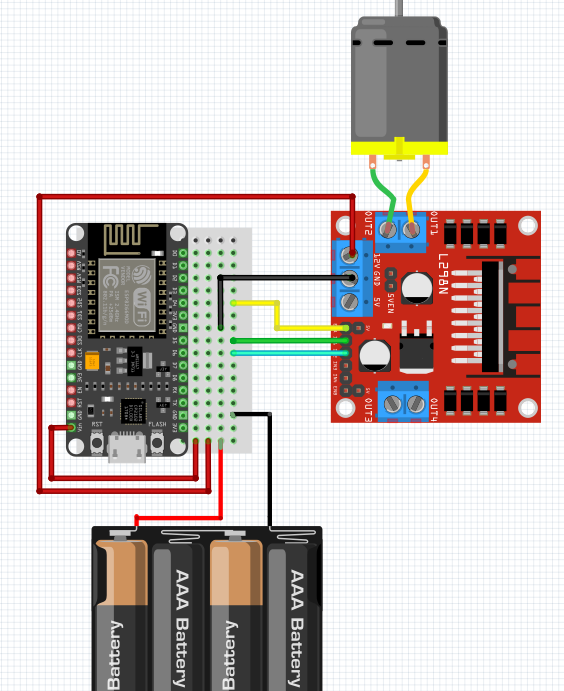
</html>

# . Silnik

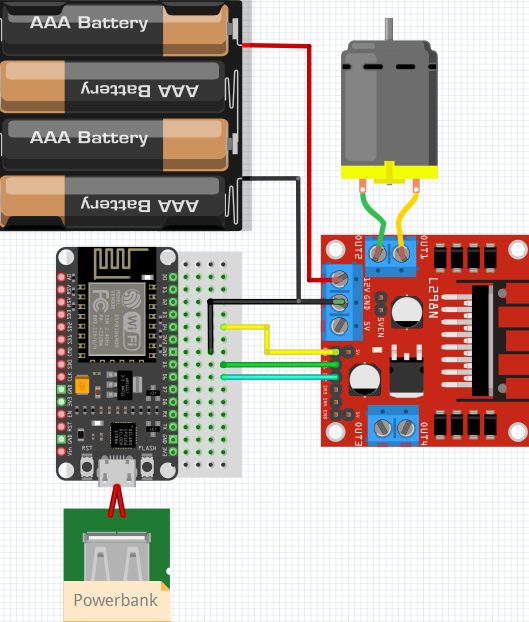
**Cel**: Spowoduj, że silnik będzie obracał się naprzemiennie w obie strony.

## 4. 1 Schemat podłączenia

Propozycja 1 (ze wspólnym zasilaniem, łatwiejsze w rozmieszczeniu na pojeździe)



Propozycja 2 (osobne źródła zasilania, układ będzie działał dłuzej, ale zajmuje więcej miejsca)



## 4. 2 Kod

const int d5in1 = 14;

const int d6in2 = 12;

const int d4enAandB = 2;

void accelerate() {

  digitalWrite(d5in1, HIGH);

  digitalWrite(d6in2, LOW);

  for (int i = 0; i < 255; i++) {

    analogWrite(d4enAandB, i);

    delay(25);

  }

}

void setup() {

  pinMode(d5in1, OUTPUT);

  pinMode(d6in2, OUTPUT);

}

void loop() {

  // clockwise

  digitalWrite(d5in1, HIGH);

  digitalWrite(d6in2, LOW);

  delay(1500);

  // hard stop

  digitalWrite(d5in1, HIGH);

  digitalWrite(d6in2, HIGH);

  delay(1500);

  // counterclockwise

  digitalWrite(d5in1, LOW);

  digitalWrite(d6in2, HIGH);

  delay(1500);

  // soft stop

  digitalWrite(d5in1, LOW);

  digitalWrite(d6in2, LOW);

  delay(1500);

  accelerate();

}

# . Auto - budowa

**Cel:** skonstruuj trójkołowy pojazd z otrzymanych elementów

## 5. 1 Zdjęcia pomocniczne





Widok od dołu.

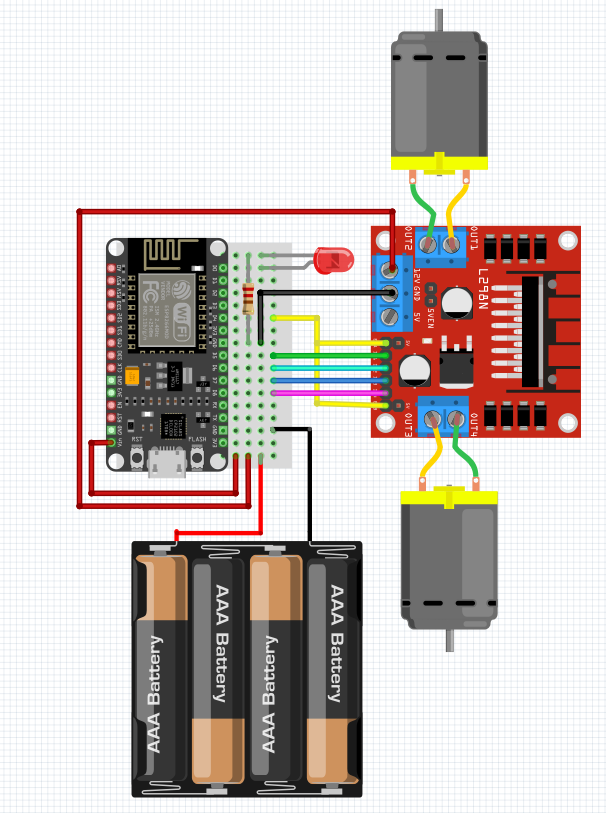




# . Auto - programowanie

**Cel**: Odpal swój własnoręcznie zbudowany pojazd! Steruj z poziomu telefonu (bezprzewodowo)

## 6. 1 Schemat podłączenia



## 6. 2 Kod

#include <ESP8266WiFi.h>

#include "ESP8266WebServer.h"

const char\* myHTML = „ //Tutaj wklej gotowy HTML z punktu 3.3// ”;

const char\* ssid = "YOUR\_SSID";

const char\* password = "YOUR\_PASSW";

ESP8266WebServer server;

const int d0Led = 16;

const int d5in1 = 14;

const int d6in2 = 12;

const int d7in3 = 13;

const int d8in4 = 15;

const int d4enAandB = 2;

void ledOn() {

  digitalWrite(d0Led, HIGH);

  server.send(200);

}

void ledOff() {

  digitalWrite(d0Led, LOW);

  server.send(200);

}

void forward() {

  digitalWrite(d5in1, HIGH);

  digitalWrite(d6in2, LOW);

  digitalWrite(d7in3, HIGH);

  digitalWrite(d8in4, LOW);

  server.send(200);

}

void back() {

  digitalWrite(d5in1, 0);

  digitalWrite(d6in2, 1);

  digitalWrite(d7in3, 0);

  digitalWrite(d8in4, 1);

  server.send(200);

}

void left() {

  digitalWrite(d5in1, HIGH);

  digitalWrite(d6in2, LOW);

  digitalWrite(d7in3, LOW);

  digitalWrite(d8in4, HIGH);

  server.send(200);

}

void right() {

  digitalWrite(d5in1, LOW);

  digitalWrite(d6in2, HIGH);

  digitalWrite(d7in3, HIGH);

  digitalWrite(d8in4, LOW);

  server.send(200);

}

void stop() {

  digitalWrite(d5in1, HIGH);

  digitalWrite(d6in2, HIGH);

  digitalWrite(d7in3, HIGH);

  digitalWrite(d8in4, HIGH);

  server.send(200);

}

void pwmVel() {

  String duty = server.arg("duty");

  Serial.println(duty);

  analogWrite(d4enAandB, duty.toInt());

  server.send(200);

}

void setup() {

  Serial.begin(9600);

  pinMode(d0Led, OUTPUT);

  pinMode(d5in1, OUTPUT);

  pinMode(d6in2, OUTPUT);

  pinMode(d7in3, OUTPUT);

  pinMode(d8in4, OUTPUT);

  WiFi.begin(ssid, password);

  Serial.print("Connecting to ");

  Serial.println(ssid);

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

    delay(500);

    Serial.print(".");

  }

  Serial.println(WiFi.localIP());

  // root

  server.on("/", []() {

    server.send(200, "text/html", myHTML);

  });

  // led

  server.on("/led/true", ledOn);

  server.on("/led/false", ledOff);

  // robot

  server.on("/forward", forward);

  server.on("/left", left);

  server.on("/right", right);

  server.on("/back", back);

  server.on("/stop", stop);

  server.on("/params", pwmVel);

  server.begin();

}

void loop() {

  server.handleClient();

}