

Raport Techniczny: Autonomiczny Pojazd z STM32

# Informacje ogólne

Nazwa projektu: Autonomiczny Pojazd z STM32  
Repozytorium: <https://github.com/cytruseqq/RC-CAR> Zdjęcia oraz nagrania (skrócony link): <https://tiny.pl/ympqzcjr>  
Autorzy:  
 - Magdalena Czyżewska (21227)  
 - Adrian Witów (21319)  
 - Michał Lepak (21255)  
Data rozpoczęcia: 16.03.2025  
Przedmiot: Systemy Wbudowane i Mikrokontrolery

# Cel projektu

Celem projektu jest opracowanie modelu autonomicznego pojazdu sterowanego za pomocą mikrokontrolera STM32.  
Pojazd porusza się w dwóch trybach:  
- Manualnym: Sterowanie przez użytkownika za pomocą interfejsu UART (Bluetooth).  
- Automatycznym: Samodzielne omijanie przeszkód przy wykorzystaniu czujnika.

# Zastosowane podzespoły

|  |  |
| --- | --- |
| Mikrokontroler STM32F3Discovery |  |
| Czujnik ultradźwiękowy do pomiaru odległości HC-SR04 |  |
| Sterownik silników L298N |  |
| 4x zestaw silników elektrycznych DC 3-6V z podwójnym wałem i przekładnią magnetyczną wraz z kołami |  |
| 2x pierścień LED RGB 8 bits |  |
| Wyświetlacz OLED 0,96’ 128x64 |  |
| Koszyk na 6 baterii typu AA |  |
| Czujnik Bluetooth HC-05 |  |
| Baterie AA 6 sztuk |  |
| Przewody: męsko-żeńskie 7 sztuk, 20cm |  |
| Przewody: żeńsko-żeńskie 16 sztuk, 20cm |  |

# Schemat graficzny 1

Obraz zawierający tekst, diagram, Plan, Inżynieria elektroniczna

Zawartość wygenerowana przez AI może być niepoprawna.

# Schemat graficzny 2

Obraz zawierający tekst, diagram, Plan, wykres

Zawartość wygenerowana przez AI może być niepoprawna.

# Opis połączeń

**Połączenia silników z mostkiem H L298N**

• **Silnik 1: L298N -> OUT1, OUT2**

**• Silnik 2: L298N -> OUT1, OUT2**

**• Silnik 3: L298N -> OUT3, OUT3**

**• Silnik 4: L298N -> OUT4, OUT4**

**Wyjaśnienie:**  
OUT1 i OUT2 mostka L298N służą do sterowania jednym kanałem, a więc jednocześnie mogą napędzać dwa równolegle połączone silniki.

**Zasilanie układu przez L298N**

**• Zasilanie bateriami: + -> L298N 12V**

**Wyjaśnienie:**  
L298N potrzebuje zasilania dla sekcji mocy – ta linia 12V dostarcza napięcie potrzebne do zasilenia silników. Dobrze, jeśli napięcie to jest w zakresie 7-12V, co umożliwia sterowanie silnikami DC.

**• Zasilanie bateriami: - -> L298N GND**

**Wyjaśnienie:**  
Podłączenie minusa baterii do masy L298N jest konieczne do zamknięcia obwodu zasilania.

**Zasilanie i uziemienie mikrokontrolera STM32**

**• L298N GND -> STM32 GND**

**Wyjaśnienie:**  
Wspólna masa to podstawa działania układu cyfrowego – umożliwia prawidłowe odniesienie sygnałów logicznych. Bez wspólnej masy STM32 nie rozpozna poziomów logicznych z L298N (np. sygnałów IN1-IN4).

**• L298N 5V -> STM32 5V**

**Wyjaśnienie:**  
L298N posiada stabilizator 5V (jeśli zamontowany zworką „5V\_EN”) – można z niego zasilić STM32, o ile ta akceptuje 5V. W wielu wersjach STM32 zasilanie to idzie do pinów VIN, a logika pracuje na 3.3V (należy sprawdzić dokumentację konkretnego modelu). Ważne: nie należy podłączać 5V bezpośrednio do pinów GPIO 3.3V mikrokontrolera!

**Sterowanie silnikami z STM32**

**• IN1: L298N -> STM32 PC8**

**• IN2: L298N -> STM32 PC9**

**• IN3: L298N -> STM32 PC6**

**• IN4: L298N -> STM32 PC7**

**Wyjaśnienie:**  
Te piny służą do sterowania kierunkiem obrotu silników poprzez sygnały logiczne (0 lub 1). Podanie odpowiedniej kombinacji na IN1/IN2 lub IN3/IN4 powoduje, że napięcie na wyjściach OUT1–OUT4 zmienia kierunek, przez co silnik kręci się w jedną lub drugą stronę. Dobór pinów PCx wynika z wolnych pinów GPIO w projekcie.

**Czujnik odległościowy HC-SR04**

**• HC-SR04 GND -> STM32 GND**

**• HC-SR04 Echo -> STM32 PA1**

**• HC-SR04 Trig -> STM32 PA4**

**• HC-SR04 VCC -> STM32 5V**

**Wyjaśnienie:**

**VCC:** czujnik wymaga 5V do poprawnej pracy.

**Trig:** wyzwala pomiar (krótki impuls HIGH).

**Echo:** po odbiciu sygnału wysyła impuls o długości zależnej od odległości.

**GND:** konieczny do poprawnej pracy sygnałów.

**Moduł Bluetooth HC-05**

**• HC-05 VCC -> STM32 5V**

**• HC-05 GND -> STM32 GND**

**• HC-05 TXD -> STM32 PA3**

**Wyjaśnienie:**

**VCC:** HC-05 działa na 3.6–6V, więc 5V to odpowiednie napięcie.

**TXD (nadaje z HC-05):** powinno trafiać do odbiornika UART w STM32 – pin PA3, jeśli używasz np. USART2 (Rx).

**GND:** wspólna masa wymagana do komunikacji szeregowej.

**Wyświetlacz OLED (I2C)**

**• OLED GND -> STM32 GND**

**• OLED VCC -> STM32 3V**

**• OLED SCL -> STM32 PB6**

**• OLED SDA -> STM32 PB7**

**Wyjaśnienie:**

**VCC 3V:** większość wyświetlaczy OLED I2C (np. SSD1306) może działać na 3.3V i to jest bezpieczne napięcie dla STM32.

**SCL/SDA:** to linie magistrali I2C. PB6 i PB7 to typowe piny I2C1 dla STM32F1 (np. STM32F103).

Konieczna jest też obecność rezystorów pull-up (zwykle 4.7kΩ do VCC), ale często są one już na module OLED.

**Pierścienie LED (WS2812)**

**• LED 5V -> L298N 5V**

**• LED GND -> STM32 GND**

**• LED DI -> STM32 PA8**

**• Pierścień 2: LED DI -> Pierścień 1 LED DO**

**Wyjaśnienie:**

**WS2812** to diody RGB sterowane cyfrowo – DI (data in) to pin wejściowy danych, DO (data out) – wyjściowy.

**Pierwszy pierścień** jest sterowany z PA8 (sygnał cyfrowy z STM32 – ważne, by poziomy były zgodne z napięciem diod, najlepiej 5V z buforem lub konwerterem).

**Drugi pierścień** jest połączony szeregowo – dane przepływają z DO pierwszego pierścienia.

**Zasilanie 5V** jest konieczne do pełnej jasności diod.

# Opis funkcji robota

|  |  |
| --- | --- |
| System sterowania manualnego oraz pomiar odległości od przeszkody |  |
| Aktualizacja animacji LED, aktualizacja wyświetlacza oraz zmiana stanu omijania dla jazdy autonomicznej |  |
| Ustawienie kolorów RGB oraz konwersja z HSV na RGB |  |
| Deklaracje zmiennych |  |

# Przebieg budowy oraz zdjęcia

|  |  |
| --- | --- |
| **Prototyp 1:** Pierwsze zaprojektowanie robota na kawałku kartonu oraz kółek z lego. |  |
| **Prototyp 2:** Zamiana kółek z lego na koła z silnikami, zastosowanie napędu 4x4. | Obraz zawierający koło, Część samochodowa, napój bezalkoholowy, plastik  Zawartość wygenerowana przez AI może być niepoprawna. |
| **Prototyp 3:** Zastąpienie kartonu panelami podłogowymi oraz zastosowanie opasek zaciskowych i kleju na gorąco okazały się skutecznym sposobem na zapewnienie stabilnej struktury robota i solidnego mocowania jego podzespołów. | Obraz zawierający koło, Część samochodowa, opona, zabawka  Zawartość wygenerowana przez AI może być niepoprawna. |
| **Finalny efekt:** Ostatnie zdjęcie przedstawia wizualny efekt robota oraz jego obudowy z płyty pilśniowej. |  |

# Podsumowanie

Projekt budowy autonomicznego pojazdu z wykorzystaniem mikrokontrolera STM32 był dla nas cennym i wymagającym doświadczeniem, łączącym teorię z praktyką. Przeszliśmy przez kilka wersji prototypu – od prowizorycznej konstrukcji z kartonu i LEGO, aż po stabilny model z napędem 4x4 i obudową z druku 3D. Każdy etap pozwalał nam lepiej zrozumieć zarówno zagadnienia związane z elektroniką, jak i aspekty mechaniczne oraz programistyczne.

Poświęciliśmy na ten projekt bardzo dużo czasu – zarówno podczas zajęć, jak i poza nimi – dopracowując konstrukcję, rozwiązując problemy techniczne i testując kolejne funkcjonalności.

Efekt końcowy spełnił nasze oczekiwania – pojazd działa zarówno w trybie manualnym, jak i automatycznym, skutecznie omijając przeszkody. Dużym wyzwaniem okazała się być obudowa, wstępnie miała być wykonana za pomocą druku 3D, natomiast nastąpiło wiele komplikacji z tym związanym. Na przyszłość warto zmienić baterie AA na akumulatory (ogniwa litowo jonowe). Mimo wielu trudności udało nam się zrealizować wszystkie założenia.

Jesteśmy zadowoleni z efektu końcowego i z pełnym przekonaniem możemy powiedzieć, że chętnie podjęlibyśmy się takiego projektu ponownie – być może w jeszcze bardziej zaawansowanej formie.

# Licencja

Projekt jest udostępniony na licencji MIT, co oznacza, że można go dowolnie używać, kopiować, modyfikować i rozpowszechniać, pod warunkiem dołączenia informacji o oryginalnych autorach.