

ARCHITEKTURA SYSTEMÓW KOMPUTEROWYCH 2024

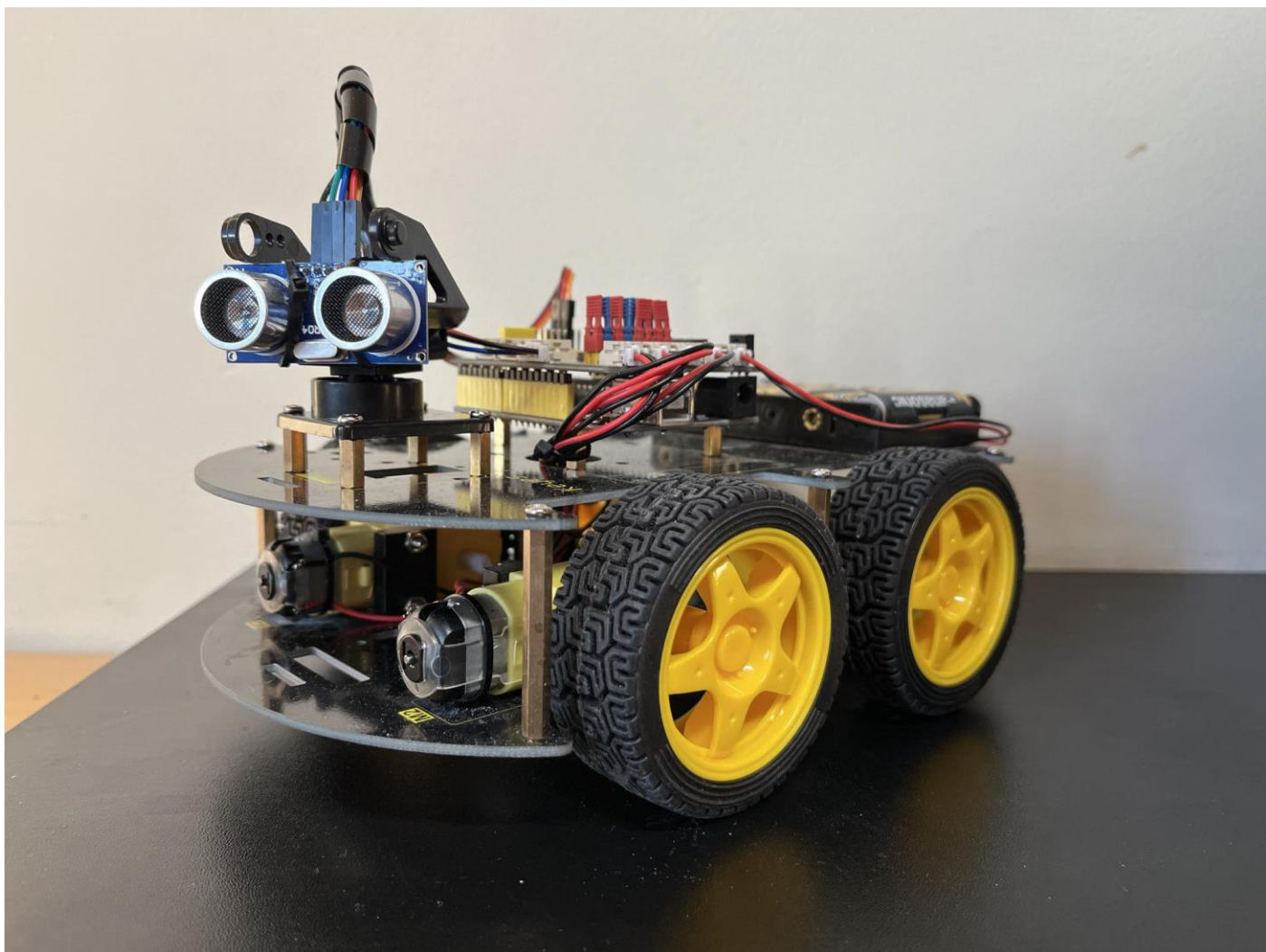
prowadzący: dr inż. RAFAŁ KLAUS

ROBOT UNIKAJĄCY PRZESZKODY

drużyna warsztatowa nr 6

Paweł Kolec 155873 L4

Adam Nowacki 155838 L4



Lp.	autorzy	indeks	nr grupy dziekańskiej	Zadania
1	Adam Nowacki	155838	4	1. Zakupy 2. Robienie zdjęć i nagrań 3. Składanie 4. Programowanie
2	Paweł Kolec	155873	4	1. Zakupy 2. Pomysł na robota 3. Składanie 4. Montaż filmu 5. Dokumentacja

Poznań, 2024

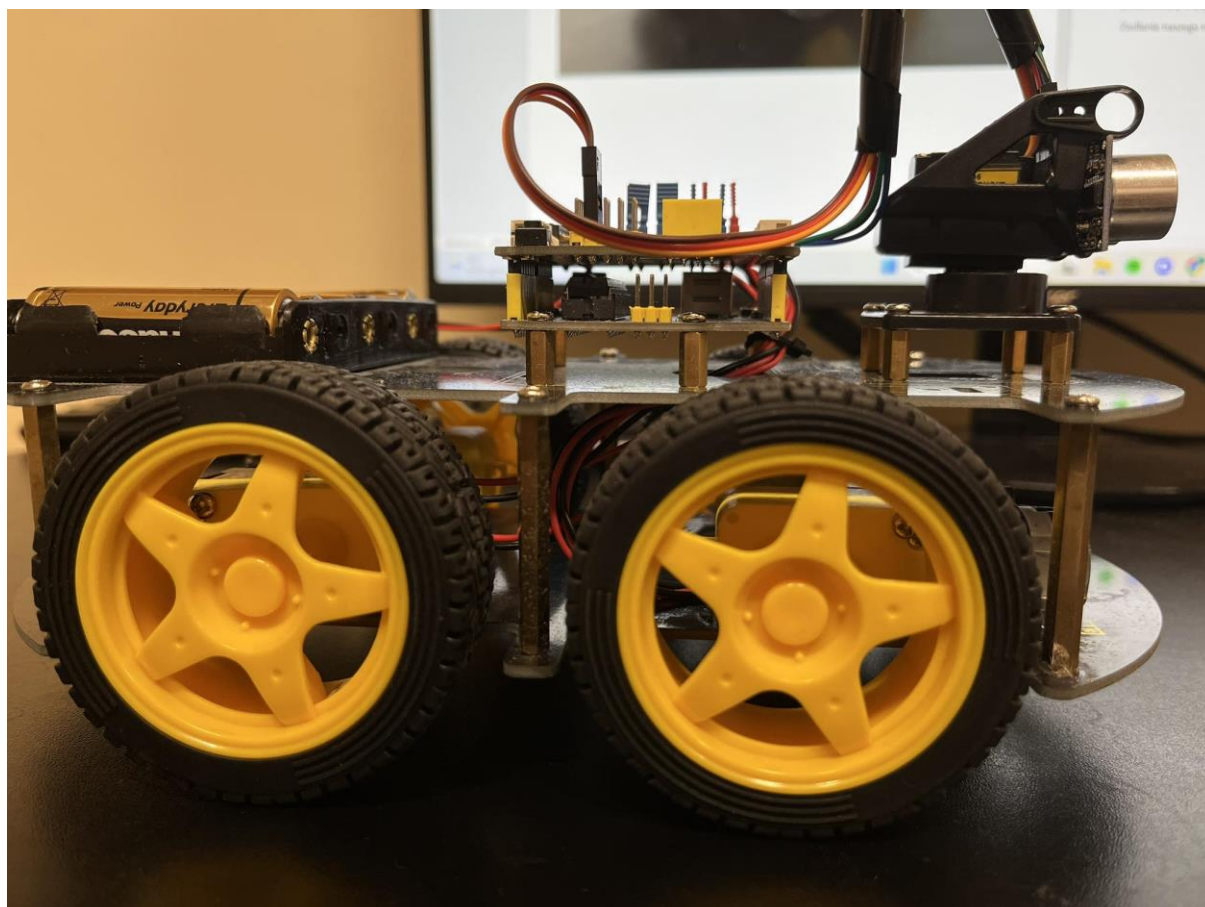
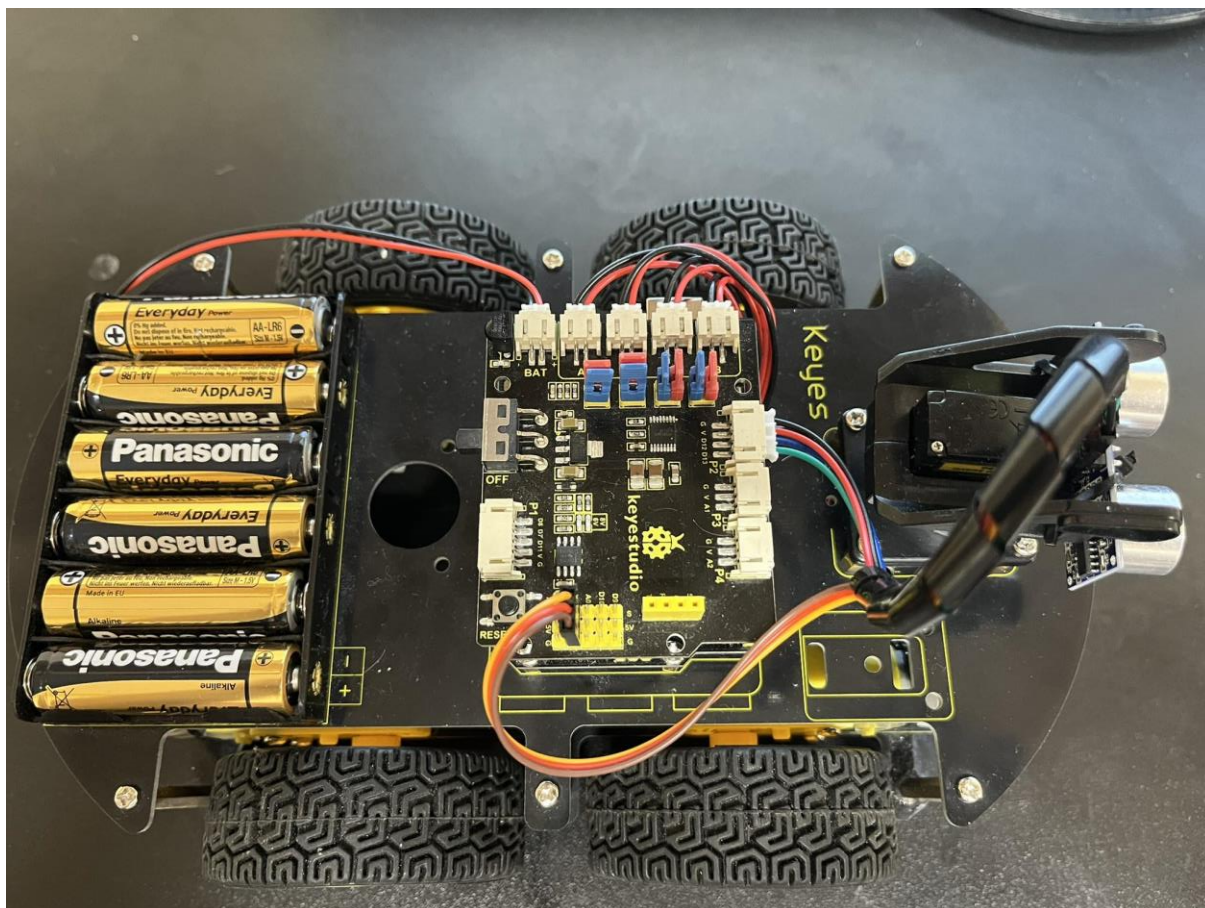
1. Wprowadzenie, specyfikacja wymagań

Robot, jakiego postanowiliśmy stworzyć to robot, którego zadaniem będzie omijanie napotkanych przeszkód. Będzie poruszać się, zatrzymując się tylko wtedy, gdy napotka przeszkodę. Zasada działania opiera się na wykorzystaniu czujnika ultradźwiękowego HC-SR04. Gdy wykryje on przeszkodę, nasz robot zatrzymuje się i uruchamia mechanizm Servo, który pozwala mu skanować obszar po obu stronach. Na podstawie analizy odległości do najbliższej przeszkody podejmuje decyzję, w którą stronę skręcić. Dzięki temu a nasz robot porusza się po zamkniętej przestrzeni, unikając kolizji.

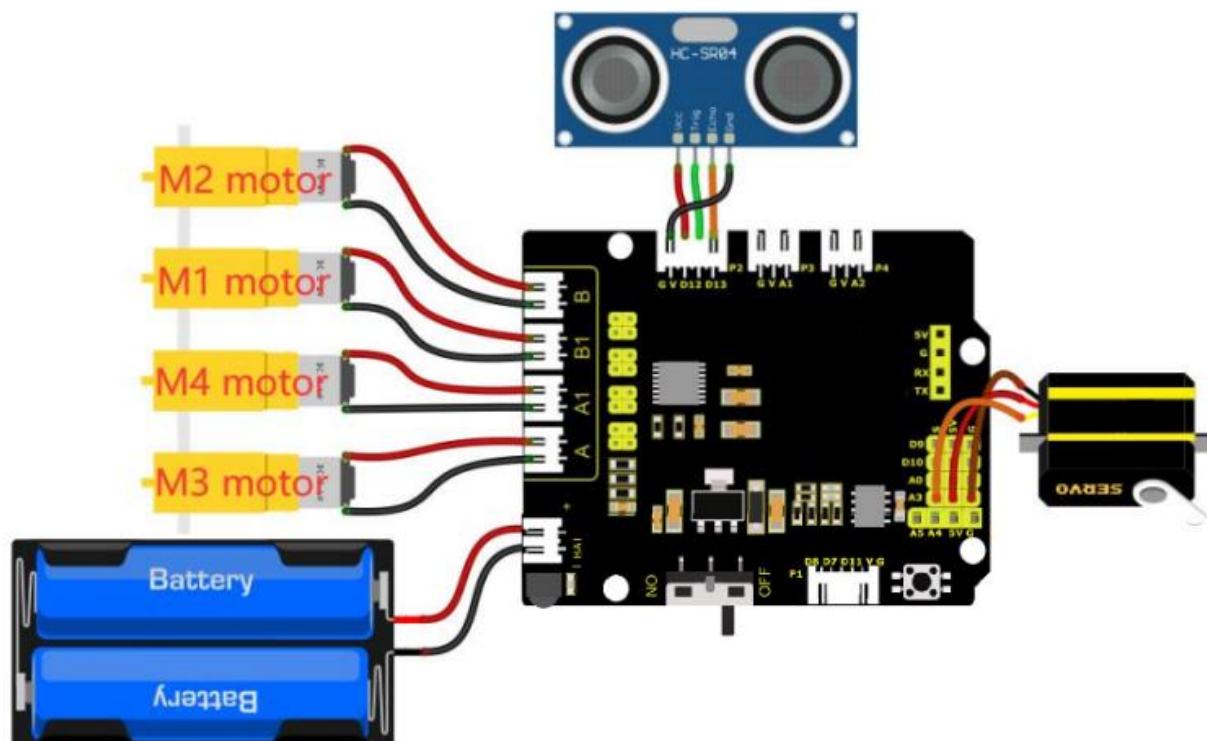
2. Rozwiązania techniczne, opis konstrukcji

Do budowy urządzenia postanowiliśmy wykorzystać mikrokontroler Arduino Uno wraz dodatkowym shieldem Motor Driver Expansion Board. Do dolnej płytki przymocowaliśmy cztery silniki z dołączonymi do nich kołami, zaś na górnej płytce umieściliśmy Arduino, czujnik ultradźwiękowy oraz baterie AA w ilości 6 sztuk. Arduino oraz czujnik ultradźwiękowy zostały zamocowane na dystansach dla większej stabilności. Czujnik ultradźwiękowy został przymocowany do mechanizmu Servo, który odpowiada za obrót czujnika. Silniki zostały przykręcone do dolnej płytki za pomocą śrub. Kable zostały poprowadzone przez otwory w płytkach.

Zasilanie naszego robota oparte jest na 6 bateriach AA o napięciu 1,5V.



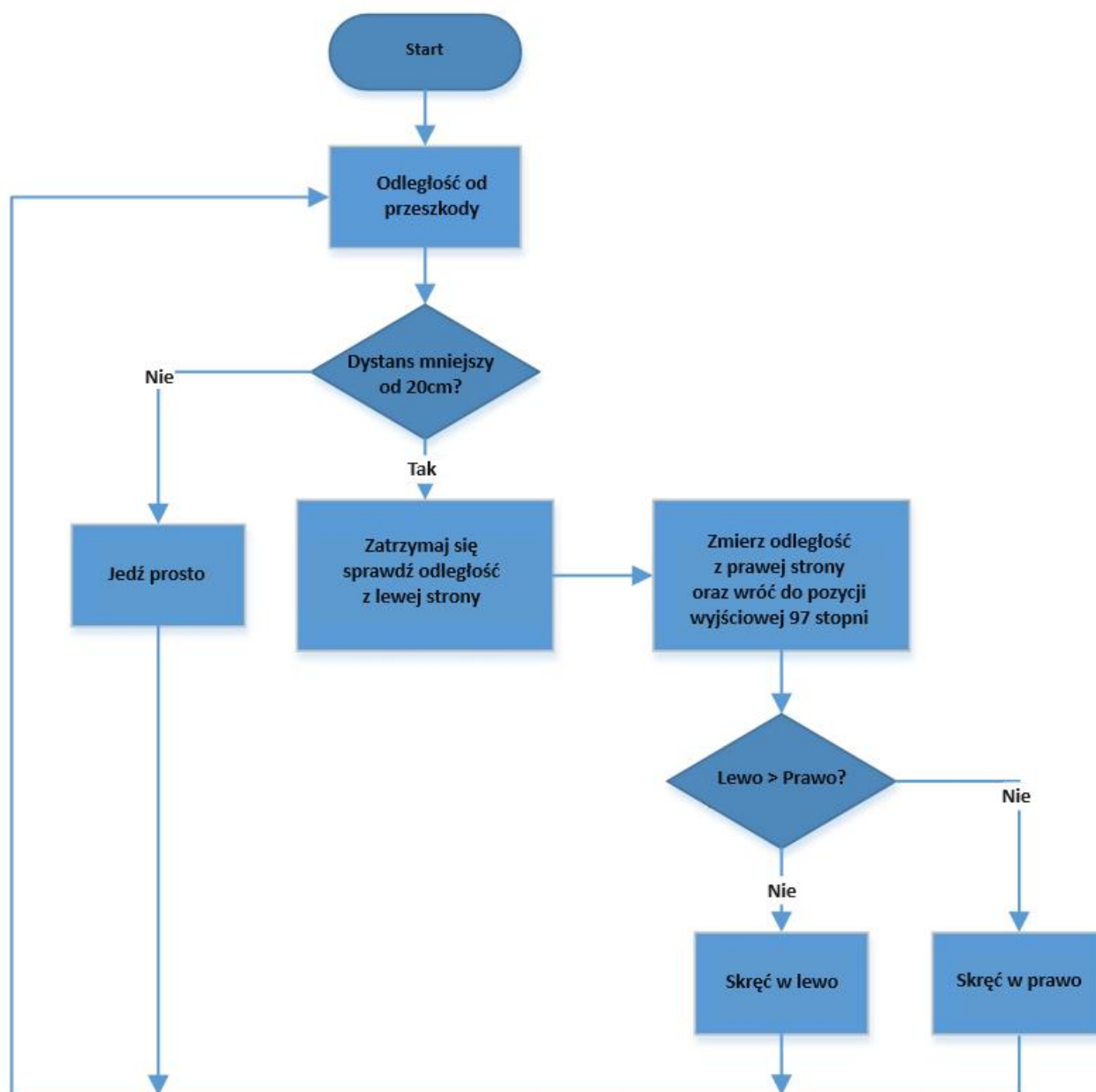
3. Schemat i opis



Schemat połączeń

Do sterownika Motor Driver Shield podłączone są:

- Cztery silniki podłączone do portów A, A1, B, B1
- Ultradźwiękowy czujnik do portów G, V, D12, D13
- Servo mechanizm do portu A3
- Zasilanie bateriami do portu BAT



Schemat blokowy programu

4. Oprogramowanie

Do zaprogramowania robota posłużyliśmy się środowiskiem Arduino IDE 2.3.2, w którym pisaliśmy nasze oprogramowanie dla robota. Język programowania użyty przez nas to c++. Biblioteka jakiej użyliśmy do obsługi czujnika ultradźwiękowego to SR04.h (<https://github.com/mrRobot62/Arduino-ultrasonic-SR04-library>).

Opis Programu

Funkcja avoid()

```
void avoid()
{
    distance=sr04.Distance();

    if((distance < 20)&&(distance != 0))
    {
        car_Stop();
        delay(1000);
        servopulse(servopin,167);
        delay(500);
        a1=sr04.Distance();
        delay(100);
        servopulse(servopin,27);
        delay(500);
        a2=sr04.Distance();
        delay(100);
        servopulse(servopin,97);
        delay(500);
        if(a1 > a2)
        {
            car_left();
            servopulse(servopin,97);
            delay(700);
        }

        {
            car_right();
            servopulse(servopin,97);
            delay(700);
        }
    }
    else
    {
        car_front();
    }
}
```

Rozpoczynamy od wywołania funkcji avoid().

Wewnątrz funkcji, odcytujemy wartość odległości za pomocą czujnika ultradźwiękowego, sprawdzamy warunek czy odległość jest mniejsza niż 20 cm i czy jest różna od zera.

Jeśli warunek jest spełniony (czyli samochód wykrywa przeszkodę blisko), następują następujące kroki:

- Zatrzymujemy samochód za pomocą funkcji car_Stop() i oczekujemy przez 1 sekundę.
- Obracamy serwomechanizm w jedną stronę na 167 stopni, odczekujemy 500 ms, mierzymy odległość za pomocą czujnika ultradźwiękowego, a następnie obracamy serwomechanizm w drugą stronę na 27 stopni, odczekujemy 500 ms i ponownie mierzymy odległość.
- Zwracamy serwomechanizm do pozycji wyjściowej (97 stopni).

d. Porównujemy odczytane odległości: jeśli odległość na lewo jest większa niż na prawo ($a_1 > a_2$), to skręcamy w lewo za pomocą funkcji `car_left()` i serwomechanizm ustawiamy na pozycji 90 stopni, czekając przez 700 ms (`delay(700)`); w przeciwnym razie skręcamy w prawo za pomocą funkcji `car_right()` i ponownie ustawiamy serwomechanizm na pozycji 90 stopni, czekając przez 700 ms.

Jeśli warunek początkowy nie jest spełniony (brak przeszkody blisko), to samochód kontynuuje jazdę do przodu za pomocą funkcji `car_front()`.

Ten kod umożliwia samochodowi unikanie przeszkód poprzez zatrzymanie się, obracanie czujnika w celu wykrycia wolnej przestrzeni i skręcanie w kierunku większej odległości.

Funkcje sterujące kierunkiem ruchu robota

```
void car_front()
{
    digitalWrite(left_ctrl,HIGH);
    analogWrite(left_pwm,155);
    digitalWrite(right_ctrl,HIGH);
    analogWrite(right_pwm,155);
}
void car_back()
{
    digitalWrite(left_ctrl,LOW);
    analogWrite(left_pwm,100);
    digitalWrite(right_ctrl,LOW);
    analogWrite(right_pwm,100);
}
void car_left()
{
    digitalWrite(left_ctrl, LOW);
    analogWrite(left_pwm, 0);
    digitalWrite(right_ctrl, HIGH);
    analogWrite(right_pwm, 155);
}
void car_right()
{
    digitalWrite(left_ctrl, HIGH);
    analogWrite(left_pwm, 155);
    digitalWrite(right_ctrl, LOW);
    analogWrite(right_pwm, 0);
}
void car_Stop()
{
    digitalWrite(left_ctrl,LOW);
    analogWrite(left_pwm,0);
    digitalWrite(right_ctrl,LOW);
    analogWrite(right_pwm,0);
}
```

Te funkcje działają na zasadzie sterowania silnikami samochodu za pomocą sygnałów cyfrowych oraz modulacji szerokości impulsów (PWM) do regulacji prędkości obrotowej silników.

Funkcje `car_front()` i `car_back()` sterują obydwojema silnikami w taki sposób, aby samochód poruszał się odpowiednio do przodu lub do tyłu. W tym celu ustawiają odpowiednie piny sterujące silnikami na wysoki lub niski poziom logiczny (HIGH lub LOW), a następnie używają funkcji `analogWrite()` do regulacji prędkości obrotowej silników poprzez sygnały PWM.

Funkcje `car_left()` i `car_right()` powodują obrót samochodu w lewo lub w prawo. Realizują to poprzez wyłączenie jednego z silników i ustawienie go w trybie "wstecz", co powoduje obrót w odpowiednim kierunku, podczas gdy drugi silnik nadal pracuje normalnie, co zapewnia ruch w przód.

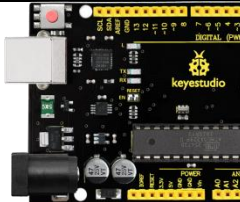
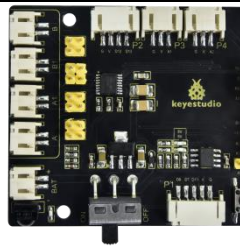


Funkcja `car_Stop()` zatrzymuje ruch samochodu poprzez wyłączenie obu silników, co powoduje zatrzymanie ich obrotów.

Sterowanie serwomechanizmem

```
void servopulse(int servopin,int myangle)
{
    for(int i=0; i<20; i++)
    {
        int pulsewidth = (myangle*11)+500;
        digitalWrite(servopin,HIGH);
        delayMicroseconds(pulsewidth);
        digitalWrite(servopin,LOW);
        delay(20-pulsewidth/1000);
    }
}
```

Ta funkcja służy do sterowania serwomechanizmem poprzez generowanie impulsów PWM. W pętli wykonywane są obliczenia szerokości impulsu na podstawie żadanego kąta. Następnie wysyłany jest impuls o odpowiedniej szerokości, który steruje serwomechanizmem, umożliwiając precyzyjne ustawienie jego kąta obrotu.

5. Oprogramowanie

Lp	Część	Ilość sztuk	Zdjęcie	Gdzie kupiona	Cena
1	Keyestudio V4.0 Board(UNO compatible)	1		Aliexpress	32,88
2	Keyestudio Motor Driver Expansion Board	1		Aliexpress	28,93
3	Czujnik ultradźwiękowy	1		Botland	8,90
4	Servo mechnizm	1		Botland	12,70

5	Silniki	4		Aliexpress	18,76
6	6 AA Uchwyt baterii	1		Aliexpress	4,22
7	Mechanizm obrotu	1		Aliexpress	3,87
8	Podstawa górna	1		Aliexpress	7,89
9	Podstawa dolna	1		Aliexpress	7,89
10	Koła	4		Aliexpress	13,65
11	M3*10MM dystanse	10		Zapasy domowe	0,00
12	M3*40MM dystanse	4		Zapasy domowe	0,00
13	M3*30MM śruby	8		Zapasy domowe	0,00
14	M3*6MM śruby	40		Zapasy domowe	0,00

15	M3 nakrętki	16		Zapasy domowe	0,00
16	M2*8MM śruby	6		Zapasy domowe	0,00
17	M3*8MM śruby	4		Zapasy domowe	0,00
18	M2 nakrętki	6		Zapasy domowe	0,00
19	M3*10MM śruby	3		Zapasy domowe	0,00
20	Kabel USB	1		Zapasy domowe	0,00

Łączne koszty budowy robota wyniosły 139,69 zł.

6. Oprogramowanie

Składanie robota nie przyniosło nam większych problemów. Przy pisaniu oprogramowania w środowisku Arduino natknęliśmy się na problem związany z kompatybilnością bibliotek dla naszego czujnika ultradźwiękowego. Po przeszukaniu internetu, odnaleźliśmy w końcu prawidłową bibliotekę, która współdziałała z naszym czujnikiem. Z innych rzeczy związanych z oprogramowaniem nie mieliśmy większego problemu.