AKADEMIA GÓRNICZO-HUTNICZA

Dokumentacja do projektu

Biblioteka do sterowania pojazdem czterokołowym

z przedmiotu

Języki programowania obiektowego

Elektronika i Telekomunikacja, 3 rok

Antoni Bajor

Gr 3, Piątek 15:00

prowadzący: Jakub Zimnol

09.01.2025

1. Uruchomienie programu

Oryginalnie biblioteka powstała do obsługi robota budowanego na potrzeby Design Labu. Do uruchomienia programu potrzebna jest płytka typu ESP32, a najlepie wersja j ESP32 WROVER-E. Oprócz tego potrzebne jest środowisko ESP-IDF, które można pobrać np. w Visual Studio Code. Zaimplementowaną bibliotekę można przetestować wgrywając na płytkę program znajdujący się tutaj: https://github.com/filizw/esp32-cam-mobile-robot.

2. Działanie biblioteki

Przy pisaniu kodu obsługującego pojazd, mamy do wyboru 2 klasy: Vehicle4WD, która pozwala na skręcanie pojazdu poprzez przeciwne kręcenie kołami, oraz Vehicle4WDSteering, która zmienia sposób skrętu na taki używajcy kół skrętnych, obsługiwanych przez serwa.

Przy tworzeniu obiektu klasy *Vehicle4WD* możemy podać początkowe parametry w takiej kolejności: **kierunek kręcenia się kół lewych**, **kierunek kręcenia się kół prawych**, **prędkość**. Natomiast przy tworzeniu obiektu klasy *Vehicle4WDSteering* możemy podać parametry **kierunek kręcenia się wszystkich kół**, **prędkość** i **promień skrętu**.

Głównymi funkcjami biblioteki są:

driveForward() - która przyjmuje czas w milisekundach, przez którą pojazd będzie będzie jechał do przodu, jeśli przekażemy 0 będzie on jechał bez końca.

driveBackwards() - która działa tak jak funkcja wcześniejsza ale pojazd jedzie do tyłu.

stop() – zatrzymuje pojazd

turnLeft() i **turnRight()** które przyjmują czas w milisekundach przez który pojazd będzie kręcił się w miejscu, w stronę odpowiednio lewą lub prawą, jeśli jest obiektem klasy *Vehicle4WD*. Natomiast w przypadku klasy *Vehicle4WDSteering* te funkcje są przeciążone i możemy przekazać im kąt w stopniach, o jaki mają skręcić się koła w odpowiednim dla funkcji kierunku. Zakres skrętu wynosi od 0° do 40° w obie strony, a więc w sumie od -40° do 40° w stosunku do kół prostych.

setSpeed() – pozwala na ustawienie prędkości pojazdu. Możliwy zakres to od 0 do 100% (0-100). Natomiast polecanym zakresem jest od 40% do 100%, ponieważ przy niższych wartościach niektóre pojazdy mogą w ogóle nie ruszyć. Dodatkowo używając skrętu z klasy *Vehicle4WD* polecam używać 100% prędkości, ponieważ przez duży opór kół, potrzebna jest większa moc napędu.

getSpeed() – zwraca prędkość

getDirection() – zwraca wektor zawierający 2 elementy: kierunek kręcenia się kół lewych i prawych. 0 – koła się nie kręcą, 1 – kręcą się do przodu, 2 – kręcą się do tyłu.

getTurnAngle() – z klasy *Vehicle4WDSteering* zwraca aktualny kąt skrętu kół

makeSquare() i **makeCircle()** – przykładowe funkcje które mogą pokazać jak działa pojazd bez dodatkowego sterowania, przyjmą 2 wartości: pierwszą która ozacza w którą strone zostanie wykonana figura 0 – lewo, 1 – prawo, i drugą która ozacza czas w milisekundach, wykonywania figury.

Wykorzystane piny GPIO:

2 – Dioda LED

25, **26**, **32**, **33** – sterowanie kierunkiem obrotu kół

14, **27** – sygnał PWM do sterowania prędkością silników

12 – sygnał PWM do sterowania serwami

Piny można zmienić w kodzie w razie potrzeby.