

Projekt na garaż złożoności – odkurzacz omijający przeszkody Arduinold

Ksymena Poradzisz, Maciej Kucab

27 luty 2023

1 Streszczenie

W ramach projektu zaliczeniowego chcieliśmy skonstruować robota omijającego przeszkody, który jednocześnie odkurza i jest w stanie odtwarzać dźwięki. Do odkurzania wykorzystaliśmy wiatrak, który kręci się "w przeciwną stronę", czyli zasysa powietrze zamiast je wydmuchiwać. Układ porusza się dzięki sterownikowi L298n, który steruje 4 silnikami z kołami. Obsługa dźwięku (która się nie udała) miała się odbywać za pomocą shielda cytron EzMP3. Projekt jest jeszcze w fazie testowania i jego głównym problemem jest zbyt słabe zasilanie. W paragrafie 2 opisujemy potrzebne elementy, których kompletaacja jest potrzebna do zrekonstruowania robota. W paragrafie 3 opisujemy sposób, w jaki projekt został zbudowany, natomiast w paragrafie 4 opisujemy stan projektu, czyli jak działa oraz planowane ulepszenia. Ostatnim paragrafem jest podsumowanie raportu.

2 Wykorzystane rozwiązania naukowo—techniczne

Do wykonania projektu wykorzystaliśmy:

- **Drukarka 3d** – za pomocą drukarki wydrukowaliśmy całą obudowę na robota oraz wiatraczek. Podstawkę oraz stół zrobiliśmy samodzielnie w programie Tinkercad, natomiast wiatraczek znaleźliśmy w internecie [1]. W programie Blender został on nieznacznie zmodyfikowany. Dodatkowo, wydrukowaliśmy element w kształcie litery L, który posłużył nam do obracania ultradźwiękowym czujnikiem odległości.
- **Elektronika** – do wykonania projektu wykorzystaliśmy płytkę Arduino Uno wraz z shieldem Cytron EzMP3 i płytkę prototypową (małą). Oprócz płytek konieczny jest sterownik (my wykorzystaliśmy L298N), serwomechanizm (który ma możliwość wykonania 180 stopni obrotu) oraz ultradźwiękowy czujnik odległości (HC-SR04). Wykorzystaliśmy również 4 silniki (DC 5V z kołem) oraz 1 silnik bez koła (MT64 3—6 V). Dodatkowo wykorzystaliśmy 1 koszyczek na 4 baterie AAA oraz 1 koszyczek na 3 baterie AA (wraz z bateriami). Oczywiście do tego wszystkiego potrzebne są kable oraz lutownica ze spoiwem.
- Dodatkowymi elementami, które zostały zakupione są pojemnik z pepco (5zł) oraz końcówka do odkurzacza (była w zestawie). Wykorzystaliśmy również element lejkopodobny (znaleziony w domu), taśmę izolacyjną, klej na gorąco i taśmę grubą z materiałem wypełniającym rurę (był na garażu i wyglądał jak podkładka na mysz, tylko 50 razy większa).

Konieczny jest również kabel łączący arduino z komputerem, kabel łączący arduino z koszykiem oraz ew. do grania muzyki (które się nie udało) potrzebujemy kartę microSD lub stick USB oraz głośnik. Do zrobienia dziur wykorzystywaliśmy wiertarkę z garażu.

3 Opis projektu

Projektem jest robot zwany Arduinold, którego zadaniem jest jeżdżenie po pomieszczeniu, unikanie przeszkód, jednoczesne odkurzanie i wydawanie dźwięków. Budowanie robota zaczęliśmy od wydruku koniecznych elementów – płaskiej płyty o wymiarach 19 x 12 x 0.5 cm oraz "stolika", którego podstawa ma takie same wymiary, natomiast nogi mają ok. 12 cm. W międzyczasie konstruowaliśmy układ – kable z silników z kołami zostały przylutowane "do siebie nawzajem" w taki sposób, aby 2 lewe koła miały wspólny kabel czerwony (z napięciem) i wspólny czarny (z uziemieniem). W ten sposób sterownik obsługuje jednocześnie dwa koła z jednej strony robota (lewej) i z drugiej (prawej). Następnie koła przykleiliśmy do wydrukowanej podstawki klejem na gorąco. Później nogi (również klejem na gorąco) przykleiliśmy do podstawki z drugiej strony, tworząc tym samym klatkę na pojemnik z pepco. W wieczku pojemnika robimy dziurę wielkości silnika oraz w pojemniku dziurę największą możliwą od strony podstawy pudełka. Do silnika przyklejamy (klejem na gorąco) wiatraczek wydrukowany w drukarce 3D. Silnik umieszczamy w dziurze na silnik, nie jest konieczny klej gdyż trzyma go tarcie. Do wystającego końca silnika lutujemy dwa kable. Do drugiej strony pojemnika (kiedyś podstawki) przyklejamy taśmą element lejkopodobny. Do tego elementu wsadzamy materiał, który ma emitować rurę i obklejamy go taśmą, natomiast do końca tego materiału przyklejamy końcówkę od odkurzacza. Dobrze by było, aby rura znajdowała się na odpowiedniej wysokości. Taki układ wsadzamy do środka klatki i przyklejamy taśmą w celu stabilizacji ale również nie przyklejamy go na stałe, aby móc jeszcze wprowadzić ewentualne poprawki. Na górę układu przymocowujemy elektronikę. My przymocowaliśmy arduino oraz sterownik taśmą, aby móc później użyć go do czegoś innego, natomiast serwomechanizm i płytkę prototypową przykleiliśmy. Do ostrza serwomechanizmu przyklejamy element w kształcie litery L, natomiast do końca tego elementu przyklejamy HC-SR04. Podłączyliśmy kable tak, jak to jest pokazane na schemacie Poniżej prezentujemy zdjęcie robota oraz jego schemat.

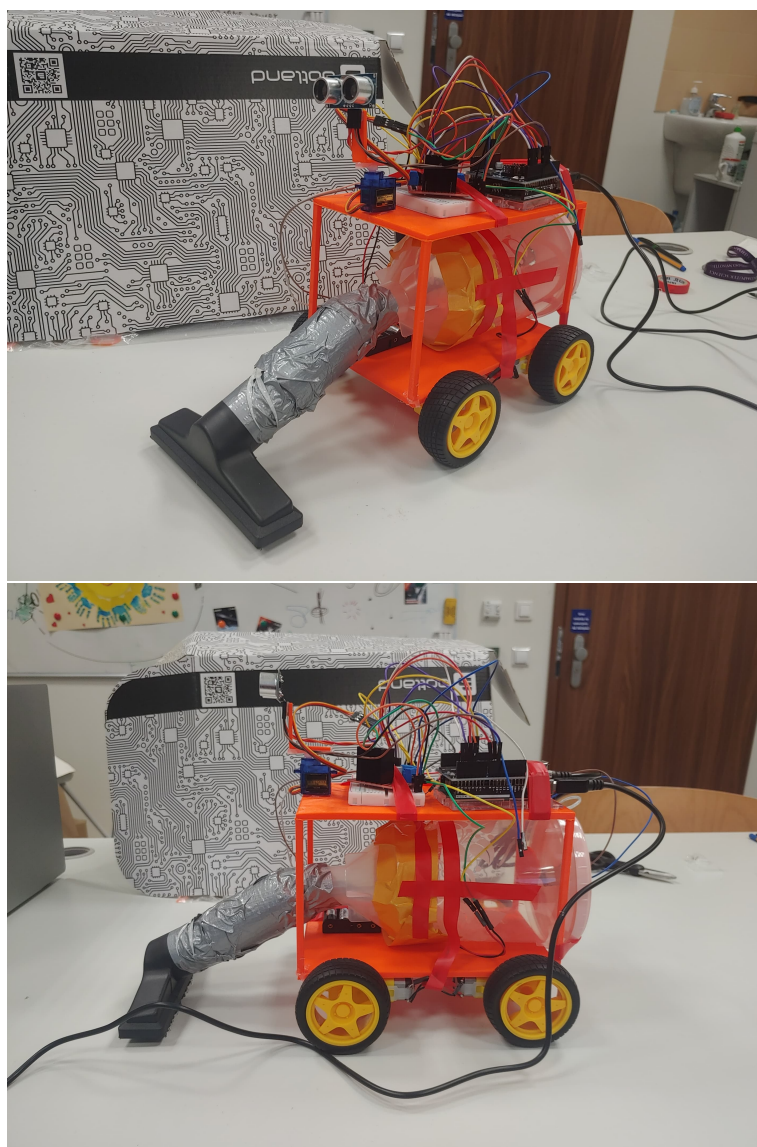


Figure 1 – Zdjęcia robota jako efekt końcowy. Na zdjęciach widać w jaki sposób zostały przyłączone kable silników i o co chodzi z tym, że 2 koła są przez sterownik widziane jako jedno.

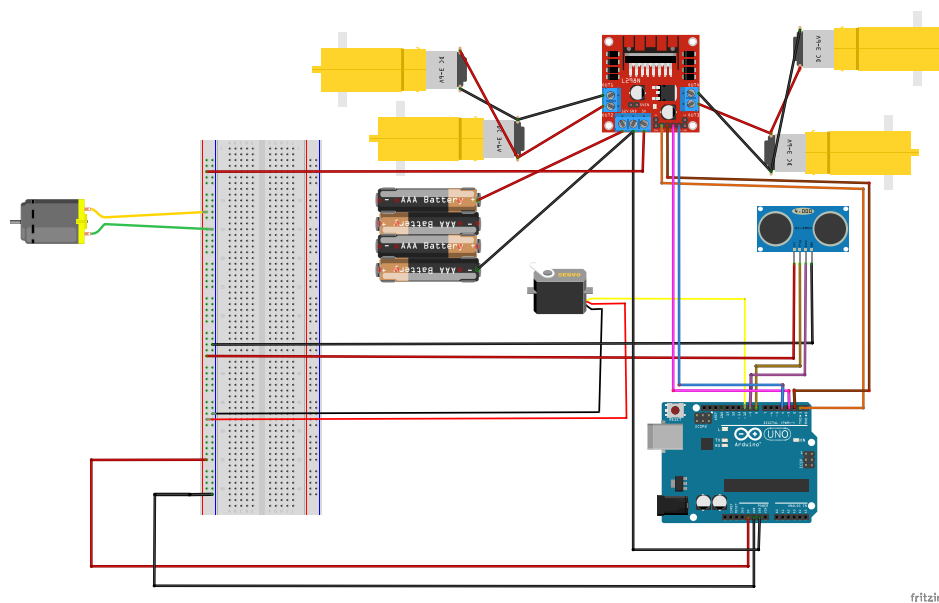


Figure 2 – Schemat układu obsługujący Arduinolda. Wykonany w programie fritzing.

Poniżej prezentujemy kod napisany w C obsługujący robota. Do kodu należy dodać jedną uwagę – gdyby układ miał więcej mocy, odkomentowane zostałyby linie w funkcjach opisujących skręt w lewo oraz skręt w prawo (jedyne komentarze z symbolami `\\`).

```
#include <Servo.h>
/*#include <CytronEZMP3.h>*/

int servotrig = 8;
int servoecho = 9;

int silniklewy1 = 1;
int silniklewy2 = 2;
int silnikprawy1 = 3;
int silnikprawy2 = 4;

float droga;
float drogalewy;
float drogaprawy; /* cm */

Servo serwomechanizm;

void setup() {
    serwomechanizm.attach(10);
```

```

    pinMode(silniklewy1,OUTPUT);
    pinMode(silniklewy2,OUTPUT);
    pinMode(silnikprawy1,OUTPUT);
    pinMode(silnikprawy2,OUTPUT);
}

float Droga(int detektor, int testowy)
{
    pinMode(testowy, OUTPUT);
    digitalWrite(testowy, LOW);
    delay(2); /*odswiez*/

    digitalWrite(testowy, HIGH);
    delay(10);
    digitalWrite(testowy, LOW); /*mignij*/

    pinMode(detektor, INPUT); /*deklaracja pomiaru czasu*/

    return pulseIn(detektor,HIGH) * 0.017;
    /*s = v*t/2, v/2 = 0.0172 cm/s*/
}

void moveforward( int t)
{
    digitalWrite(silniklewy1,HIGH);
    digitalWrite(silniklewy2,LOW);
    digitalWrite(silnikprawy1,HIGH);
    digitalWrite(silnikprawy2,LOW);
    delay(t); /*t w milisekunach*/
}

void moveback(int t)
{
    digitalWrite(silniklewy1,LOW);
    digitalWrite(silniklewy2,HIGH);
    digitalWrite(silnikprawy1,LOW);
    digitalWrite(silnikprawy2,HIGH);
    delay(t); /*t w milisekunach*/
}

void moveleft( int t)
{
    digitalWrite(silnikprawy1,LOW);
    digitalWrite(silnikprawy2,HIGH);
    //digitalWrite(silnikprawy1,HIGH);

```

```

//digitalWrite(silnikprawy2,LOW);
delay(t); /*t w milisekunach*/

}

void moveright(int t)
{

//digitalWrite(silniklewy1,HIGH);
//digitalWrite(silniklewy2,LOW);
digitalWrite(silniklewy1,LOW);
digitalWrite(silniklewy2,HIGH);
delay(t); /*t w milisekunach*/

}
void stop(int t)
{
digitalWrite(silniklewy1,LOW);
digitalWrite(silniklewy2,LOW);
digitalWrite(silnikprawy1,LOW);
digitalWrite(silnikprawy2,LOW);
delay(t);
}

void loop() {
    serwomechanizm.write(90);
    droga = Droga(servoecho, servotrig); /*mierzymy droge.*/
    Serial.println(droga); /*wypluj droge do konsoli. */
    if (droga > 25)
    {
        moveforward(350);
    }
    else
    {
        stop(300); /*zatrzymaj sie*/
        moveback(800);
        stop(100);
        serwomechanizm.write(35); /*obracaj w lewo*/
        delay(500);
        drogalewy = Droga(servoecho, servotrig);
        serwomechanizm.write(145); /*obroc serwo w 2 strone*/
        delay(500);
        drogaprawy = Droga(servoecho, servotrig);
        serwomechanizm.write(90);
        delay(500);
        if (drogalewy > drogaprawy)
        {
            moveleft(2000);
        }
    }
}

```

```

    }
    else
    {
        moveright(2000);
    }
}
}

```

W następnym paragrafie opisujemy co robi robot, czego nie robi i jak to naprawić.

4 Stan pracy nad projektem

Robot w poprawny sposób rozpoznaje przeszkody oraz w poprawny sposób próbuje je ominąć. Jedyne zastrzeżenie jest takie, że robot rozpoznaje je na pewnej (dużej) wysokości, więc może nie wykryć np. małego pudełka na drodze. Nie jest to jednak dużym problemem, gdyż jego głównym zadaniem jest niewjeżdżanie w ściany.

Robot poprawnie próbuje ominąć przeszkodę. Problemem jest jednak zbyt mała moc podana na silniki, co skutkuje (razem z ciężarem robota) że skręcanie wychodzi mu bardzo nie płynnie, a czasami nawet wcale. Kod został napisany tak, aby robot obracał się "jak czołg" (czyli gdy chce jechać w lewo, to lewe koła obracają się do tyłu, a prawe do przodu) jednak przez słabą moc podawaną na silniki robot w takim przypadku stoi w miejscu. Dlatego kod został zmieniony tak, że tylko lewe lub tylko prawe koła wykonują ruch obrotowy. Wtedy skręcanie wykonuje się częściej.

Robot poprawnie obraca silnikiem z wiatraczkiem, jednak również ma za małą moc, by móc jednocześnie obracać nim oraz jechać. Dlatego silnik z wiatraczkiem jest odłączony. Dodatkowo, konieczny jest większy silnik, do którego mamy już wydrukowany większy wiatrak, aby było czuć "ssanie". Teraz niestety przez rurę go nie czuć, chociaż bez rury podmuch powietrza jest wyczuwalny. Nie możemy jednak dać większego wiatraczka, gdyż jest on zbyt ciężki i zbyt duży na ten silnik, i cały układ zaczyna się trząść i skakać. Robot nie mówi, nie udało nam się zaimplementować biblioteki do odtwarzania dźwięku. Główny powód jest taki że robot już i tak ma bardzo mało mocy z baterii, więc dokładanie kolejnych funkcji na tym etapie nie ma większego sensu. Dodatkowo, nie udało nam się zmieścić w budżecie żeby dokupić głośnik.

Robot ma za mało zasilania, przez co konieczne jest, aby był podłączony do laptopa, by móc jechać. Bardzo źle przemyśleliśmy energetyczną wydajność robota, co jest tak naprawdę głównym powodem, dla którego robot nie działa.

5 Podsumowanie

Udało nam się skonstruować robota, który próbuje spełniać większość zapostulowanych przez nas celów. Robot niestety nie działa, a główną tego przyczyną jest brak odpowiedniego zasilania (do jeżdżenia), brak odpowiedniego silnika (do wysysania powietrza) oraz brak wiedzy na temat programowania shielda EzMP3 jednocześnie z funkcjami jeżdżącymi (i jeszcze raz brak odpowiedniego zasilania). Robot jednak ma poprawnie skonstruowany kod i algorytm wykrywania przeszkód. Ma też poprawnie skonstruowany system wysysania, który jest zwyczajnie zbyt słaby (przez brak odpowiedniego zasilania). Możliwym ulepszeniem jest zaimplementowanie systemu tworzącego podciśnienie, który byłby dużo bardziej wydajny. Niestety jednak większość ulepszeń nie jest w tym momencie możliwa do wykonania, gdyż ogranicza nas budżet i czas.

Literatura

[1] <https://sketchfab.com> – odnośnik do pracy z wiatraczkiem.