

# ALGORYTM ROBOTA LEGO POKONUJĄCEGO KANION

MARIA SOBOTA, BARTOSZ MASIAK, DANIEL SOŁTYS,  
KASPER KORNAK

*AGH Akademia Górniczo-Hutnicza im. Stanisława Staszica*

*Wydział Inżynierii Mechanicznej i Robotyki,*

*Katedra Robotyki i Mechatroniki,*

*Al. Mickiewicza 30, 30-059 Kraków*

*msobota@student.agh.edu.pl, bmasiak@student.agh.edu.pl,*

*danielsoltys@student.agh.edu.pl, kasperkornak@student.agh.edu.pl*

**SŁOWA KLUCZOWE:** algorytm, lego, schemat blokowy, robot, kod, przepaść, kanion

**STRESZCZENIE:** Urbanizacja oraz intensywny rozwój przemysłu pociągają za sobą konieczność szukania innowacyjnych rozwiązań transportowych. Urządzeniem wychodzącym naprzeciw potrzebom komunikacyjnym rozwijającego się świata jest robot zdolny do pokonywania kanionów, przepaści, koryt rzecznych. Odpowiedni algorytm jest tym co pozwala urządzeniom tego typu optymalnie i bezpiecznie wykonywać swoją pracę. Schemat wykonywanych działań pozwala dokładnie opisać funkcję każdego komponentu, które to pozwalają stworzyć cały mechanizm zdolny do pokonywania przepaści.

## **1. Wstęp**

Zagadnienie, na którym się skupiono jest algorytm determinujący działanie robota pokonującego przeszkodę w postaci przepaści o zróżnicowanej głębokości i szerokości niżej nazywanej "kanionem". Programowana konstrukcja zbudowana jest z klocków oraz czujników LEGO Mindstorms z serii NXT za pomocą programu Fusion 360. Głównym zadaniem, nad którym pracowano było stworzenie schematu blokowego<sup>[1]</sup> opisującego każdą część pracy robota oraz fizyczna implementacja algorytmu dla wybranego podzespołu na platformie Arduino<sup>[2]</sup>. Robot, którego dotyczy algorytm pokonuje kanion poprzez wysunięcie mostu, na którym podwieszony jest główny segment urządzenia. Gdy most dociera na przeciwległą krawędź przepaści opuszcza się spoczywając na podporach umożliwia to podniesienie głównego segmentu robota i przeniesienie go po moście poprzez mechanizm przejazdowy. Zadaniem projektowanego algorytmu są: wykrycie czy robot jest w stanie pokonać kanion, bezpieczne wycofanie robota jeśli przeprawa nie jest możliwa oraz przejazd na drugą stronę przepaści jeśli nie ma ku temu przeciwwskazań.

## **2. Cel i zakres pracy**

Celem prac było napisanie schematu blokowego algorytmu sterującego robotem pokonującym kanion.

W rozdziale 1 dokonano wprowadzenia do tematu obejmującego podstawowe założenia działania algorytmu robota pokonującego kanion.

W rozdziale 2 przedstawiono cel i zakres prac.

W rozdziale 3 przedstawiono schemat budowy robota wraz z oznaczeniami poszczególnych komponentów elektronicznych.

Rozdział 4 zawiera przegląd zastosowanych rozwiązań wraz z dokładnym opisem całego schematu.

W Rozdziale 5 opisano sposób fizycznej implementacji algorytmu.

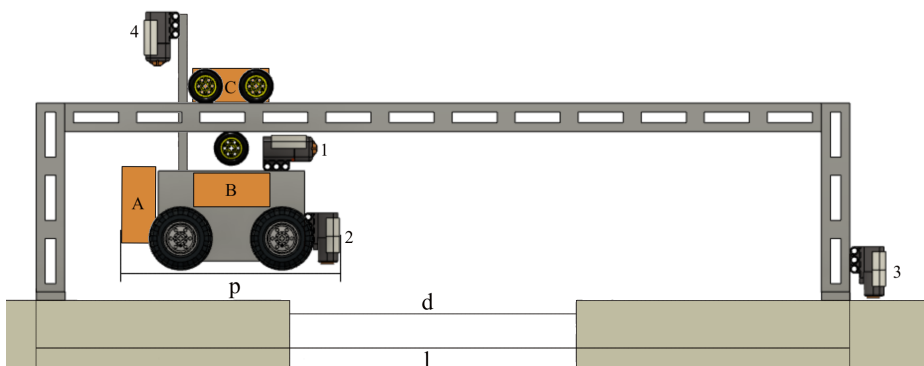
Rozdział 6 zawiera tablicę morfologiczną pokazującą możliwe rozwiązania zagadnień.

W rozdziale 7 zawarto podsumowanie prac.

Rozdział 8 zawiera bibliografię

Na końcu pracy dołączono załączniki dające szerszy ogłnd na całość projektu.

### 3. Model robota



Rys. 1 Schemat robota z opisanymi silnikami i czujnikami

Na Rys.1 przedstawiono schemat robota pokonującego kanion z oznaczonymi poszczególnymi elementami, pracę których nadzoruje opisywany algorytm. Sam robot składa się z czterech, głównych części:

- podwozia pozwalającego podejść do krawędzi kanionu
- podnośnika
- mechanizmu służącego do przejazdu po moście
- mostu

Elementami opisanymi na Rys. 1. są:

A - silnik podwozia

B - silnik podnośnika

C - silnik mechanizmu przejazdowego

1 - czujnik stykowy

2 - czujnik ultradźwiękowy

3 - czujnik ultradźwiękowy

4 - czujnik stykowy

p - długość części podwozia będącej w kontakcie z nawierzchnią

d - szerokość kanionu (pokonywanej przepaści)

l - długość mostu

## 4. Omówienie algorytmu sterującego.

### Krótki opis działania algorytmu.

Algorytm ma na celu wykonanie serii czynności robota (ruchowych, sensorycznych i obliczeniowych) które pozwolą mu na bezpieczne przeprowienie się przez kanion.

### Omówienie pracy algorytmu na podstawie schematu blokowego.

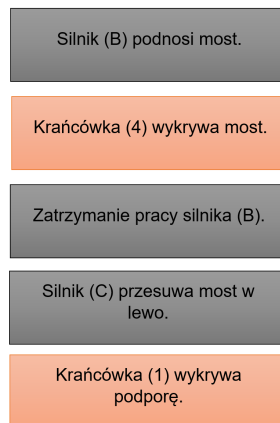
*Uwaga! W schemacie blokowym występują odnośniki liczbowe i alfabetyczne. Odnoszą się one do wyżej przedstawionego schematycznego rysunku robota (Rys. 1).*

#### **Krok 1.** Ustawienie pozycji startowej robota.

Pierwszym krokiem naszego algorytmu jest ustawienie pozycji startowej robota. Ma to na celu zapewnienie dokładnie takich samych warunków startowych przy każdym podejściu robota do pokonania przeszkody.

W tym celu silnik (B) zaczyna swoją pracę i podnosi most, aż do momentu, gdy krańcówka (4) dotknie górnej granicy mostu. W tym momencie silnik przestaje pracować i zaczyna się praca silnika (C), który wysuwa most w lewo, aż krańcówka (1) wykryje lewą podporę mostu, co będzie sygnałem, że podwozie znajduje się w prawej skrajnej pozycji mostu. W tym momencie silnik (C) przestaje pracować i tym samym kończy się część algorytmu odpowiedzialna za ustawienie pozycji startowej robota.

USTAWIANIE POZYCJI STARTOWEJ:



Rys. 2 algorytm ustawienia pozycji startowej

## Krok 2. Wykrywanie przeciwległej krawędzi.

Kolejnym krokiem jest sprawdzenie czy nasz robot jest w stanie pokonać dany kanion. Robot musi najpierw sprawdzić czy pokonanie kanionu jest w zasięgu jego możliwości, bo w przeciwnym razie nieudana próba pokonania kanionu mogłaby zakończyć się upadkiem i potencjalnym uszkodzeniem lub zniszczeniem robota.

Najpierw robot musi podejść do krawędzi kanionu. W tym celu silnik podjazdowy (A) rozpoczyna swoją pracę. Działa aż do momentu, gdy czujnik (2) wykryje przepaść. Następnie silnik (C) zaczyna wysuwać most jednocześnie licząc wykonane obroty. Jeżeli wykona ilość obrotów odpowiadającą wysunięciu się całego mostu i czujnik (3) nie wykryje przeciwległej krawędzi to algorytm przechodzi do **kroku 2.1**, a jeżeli czujnik (3) wykryje w tym czasie przeciwległą krawędź to algorytm przechodzi do **kroku 2.2**.

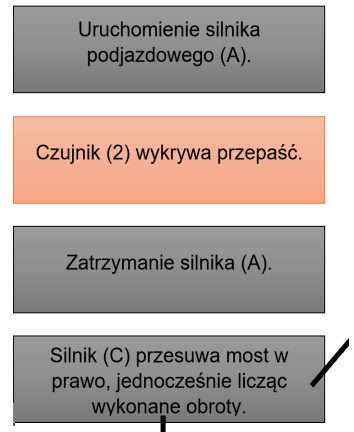
### Krok 2.1. Algorytm anulowania akcji.

Silnik nie wykrył przeciwległej krawędzi, więc następuje anulowanie akcji.

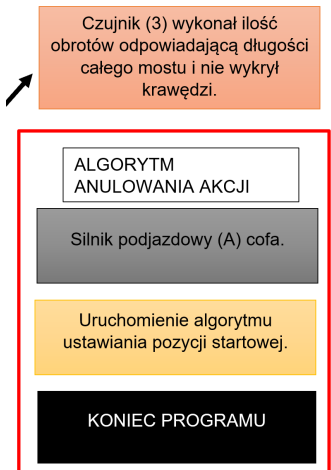
Silnik podjazdowy (A) cofa się na bezpieczną odległość, a następnie następuje pojedyncze uruchomienie algorytmu ustawiania pozycji startowej robota. Po wykonaniu wyżej wskazanych czynności robot kończy pracę. Nie jest on w stanie pokonać danego kanionu.

### Krok 2.2. Sprawdzenie czy kanion nie jest zbyt długi.

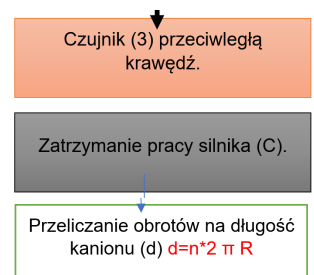
W momencie, gdy czujnik (3) wykryje przeciwległą krawędź silnik (C) przestaje pracować i następuje



Rys. 3 algorytm wykrywania krawędzi kanionu



Rys 4. algorytm anulowania akcji



Rys. 5 badanie szerokości kanionu

przeliczenie obrotów silnika na długość kanionu ( $d$ ). Schemat obliczeń został przedstawiony w zielonej ramce. Następnie robot podejmuje decyzję czy podejmie próbę przeprawienia się przez kanion. Jeżeli długość mostu  $l$  (Rys. 1) -  $d > 2p$  to długość kanionu jest odpowiednia i robot przechodzi do przeprawiania się przez kanion **krok 3.1.**, a jeżeli  $l - d \leq 2p$  to kanion jest zbyt długi i robot rezygnuje z próby pokonania kanionu **krok 3.2.**

### Krok 3.1. Przeprawa przez kanion.

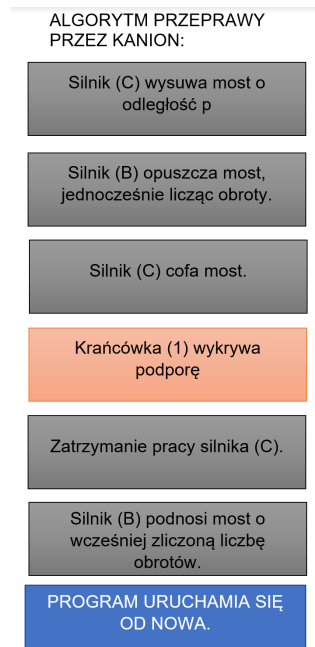
W tym kroku przedstawiona zostanie główna funkcja robota i tym samym najbardziej spektakularna tj. przeprawienie się przez kanion.

Algorytm rozpoczyna się od uruchomienia się silnika (C) i wysunięcie go o brakującą odległość  $p$ . Następnie silnik (B) opuszcza most jednocześnie zliczając obroty. Silnik (C) rozpoczyna cofanie mostu, co przy jednoczesnym uniesieniu podwozia skutkuje rozpoczęciem przeprawiania się przez kanion. W momencie, gdy krańcówka (1) wykryje przeciwległą podporę, robot dostaje informację, że znalazł się po drugiej stronie kanionu i silnik (C) zatrzymuje pracę. Następnie silnik (B) wykonuje ponownie wcześniej zliczoną ilość obrotów, ale w przeciwnym kierunku, tym samym opuszczając robota na podłoże. Następnie program uruchamia się od nowa, robot ustawi się ponownie w pozycji startowej i może przystąpić do ponownej próby pokonania kolejnego kanionu.

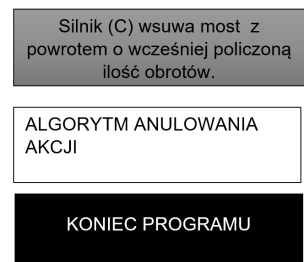
### Krok 3.2. Kanion zbyt długi. Anulowanie akcji.

W tym przypadku kanion okazał się zbyt długi, aby robot mógł go bezpiecznie pokonać i w całości, stabilnie ustawić się na jego przeciwległym brzegu. Robot rezygnuje z próby pokonania kanionu.

Silnik (C) wsuwa most z powrotem o wcześniej zliczoną ilość obrotów, a robot przechodzi do algorytmu anulowania akcji, który został omówiony w kroku 2.1.



Rys. 6 algorytm przeprawy



Rys. 7 algorytm wykorzystywany przy za szerokim kanionie

Po wykonaniu przedstawionych kroków robot kończy program i wychodzi z głównej pętli programu. Nie ma sensu uruchamiać programu od nowa, ponieważ robot i tak nie będzie w stanie pokonać danej przeszkody.

*Uwaga! Schemat blokowy do którego odnieśliśmy się w tym artykule w całości znajduje się na profilu naszego projektu na Hackaday.io [2].*

## 5. Implementacja na platformie arduino

W celu przetestowania założeń rozpisanego algorytmu stworzono kod, który zaimplementowano na platformie arduino<sup>[2]</sup>. Pracę poszczególnych podzespołów zasymulowano poprzez użycie wyświetlacza lcd, silników i czujników. W Załącznikach (Załącznik 1.) zamieszczono odnośnik do repozytorium, w którym znajduje się implementowany kod.

## 6. Tablica morfologiczna

Tablica morfologiczna części algorytmicznej robota pokonującego kanion				
Nr	problem do rozwiązania:	Rozwiązanie 1:	Rozwiązanie 2:	Rozwiązanie 3:
1	Język programowania	Python	C	Assembler
	kryterium 1:	łatwe	łatwe	średnie
	kryterium 2:	wolny	średni	szybki
2	Wykrywanie momentu w którym robot pokonał przepaść:	zliczanie obrotów silnika (C) oraz wykorzystanie wskazania czujnika (2)	zliczanie obrotów silnika (C)	wykorzystanie czujnika krańcowego (1)
	kryterium 1:	trudne	trudne	łatwe
	kryterium 2:	wolny	średni	szybki

3	Wykrywanie momentu podniesienia podwozia:	zliczanie obrotów silnika (B)	wykorzystanie czujnika krańcowego (4)	
	kryterium 1:	trudne	trudne	
	kryterium 2:	średni	wolny	

Nr.	Kryteria			
1	Łatwość implementacji	łatwe	średnie	trudne
2	Szybkość działania	wolny	średni	szybki

## 7.Podsumowanie

W wyniku prac otrzymano sprawny algorytm pozwalający wykonać robotowi jego podstawowe zadanie jakim jest pokonanie kanionu oraz bezpieczne zachowanie w razie braku możliwości wykonania tego zadania.

W procesie tworzenia algorytmu wyzwanie stanowiło przede wszystkim wysuwanie mostu na prawidłową odległość. Znalezienie rozwiązania, które optymalnie wykorzystywanie dostępne czujniki i pochodzące z nich informacje pozwoliło na wysuwanie mostu z dużą dokładnością i przewidywanie czy pokonanie kanionu jest możliwe.

W dalszym toku prac planowane jest stworzenie fizycznego modelu robota i zaimplementowanie w nim stworzonego algorytmu w celu przeprowadzenia bardziej szczegółowych testów.

## 8.Bibliografia

[1] [https://pl.wikipedia.org/wiki/Schemat\\_blokowy](https://pl.wikipedia.org/wiki/Schemat_blokowy) [Dostęp: 26.05.2021]



[2] Arduino <https://www.arduino.cc> [Dostęp 26.05.2021]

[3] Hackaday.io “Robot pokonujący kanion.” Projekt realizowany w ramach kursu Podstawy Mechatroniki na Akademii Górniczo-Hutniczej im. Stanisława Staszica w Krakowie. <https://hackaday.io/project/179947-robot-pokonujcy-kanion> [Dostęp: 26.05.2021].

### **Załączniki:**

Załącznik 1. Repozytorium zawierające implementacje części stworzonego algorytmu na platformie arduino <https://github.com/daniel834/mechatronika>