

Autonomiczny robot społeczny „Rysiek”



Koło Naukowe Robotyków
„KoNaR”

Krzysztof Andrzejewski
Wojciech Bohdan
Krzysztof Kowaczek
Tomasz Lubelski
Michał Mastej
Eryk Możdżeń
Dominik Pluta
Kamil Winnicki

1 O projekcie

Motywacja

Wzrasta potrzeba technologii, które łączą funkcjonalność z empatyczną interakcją - zwłaszcza w kontekście społeczeństw starzejących się, rosnącej samotności wśród młodych oraz sektorach usługowych (edukacja, opieka, rozrywka, handel). Rynek poszukuje rozwiązań, które nie tylko wykonują zadania, ale także budują emocjonalne więzi, redukują stres lub wspierają codzienne mikro-potrzeby (np. przypomnienie o lekach, inicjowanie kontaktu). Dodatkowo, rośnie popyt na przyjazne roboty edukacyjne dla dzieci, które uczą przez zabawę, oraz na automatyzację usług z elementami „osobowości” (np. hostele w gastronomii).

Założenia projektowe

- inicjowanie oraz utrzymywanie interakcji z człowiekiem,
- komunikacja za pomocą kolorowych świateł i ruchów,
- możliwość implementacji różnych scenariuszy użycia,
- przyjazny *design* zwierzęcia.

Cel

Stworzyć robota, który nie jest „narzędziem”, lecz partnerem w codziennych wyzwaniach – od przypomnienia o szklance wody, dawce leku lub zdrowej przekąsce po rozbawienie gestem.

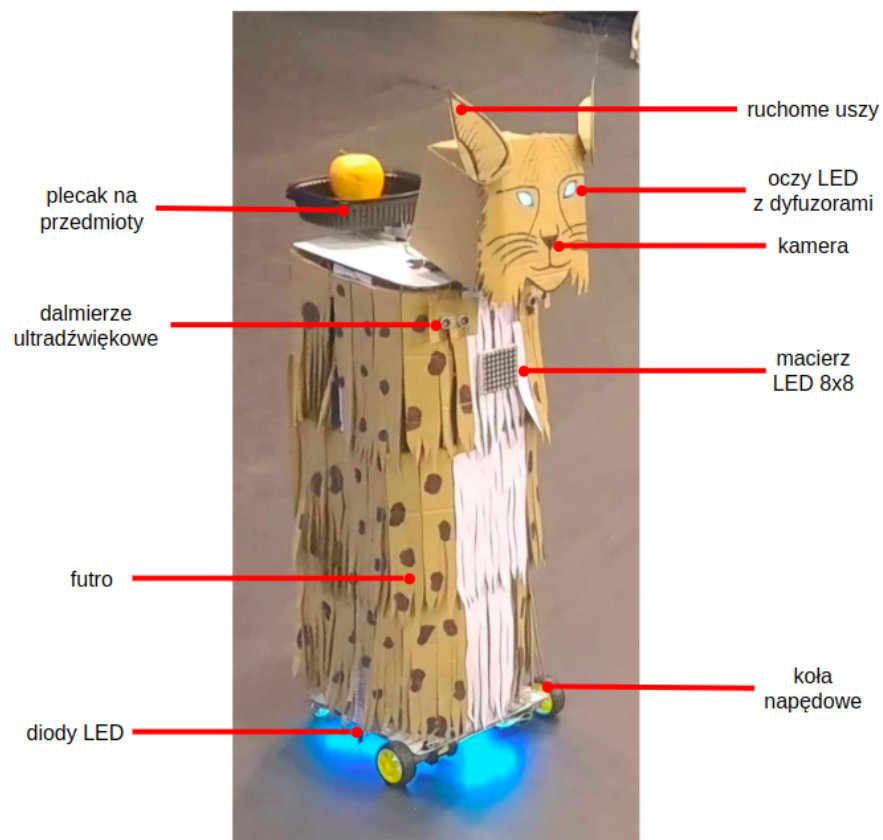
2 Budowa mechaniczna

Realizując założenia projektowe zdecydowano się na sylwetkę rysia, głównie z powodu sympatycznego usposobienia. Pełna konstrukcja została przedstawiona na rys. 1. Robot składa się z:

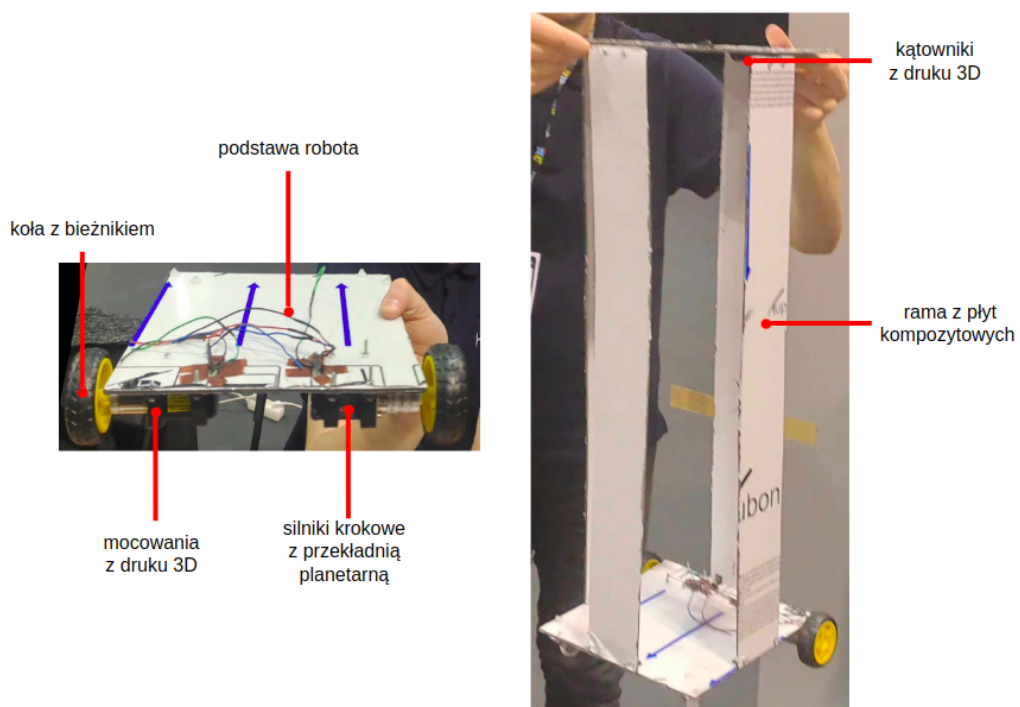
- smukłej ramy nośnej wraz z silnikami napędowymi (rys. 2),
- pokrycia wierzchniego ze spreparowanego karonu imitującego futro,
- animatronicznej głowy,
- pojemnika/plecaka na produkty zawieszzonego na czujniku tensometrycznym,

3 Układy elektroniczne

Elektronika robota została wykonana ręcznie na podstawie płytki rozwojowej *Nucleo* oraz płytki prototypowej. Uproszczony schemat połączeń przedstawia rys. 3.



Rysunek 1: Widok robota z zewnątrz



Rysunek 2: Rama robota

4 Oprogramowanie mikrokontrolera

W roli jednostki sterującej niskopoziomowymi funkcjonalnościami robota wybrano mikrokontroler STM32L476 umieszczony na płytce rozwojowej *Nucleo*. Oprogramowanie napisano w języku C z użyciem bibliotek *Hardware Abstraction Layer* udostępnionych przez producenta układu. Mikrokontroler odpowiada za:

- odczyt czujników,
- sterowanie pozycją i prędkością napędów,
- sterowanie diodami w oczach, podwoziu i macierzy LED,
- komunikacja z sterownikiem nadrzędnym.

Układy sensoryczne

Robot wykorzystuje czujniki ultradźwiękowe HC-SR04 do pomiaru odległości. Sygnał wyzwalający jest generowany przez timer w trybie PWM, a czas powrotu echa jest mierzony za pomocą przerwań GPIO i timera. Belka tensometryczna z przetwornikiem HX-711 monitoruje obciążenie w plecaku przekształca surowe dane na wartość w kilogramach. Dodatkowo, czujnik krańcowy umożliwia wyzerowanie obrotu mechanizmu głowy aby umożliwić sterowanie pozycją jej wychylenia.

Układy wykonawcze

Silniki krokowe są kontrolowane przez sterowniki TB6612 - sekwencja kroków realizowana jest poprzez dynamiczne przełączanie stanów pinów GPIO. Prędkość regulowana jest programowym dzielnikiem opartym na timerze. Serwomechanizmy SG-90 oraz pasek LED RGBW sterowane są sygnałami PWM. Macierz LED 8x8 jest odświeżana w przerwaniu timera, wykorzystując multipleksowanie rzędów i kolumn.

Komunikacja

Mikrokontroler odbiera komendy sterujące z sterownika wyższego poziomu NVIDIA Jetson Orin Nano przez magistralę szeregową UART z protokołem własnej implementacji. Ramka danych (rys. 4) zawiera:

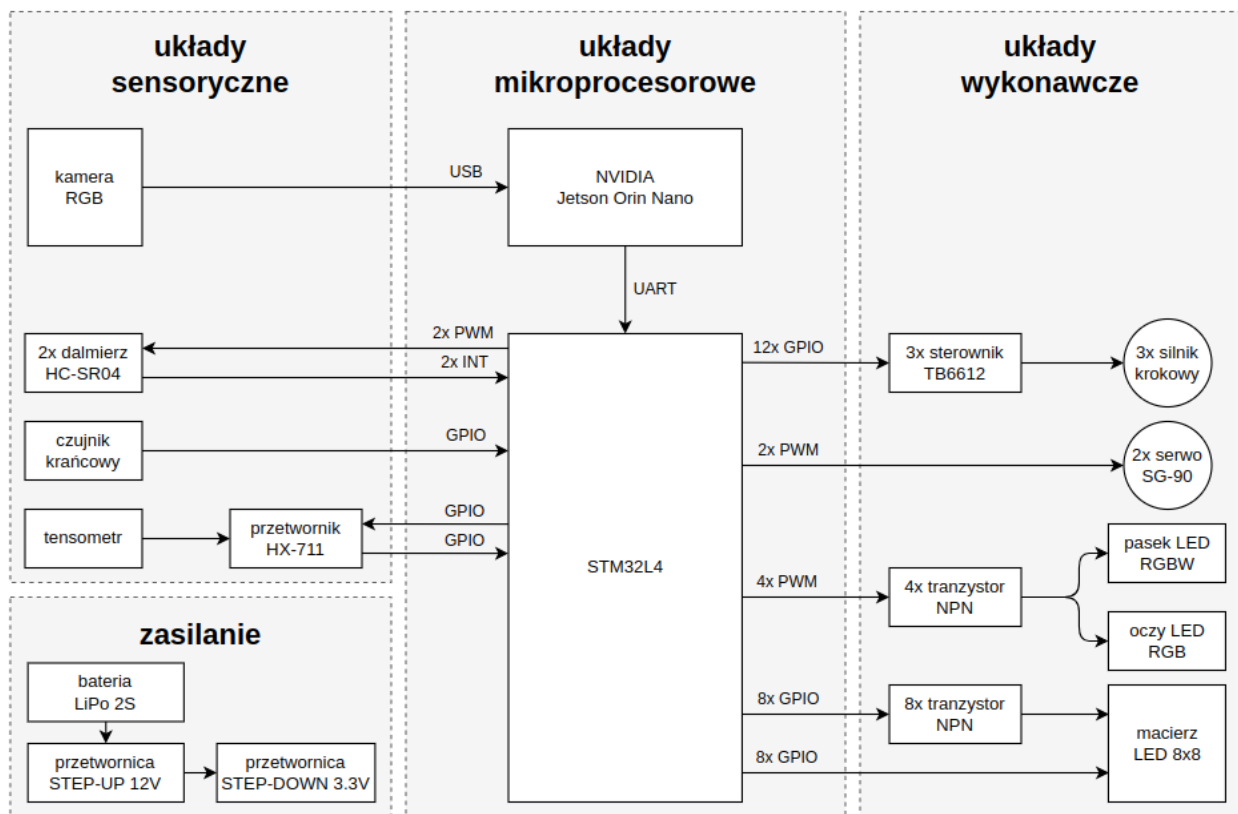
- bajty stałe do identyfikacji początku/końca ramki,
- jeden z predefiniowanych obrazów na matrycy LED,
- wartości prędkości silników,
- sumę kontrolną do analizy poprawności danych.

Dane odbierane są buforowo z użyciem DMA, a parser weryfikuje poprawność ramki przed aktualizacją stanu robota.

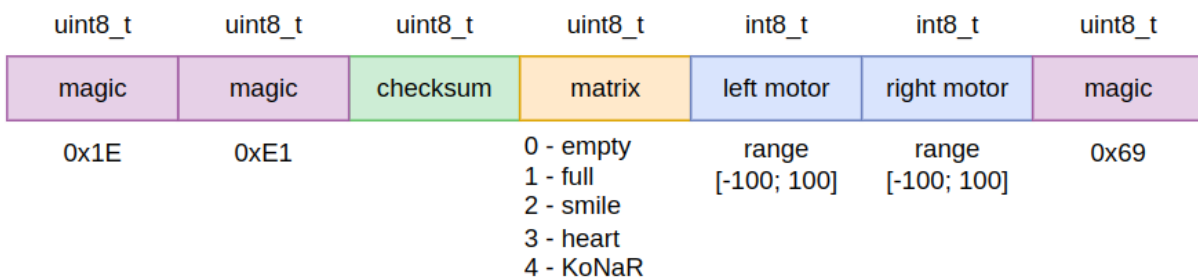
Logika niskopoziomowa

STM32 działa jako niezależny system odpowiedzialny za bezpośrednie reakcje na zdarzenia fizyczne, nadpisując komendy z jednostki nadrzędnej (Jetson Orin Nano) w sytuacjach wymagających natychmiastowej reakcji. Gdy robot wykryje nagły ubytek masy w plecaku (np. gdy użytkownik zabierze smakołyk), mikrokontroler automatycznie zatrzymuje ruch silników, inicjuje animację „zadowolenia” na macierzy LED oraz wprowadza głowę i uszy w pozycję wyrażającą reakcję emocjonalną.

Jednocześnie STM32 stale zarządza płynnymi ruchami głowy i mimiką LED, tworząc wrażenie „żywego” zachowania, nawet gdy wyższe warstwy oprogramowania skupiają się na zadaniach analitycznych. Ta warstwa zapewnia, że robot pozostaje responsywny i bezpieczny w bezpośrednich interakcjach z człowiekiem.



Rysunek 3: Uproszczony schemat układów elektroniki robota



$$\text{checksum} = (\text{magic1} + \text{magic2} + \text{matrix} + \text{motor_L} + \text{motor_R} + \text{magic3}) \% 256$$

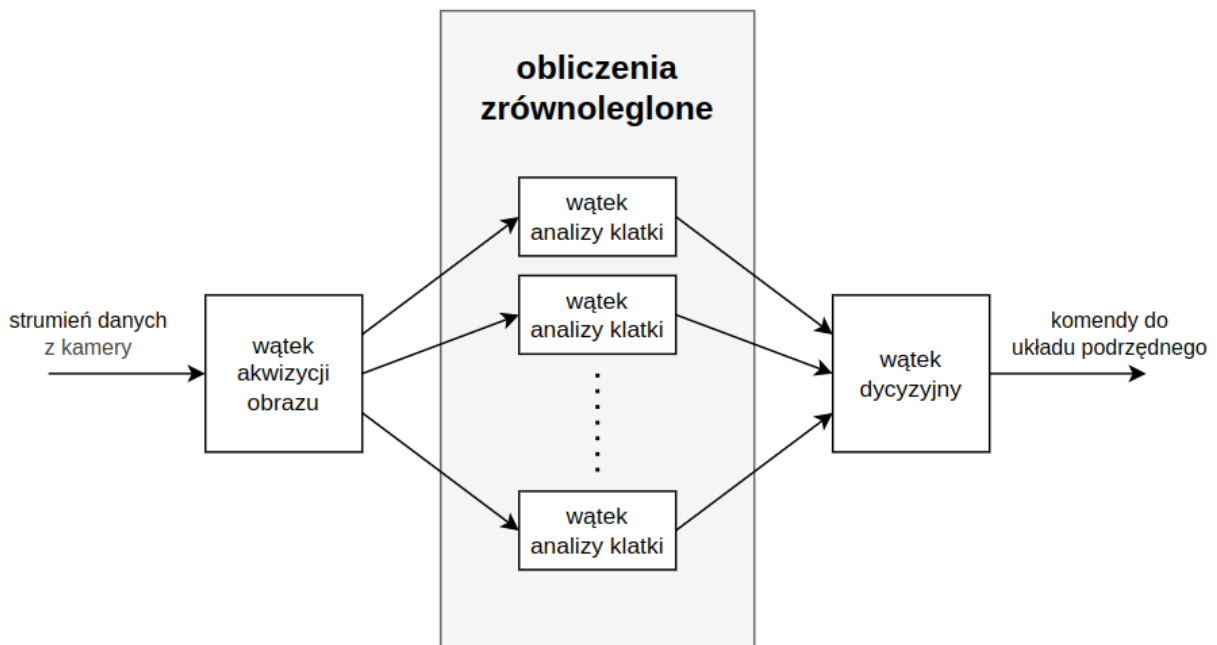
Rysunek 4: Binarna ramka danych przychodząca ze sterownika nadrzędnego

5 Oprogramowanie mikrokomputera

Oprogramowanie sterownika nadrzędnego korzysta z biblioteki OpenCV w języku Python. Obraz z kamery RGB analizowany jest w celu rozpoznawania sylwetki człowieka oraz gestów rąk. Skrypt wykorzystuje szybki algorytm, który działa na podstawie zestawu prostych cech tzw. cechy Haar'a. Są to sekwencje przeskalowanych funkcji o kwadratowym kształcie, które stosujemy do rozpoznawania szczególnych obszarów twarzy, takich jak usta, nos, oraz oczy. Dodatkowo skrypt jest w stanie rozpoznawać w której części obrazu (po prawej, czy po lewej) rozpoznaje ludzką sylwetkę.

Następnie komunikuje się z Nucleo, który steruje silnikami i w ten sposób Rysiek jest podąża za daną osobą. Aby uniknąć wykrywania wielu ludzi jednocześnie na obrazie oraz ciągłego zmieniania osoby za którą robot podąża, zastosowaliśmy skrypt który wykorzystuje wzrost osoby, za którą Rysiek ma podążać. W ten sposób pomimo wielu osób na obrazie kamery, robot nie gubi się w tłumie ludzi i podąża za odpowiednią osobą.

Równolegle następuje wielowątkowe rozpoznawanie gestów ręki (rys. 5). Wybrane gesty mają także dodatkową funkcję: pozwalają rozpocząć oraz zatrzymać algorytm śledzenia człowieka. Baza danych gestów możliwych gestów dłoni została utworzona na podstawie kilkudziesięciu ujęć rąk przez dwie różne osoby. Kiedy robot rozpozna gest ręki uruchamiana jest odpowiednia reakcja taka jak wyświetlenie odpowiedniego obrazu na macierzy LED umieszczonej na klatce piersiowej robota.



Rysunek 5: Schemat analizy danych z kamery