



LABbot

Instrukcja obsługi



KONSTRUKCJA	2
ZASILANIE	2
OBSŁUGA ROBOTA	5
URUCHOMIENIE ROBOTA	5
WYŁĄCZENIE ROBOTA	5
ŁADOWANIE	5
ZDALNY DOSTĘP DO ROBOTA	6
ARCHITEKTURA PROGRAMOWA	7
NAJCZĘSTSZE PROBLEMY	9

Konstrukcja

Masa [kg]	12 kg		
Rozmiar [m]	30 x 30 x 35 cm		
Silniki	Napięcie [V]	Moment [kg/cm]	Obroty [RPM]
	12	10	146
Akumulator	12 V, 12Ah z zabezpieczeniem przed głębokim rozładowaniem		
Zasilanie	Przetwornica podwyższająca napięcie (19V), regulowana przetwornica podwyższającą-obniżająca (2-16V)		
Czas pracy na baterii	~4-8h		
Komputer pokładowy	Intel Core i3-4010U (4th generation) 4 GB DDR3L RAM 64 GB mSata hard drive (Intel SSD) Wireless b/g/n card 4x USB 3.0 ports Infrared port Power consumption (~5W idle, ~25W full load)		
Sterownik silników	Arduino Leonardo + 3A motor shield (Pololu Dual MC 33926)		
Oprogramowanie	Ubuntu 14.04.1 + ROS Indigo Igloo		
Akcesoria	Ładowarka sieciowa		

Tabela 1. Parametry robota

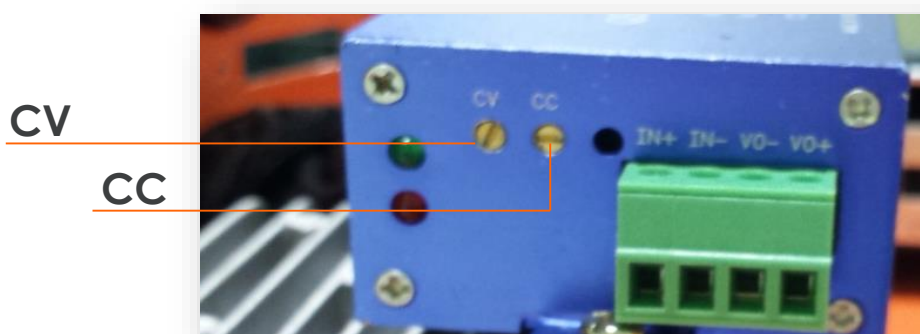
Zasilanie

Moduł zasilania robota stanowi akumulator żelowy o napięciu znamionowym 12V i pojemności 12 Ah. Akumulator jest połączony z zabezpieczeniem, które chroni go przed nadmiernym rozładowaniem.



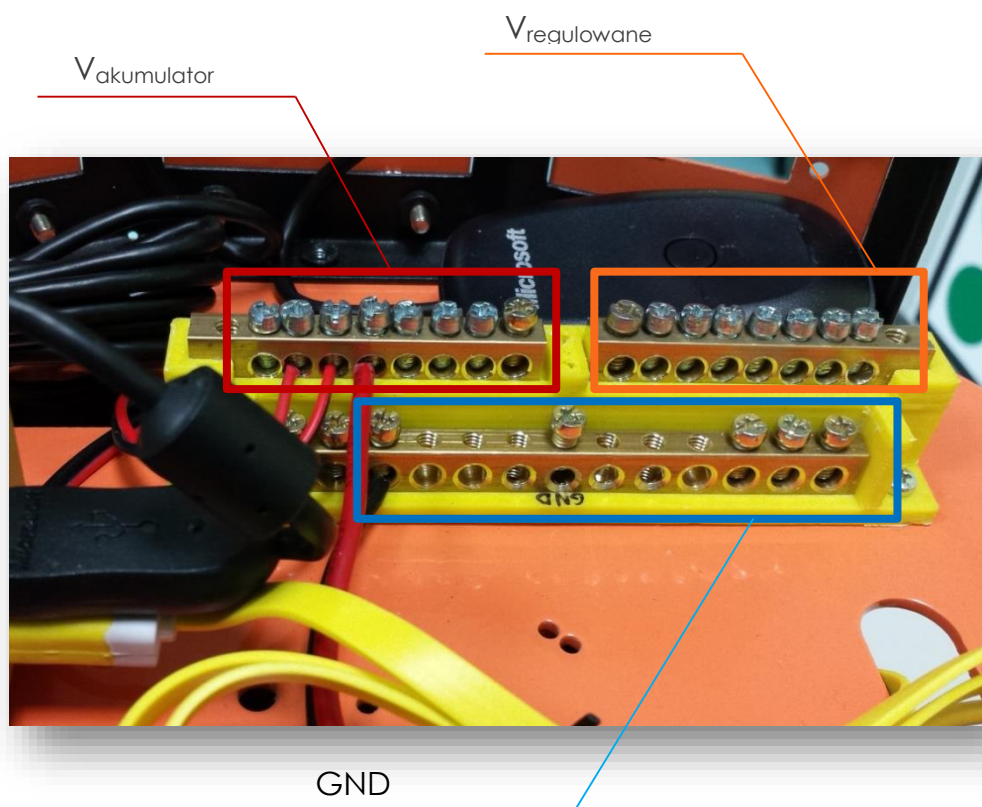
Rys. 1. Przetwornice

Robot wyposażony jest w dwie przetwornice (Rysunek 1). Przetwornica oznaczona numerem **1** generuje napięcie 19V niezbędne do zasilania komputera pokładowego. Przetwornica oznaczona numerem **2** jest przetwornicą typu *boost-buck* umożliwiającą obniżanie i podwyższanie napięcia (w zakresie 2-16V). Regulacji napięcia dokonuje się obracając śrubę o oznaczeniu **CV**. Śruba **CC** służy do ustawienia ograniczenia prądowego przetwornicy (Rysunek 2.).



Rys. 2. Przetwornica typu boost-buck.

Podłączenie dodatkowych modułów jest możliwe poprzez wykorzystanie złącza, które zaprezentowano na Rysunek 3. Na trzech listwach zaciskowych dostępne jest napięcie bezpośrednio z akumulatora ($V_{\text{akumulator}}$), napięcie regulowane z przetwornicy ($V_{\text{regulowane}}$) oraz połączenie masy (GND).

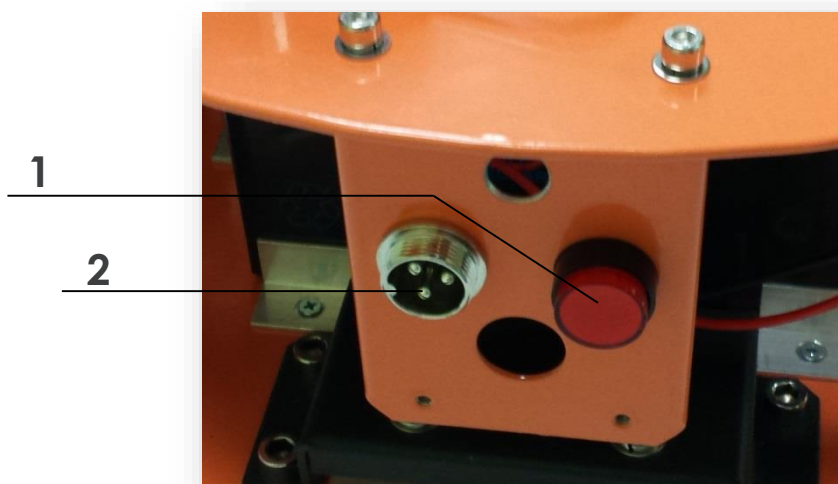


Rys. 3. Listwa zaciskowa.

Obsługa robota

Uruchomienie robota

Rysunek 4 przedstawia przycisk umożliwiający uruchomienie robota (1). Podłącza on zasilanie z akumulatora do pozostałych elementów systemu. Komputer pokładowy jest skonfigurowany do automatycznego uruchomienia po podłączeniu zasilania.



Rys. 4. Przycisk uruchomienia robota.

Wyłączenie robota

W celu bezpiecznego wyłączenia komputera pokładowego należy nacisnąć niebieski przycisk na jego obudowie. Po około 10 sekundach nastąpi wyłączenie komputera, po którym należy odłączyć zasilanie za pomocą czerwonego przycisku na tyle obudowy.

Ładowanie

W celu ładowania robota należy wykorzystać dołączoną w zestawie ładowarkę. Końcówka ładowarki jest wyprofilowana i uniemożliwia błędne połączenia. Miejsce podłączenia ładowarki przedstawia Rysunek 4 (2).

Zdalny dostęp do robota

Po uruchomieniu robot mobilny przełącza kartę bezprzewodową w tryb generowania sieci.

Domyślna nazwa:	DeepBlue-RX-wifi
Hasło:	password

Gdzie X oznacza numer robota (1/2/3/...).

W celu uzyskania dostępu zdalnego do robota należy ustawić **adres IP 10.0.0.10** i **maskę podsieci 255.255.255.0** na komputerze, a następnie podłączyć się do sieci **DeepBlue-RX-wifi** i za pomocą protokołu SSH nawiązać połączenie z adresem **10.0.0.1**.

Domyślne dane logowania

Username:	labbot
Password:	password

Po zalogowaniu należy wywołać polecenie:

tmux a -t labbot

które, korzystając z menedżera terminali **tmux** (<http://en.wikipedia.org/wiki/Tmux>), podłącza się do istniejących sesji. Za pomocą skrótu **CTRL+A, W** uruchamiane jest menu wyboru aktywnego okna. W oknie **roscore** uruchomione jest polecenie **roscore**, natomiast w oknie **roslaunch_labbot** uruchomione jest polecenie **roslaunch labbot labbot_launch**. W celu zatrzymania któregoś z nich należy nacisnąć w odpowiednim oknie klawisze **CTRL+C**.

Przydatne skróty klawiszowe (**uwaga!** wszystkie skróty należy poprzedzić naciśnięciem **CTRL+A**):

- c – nowe okno,
- w – lista okien,
- , - nazwij okno,
- & - zamknij okno
- ? – wyświetl skróty.

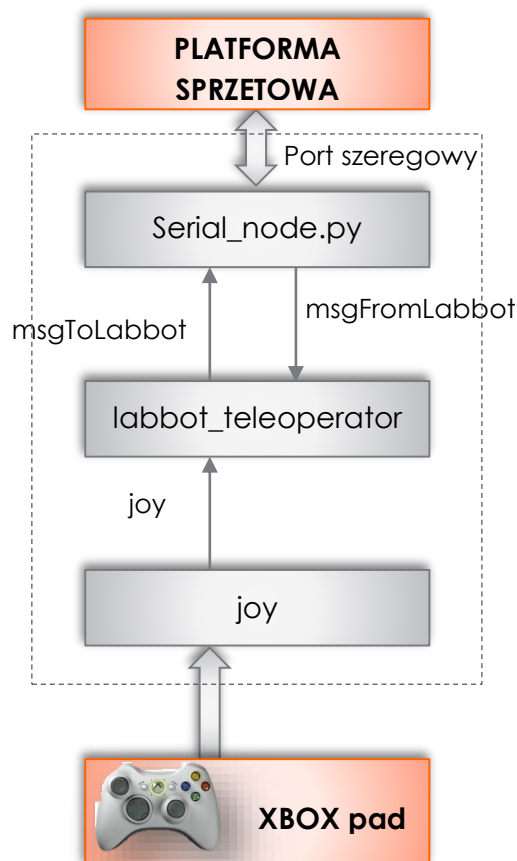
Architektura programowa

Na komputerze pokładowym zainstalowany jest system Ubuntu 14.04.1 LTS w wersji 64-bitowej wraz z pakietem oprogramowania ROS Indigo Igloo wersji desktop-full.

Skrypt obrazujący pełen przebieg procesu instalacyjnego (wraz z opisem) jest dostępny w repozytorium:

https://github.com/PUTvision/ROS-labbot/blob/master/utils/labbot_ros_indigo_install_script

Architektura programowa została zaprezentowana na Schemacie 1.



Schemat 1. Architektura programowa robota LABbot.

Przykładowe oprogramowanie jest uruchamiane przy starcie systemu za pomocą skryptu **labbot_start** dostępnego w repozytorium:

https://github.com/PUTvision/ROS-labbot/blob/master/utils/labbot_start

Pierwszy element, dostępny w ROS modułu **joy**, odczytuje stan przycisków gamepada i publikuje je w temacie **/joy**. Dane te są wykorzystywane przez moduł

labbot_teleoperator, który na podstawie wychylenia gałek wyznacza prędkość dla silników robota. Dane te są publikowane w temacie **/labbot/msgToLabbot** i poprzez moduł **serial_node.py** (z pakietu **rosserial**) przekazywane do sterownika silników (Arduino Leonardo).

Przesyłane specyficzne wiadomości oraz moduł **labbot_teleoperator** został zawarty w katalogu **labbot_ws** (workspace typu catkin). Link do repozytorium:

https://github.com/PUTvision/ROS-labbot/tree/master/labbot_ws

Najczęstsze problemy

W celu przetestowania poszczególnych elementów systemu należy:

a) gamepad (moduł joy):

- sprawdzić czy w systemie dostępny jest joystick:

ls /dev/input

- na liście wyników poszukiwać dostępnego gamepada (w postaci **jsX**)
- uruchomić moduł joy:

roslun joy joy_node

- podejrzeć wiadomości publikowane temacie joy (typu sensor_msgs/Joy):

rostopic echo joy

b) połączenie z sterownikiem silników (modułu roserial):

- sprawdzić dostępne w systemie porty szeregowo:

ls /dev

- na liście poszukiwać urządzenia o nazwie **tttyACMX**
- dla znalezionej portu uruchomić:

roslun roserial_python serial_node.py /dev/ttyACM0

- podejrzeć wiadomości publikowane w temacie msgFromLabbot

rostopic echo msgFromLabbot