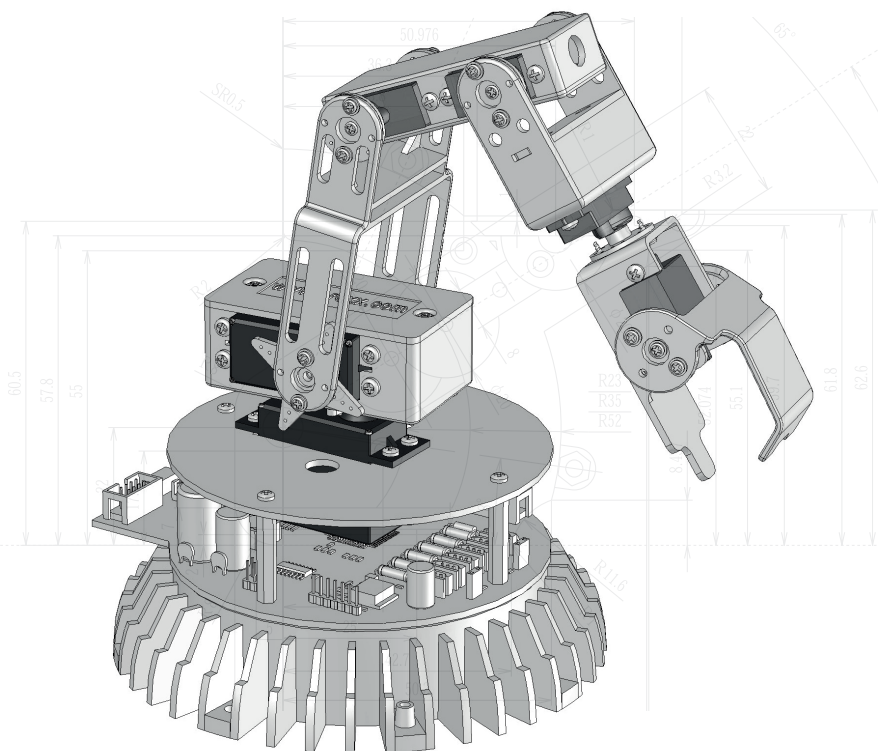




ROBOT DYDAKTYCZNY ROBOT ARM HOBBY V3

INSTRUKCJE MONTAŻU: Model RA2-HOBBY



© AREXX - THE NETHERLANDS V0513

Spis treści

1.	Opis produktu Ramię Robota	5
2.	Wymagane narzędzia	7
3.	Lista części	8
4.	Instrukcje montażu	10
5.	Elementy elektroniczne	23
6.	Instalacja programu	27
7.	Programator i Loader	40
	7.1 RobotLoader	41
	7.2 Podłączenie USB na Windows	42
	7.3 Podłączenie USB na Linux	45
	7.4 Test interfejsu USB	46
	7.5 Otwieranie portu na Linux	47
	7.6 Autotest	48
	7.7 Kalibracja	50
	7.8 Test klawiatury	52
8.	Oprogramowanie RACS	53
9.	Programowanie Ramienia Robota	59
xx.	DODATEK	
	A. Schemat elektryczny Ramienia Robota	68
	B. Schemat elektryczny układu zasilania	69
	C. Schemat elektryczny złączy	70
	D. Schemat elektryczny klawiatury	71
	E. Schemat płytki PCB	72

AREXX i ROBOT ARM znaki towarowe zastrzeżone AREXX Engineering - HOLLAND.

© Wersja polska 2014: AREXX Engineering (NL).

Wszystkie prawa zastrzeżone. Każda reprodukcja lub adaptacja całości lub części podręcznika wymaga pisemnej zgody ze strony importera europejskiego:

AREXX Engineering - Zwolle (NL).

Producent i dystrybutor nie ponoszą odpowiedzialności za szkody spowodowane niewłaściwym stosowaniem produktu, błędnym montażem lub nieposzanowaniem instrukcji zawartych w tym podręczniku.

Producent i dystrybutor zastrzegają sobie prawo do zmiany treści niniejszego podręcznika bez uprzedzenia.



Producent:
AREXX Engineering
DAGU HI-TECH



Importer europejski:
AREXX Engineering
ZWOLLE The Netherlands

Wsparcie techniczne dostępne na:

WWW.AREXX.COM
WWW.ROBOTERNETZ.DE

© AREXX Holland and DAGU China

© Polskie tłumaczenie: AREXX - Holandia

Impressum

©2007 AREXX Engineering

Nervistraat 16
8013 RS Zwolle
The Netherlands

Tel.: +31 (0) 38 454 2028
Fax.: +31 (0) 38 452 4482

E-Mail: Info@arexx.nl

Wszystkie prawa zastrzeżone. Każda reprodukcja lub adaptacja całości lub części podręcznika wymaga pisemnej zgody.

Specyfikacja produktu oraz dostarczane treści mogą podlegać zmianom. Treść podręcznika może ulec zmianie bez wcześniejszego uprzedzenia.

Darmowe aktualizacje podręcznika można znaleźć na:

<http://www.arexx.com/>

"Robot Arm PRO i -Hobby" to znaki towarowe zastrzeżone przez AREXX Engineering. Wszystkie inne znaki towarowe należą do ich właściciela. Nie ponosimy odpowiedzialności za treści zawarte na zewnętrznych stronach internetowych wspomnianych w tym podręczniku!

Informacje dotyczące ograniczonej gwarancji oraz odpowiedzialności

Gwarancja udzielona przez AREXX Engineering ogranicza się do zamiany lub naprawy Ramienia Robota oraz jego elementów w okresie trwania gwarancji jeśli usterka dotyczy wad produkcyjnych: brak elementów elektronicznych, wadliwe lub nieprawidłowo zamontowane elementy mechaniczne itp.

Gwarancja nie dotyczy bezpośrednio uszkodzonych elementów podczas użytkowania. Takie założenie wyklucza wszelkie roszczenia, które podlegają odpowiedzialności prawnej.

Gwarancja nie obejmuje elementów robota lub akcesoriów robota, które zostały trwale zmienione, np. przylutowanie innych, zewnętrznych elementów, wywiercenie dziur samodzielnie itd. Nie obejmuje również uszkodzeń dokonanych poprzez brak wiedzy lub ignorowanie ostrzeżeń zawartych w podręczniku!

Gwarancja traci ważność, jeśli użytkownik nie przestrzega zasad z podręcznika! AREXX Engineering nie ponosi odpowiedzialności za szkody powstałe wskutek nieprawidłowego użytkowania niezgodnego z instrukcjami zawartymi w podręczniku. Należy stosować się do wszystkiego, co zawiera sekcja "Zalecenia dotyczące bezpieczeństwa".

Proszę zwrócić uwagę na umowy licencyjne oprogramowania na CD-ROM!

WAŻNE!

Przed pierwszym użyciem ramienia robota całkowicie zapoznaj się z treścią podręcznika! W podręczniku zawarte są informacje na temat odpowiedniego sposobu użycia oraz potencjalnych zagrożeń! Ponadto, zawiera informacje, które mogą być nieoczywiste dla wszystkich użytkowników.

Ważne zalecenia dotyczące bezpieczeństwa

Ramię robota jest wyposażone w wysoce czułe części. Elementy elektroniczne są wrażliwe na wyładowanie elektrostatyczne. Elementy podnoś delikatnie za boki w bezpiecznych miejscach. Unikaj bezpośredniego kontaktu z elementami na płycie PCB.

Symbole

W podręczniku użyto następujących symboli:



"Uwaga!" Używany, aby ostrzec lub zwrócić uwagę na ważne szczegóły. Zignorowanie ostrzeżeń może skutkować uszkodzeniem robota i/lub jego części oraz może być szkodliwe dla twojego zdrowia lub zdrowia innych osób!



"Informacja" Symbol wskazuje użyteczne wskazówki lub informacje uzupełniające. Podążanie za wskazówkami nie jest konieczne, ale może ułatwić pracę.

Zalecenia dotyczące bezpieczeństwa

- Sprawdź polaryzację baterii i źródła zasilania.
- Unikaj wilgoci. Jeśli nastąpi kontakt produktu z wodą, należy usunąć baterie lub odłączyć układ od zasilania.
- Jeśli robot jest nieużywany przez dłuższy czas, odłącz go od zasilania.
- Sprawdź robot i jego kable przed użyciem - to pomoże uniknąć awarii.
- Jeśli uważasz, że praca z urządzeniem nie jest już bezpieczna, odłącz je i upewnij się, że nie zostanie nieświadomie użyte.
- Gdy nie jesteś pewien funkcji, bezpieczeństwa lub połączenia elementu, skonsultuj się ze specjalistą.
- Nie podejmuj pracy z urządzeniem w niekorzystnych miejscach lub warunkach.
- Nie przeciążaj serwomechanizmów.
- Urządzenie zawiera wysoce czułe komponenty, np. elementy elektroniczne są wrażliwe na wyładowanie elektrostatyczne. Elementy podnoś delikatnie za boki w bezpiecznych miejscach. Unikaj bezpośredniego kontaktu z elementami na płytce PCB.

Użytkowanie

Niniejszy projekt został stworzony do celów eksperymentalnych dla każdego zainteresowanego robotyką. Głównym zadaniem jest nauka programowania urządzenia w języku C. Produkt nie jest zabawką; nie jest przeznaczony dla dzieci poniżej 14 roku życia!

Dane ramię robota nie jest również robotem przemysłowym zapewniającym dokładny przemysłowy opis oraz wydajność!

Robot może być użyty jedynie wewnątrz budynku. Produkt nie może zostać zmoczony lub poddany działaniu jakiegokolwiek wilgoci. Unikaj skraplania wody na elementach: zachowaj ostrożność po wniesieniu robota z zimnego miejsca. Pozwól elementom dostosować się do nowego środowiska i odczekaj pewien czas zanim rozpoczniesz używanie.

Nie przestrzeganie powyższych zaleceń może doprowadzić do uszkodzenia produktu i może wpłynąć na powstanie dodatkowych zagrożeń takich jak zwarcie, ogień, porażenie prądem itd.

Należy dokładnie zapoznać się z treścią tego podręcznika.

1. OPIS PRODUKTU RAMIĘ ROBOTA

Ramię Robota to robot w przystępnej cenie dedykowany hobbistom. Jest doskonałym sposobem na nauczenie się podstaw elektroniki, mechaniki i programowania. Ramię Robota jest sterowane przez mikrokontroler ATMEGA64, który programuje się z poziomu programu Open Source Tools w języku C. Użytkownik w łatwy i szybki sposób może wgrać do mikrokontrolera swój własny program poprzez port USB oraz oprogramowanie Uploadera.

I/O (wejścia/wyjścia) wraz z elastyczną magistralą I2C pozwalają na dodawanie nadprogramowych modułów, co umożliwi robotowi reakcję na otoczenie.

Zawartość opakowania:

- Pełny zestaw konstrukcyjny Robot Arm (mechanika i elektronika)
- USB z pinem
- CD-ROM zawierający wymagane oprogramowanie i instrukcje

1.2. Szczegóły techniczne:

- mikrokontroler ATMEGA64
- wiele I/O wejść/wyjść
- magistrala I2C
- 4 mini-serwonapędy DGServo 12g
- 2 maxi-serwonapędy DGServo S07NF STD
- plastikowe ramię i metalowe podłoże oraz baza
- długość ramienia: 260 mm
- wysokość (z bazą): 320 mm
- średnica podłoża: 150 mm
- zasilanie: 9-14V

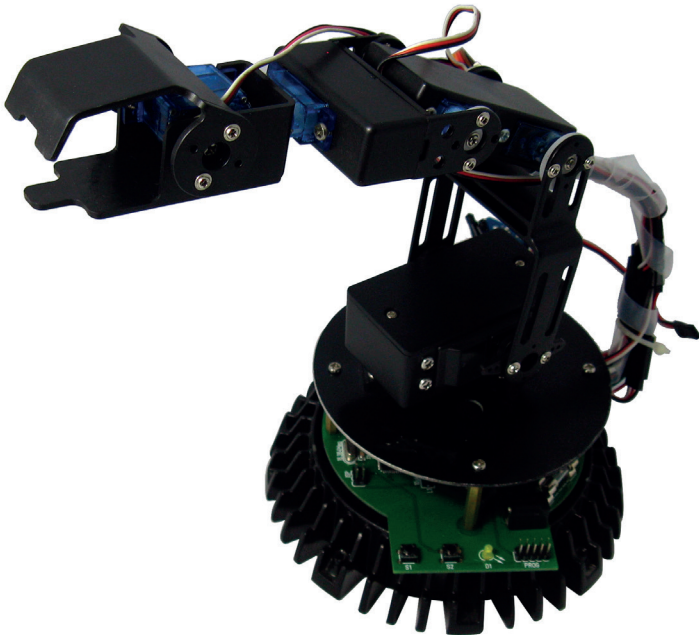


Ostrzeżenie!

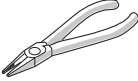
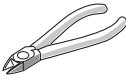
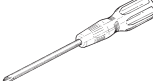
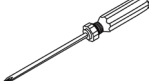
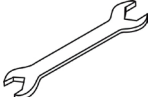
- * Prawo do zwrotu przemiesza w momencie otwarcia plastikowych opakowań elementów.
- * Przeczytaj uważnie podręcznik zanim zamontujesz wszystkie elementy.
- * Zachowaj ostrożność podczas pracy z narzędziami.
- * Nie montuj robota w obecności małych dzieci. Może stać im się krzywda .
- * Sprawdź, czy baterie zostały umieszczone poprawnie.
- * Upewnij się, że koszyk baterijny jest zawsze suchy. Jeśli robot zostanie zamoczony, usuń baterie i bardzo delikatnie osusz elementy.
- * Wyjmij baterie jeśli Ramię Robota nie będzie używane dłużej niż tydzień.

1.3. Co można zrobić z Ramieniem Robota?

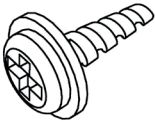
- Przesłać do robota przykładowy lub nowy program.
- Sterować ramię za pomocą klawiatury.
- Sterować i programować Ramię Robota przez program RACS.
- Rozszerzać Ramię Robota gotowymi do użycia modułami aby robot mógł reagować na środowisko: widzieć, czuć, słyszeć.
- Tak jak roboty przemysłowe wykonują różne zadania, np. budują samochód, ten robot również może wykonywać dla Ciebie pewne prace.
- Ramię Robota może komunikować się ze środowiskiem oraz innymi jednostkami przez magistralę I2C.
- Sztuczna inteligencja: robot automatycznie ulepsza swoje oprogramowanie dzięki swojemu samouczącemu się oprogramowaniu.



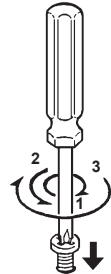
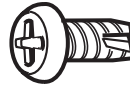
2. WYMAGANE NARZĘDZIA

szcypce półokrągłe	szcypce tnące boczne	zestaw śrubokrętów	śrubokręt	klucz płaski
			 Dołączony	 Dołączony

Wkręt samogwintujący (Parker)



Wkręty samogwintujące działają jak wkręty do drewna. Wwiercają się w materiał formując miejsce na gwint wkrętu. Dlatego właśnie ten wkręt ma końcówkę ostrzej zakończoną niż zwykle wkręty.



Wkręty samogwintujące mają wycięcie na łbie, które ułatwia wwiercanie się w materiał. Aby w szybki sposób wkręcić i poprawnie umocować wkręty wykonaj kroki:

- 1 Wkręć wkręt w materiał
- 2 Lekko poluzuj wkręt
- 3 Dokręć ponownie do końca

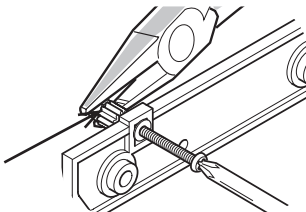
Jeśli wkręt nie zostanie wkręcony do końca, może się jeszcze bardziej poluzować i stopniowo powiększać otwór sprawiając, że wkręt będzie niedopasowany.

Nakrętka zabezpieczająca

Mocowanie nakrętki



nakrętka



Nie dokręcaj na siłę! Pod wpływem dużej siły plastik może się złamać!

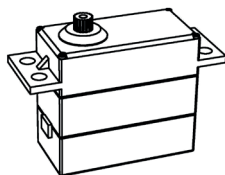
Nadg

Zestaw zawiera jeden podwójny klucz płaski. Używaj go do nakrętek typu M2 i M3. Zamiennie można użyć szcypcy.



3. LISTA CZĘŚCI

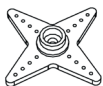
Serwomotor



○ 4 x Mini
DGServo 12g

○ 2 x Maxi
DGServo S07NF STD

Orczyk



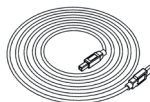
○ 2x

CD



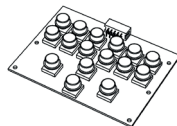
○ 1x

Kabel USB
typu A-B



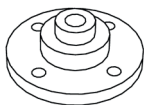
○ 1x

Klawiatura



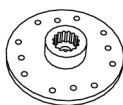
○ 1x

Tarcza dystansowa



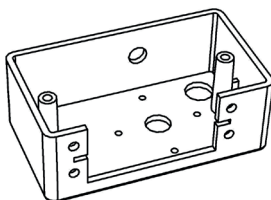
○ 4x

Tarcza do serwowatoru



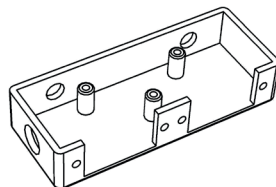
○ 4x

Koszyk na duży serwowator



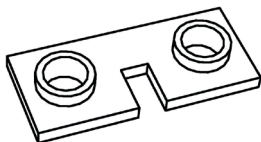
○ 1x

Podwójny koszyk na serwowator



○ 1x

Podkładka dystansowa dla serwowatoru



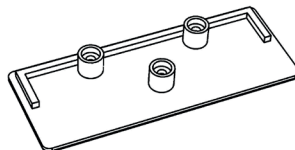
○ 2x

Przykrywka koszyka na duży serwonapęd



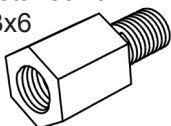
○ 1x

Przykrywka koszyka na serwonapędy



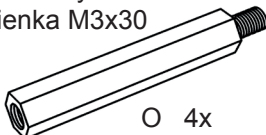
○ 1x

Śruba dystansowa gruba M3x6



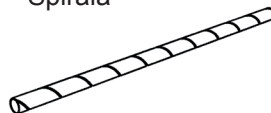
○ 4x

Śruba dystansowa cienka M3x30



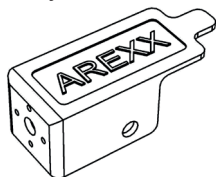
○ 4x

Spirala



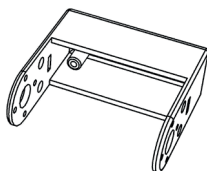
○ 1x

Dolna szczeka
chwytaka



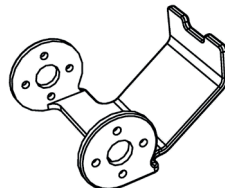
○ 1x

Nadgarstek robota



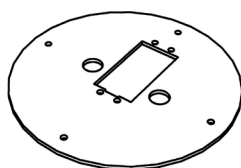
○ 1x

Gorna szczeka
chwytaka



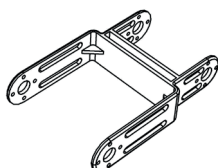
○ 1x

Podloze robota



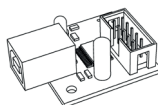
○ 1x

Połączenie
przegubów



○ 1x

Konwerter
USB



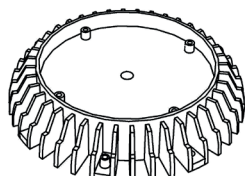
○ 1x

Przewód
taśmowy



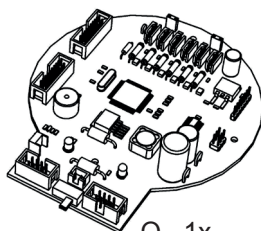
○ 1x

Baza ramienia robota



○ 1x

Płytkę PCB



○ 1x

Taśma do
klawiatury
1,5m



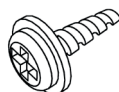
○ 1x

Przewód do
serwomotorów



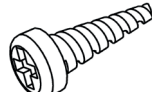
○ 1x

Maly wkręt do
serwomotoru
M2x6



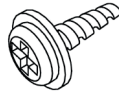
○ 8x

Wkręt
samogwintujący
M2.3x8



○ 4x

Duzy wkręt do
serwomotoru
M2.3x6



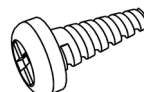
○ 10x

Wkręt
samogwintujący
M2.6x6



○ 8x

Wkręt
samogwintujący
M3.5x8



○ 4x

O-Ring M3



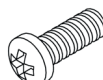
○ 3x

Śruba M3x6



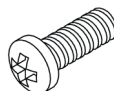
○ 4x

Śruba M3x8



○ 8x

Śruba M3x10



○ 2x

Nakrętka
M3



○ 4x

Nakrętka
zabezpieczająca
M3



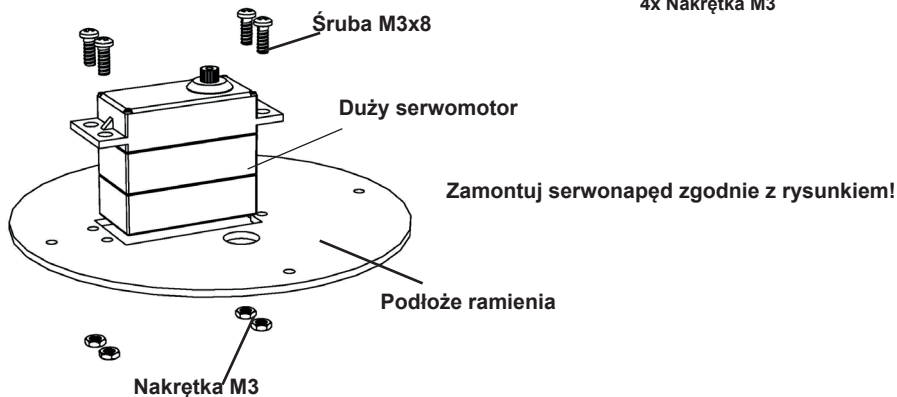
○ 4x

4. INSTRUKCJE MONTAŻU

Montaż dolnej części serwonapędu:

Potrzebne będą następujące części:

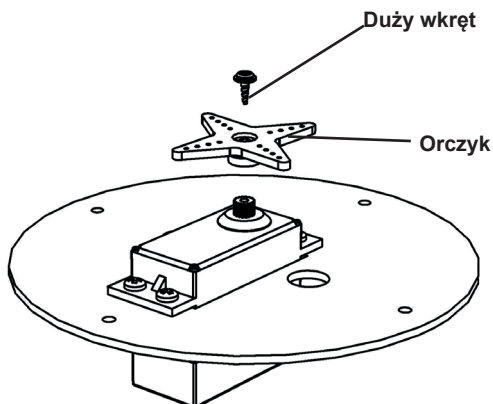
1x Podłoże ramienia
1x Duży serwomotor
4x Śruba M3x8
4x Nakrętka M3



Montaż orczyka:

Potrzebne będą następujące części:

1x Podłoże z serwonapędem
1x Orczyk
1x Duży wkręt do serwomotoru M2.3x6



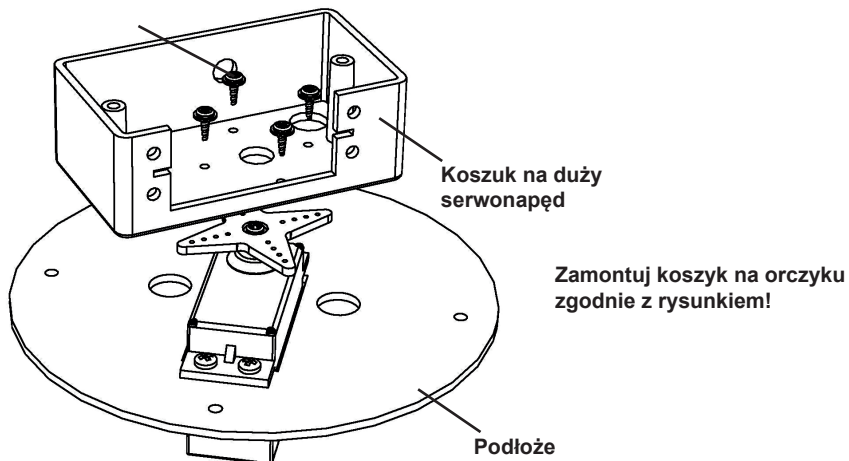
Zamontuj orczyk na serwonapędzie zgodnie z powyższym rysunkiem!

Montaż górnej części serwonapędu:

Potrzebne będą następujące części:

1x Podłoże
1x Koszyk na duży serwonapęd
4x Mały wkręt do serwomotoru M2x6

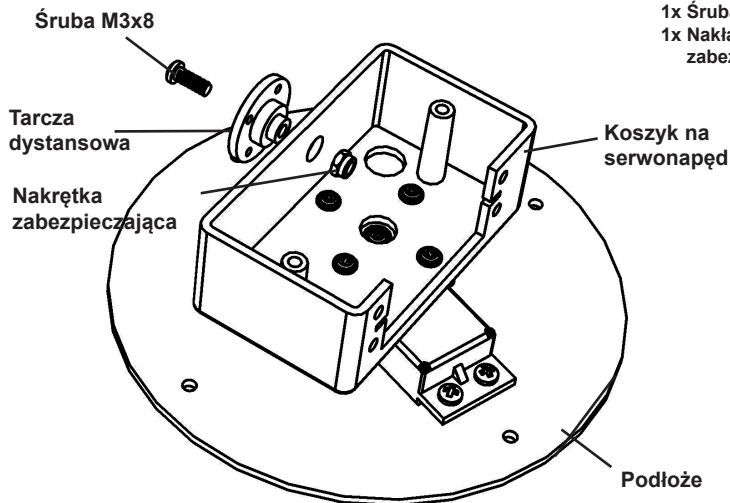
Mały wkręt M2x6



Montaż koszyka na serwonapęd:

Potrzebne będą następujące części:

1x Podłoże z zamontowanym
koszykiem na serwonapęd
1x Tarcza dystansowa
1x Śruba M3x8
1x Nakładka
zabezpieczająca M3

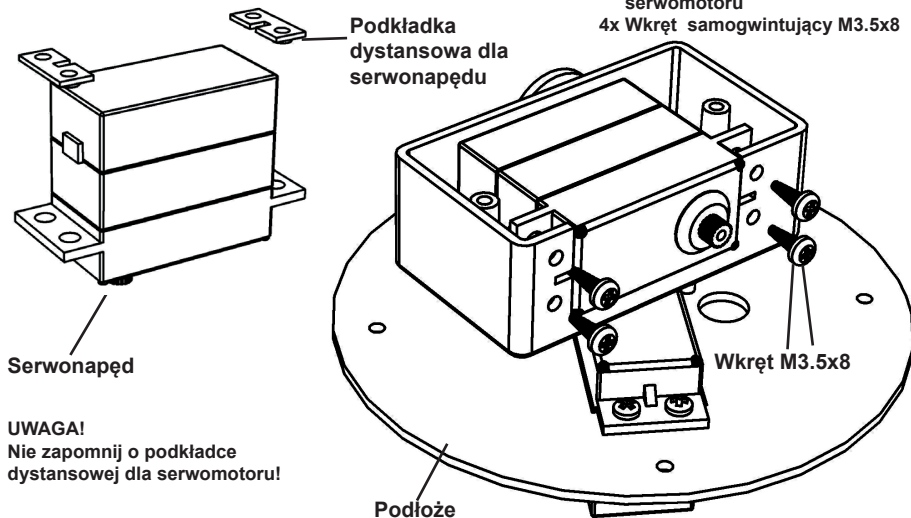


Zamontuj tarczę dystansową zgodnie z powyższym rysunkiem!

Montaż serwonapędu:

Potrzebne będą następujące części:

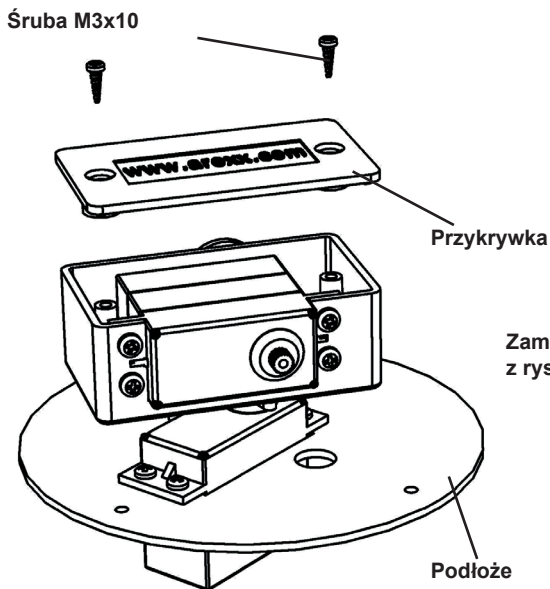
- 1x Podłoże z elementami
- 1x Duży serwomotor
- 1x Podkładka dystansowa dla serwomotoru
- 4x Wkręt samogwintujący M3.5x8



Montaż przykrywki koszyka na serwonapęd:

Potrzebne będą następujące części:

- 1x Podłoże elementami
- 1x Przykrywka koszyka na duży serwonapęd
- 1x Śruba M3x10

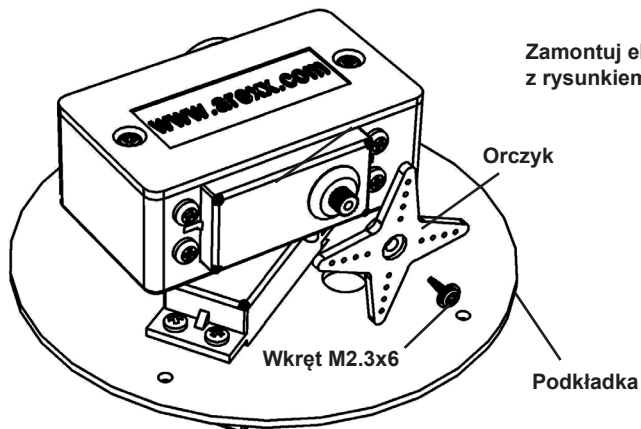


Zamontuj elementy zgodnie z rysunkiem!

Montaż orczyka:

Potrzebne będą następujące części:

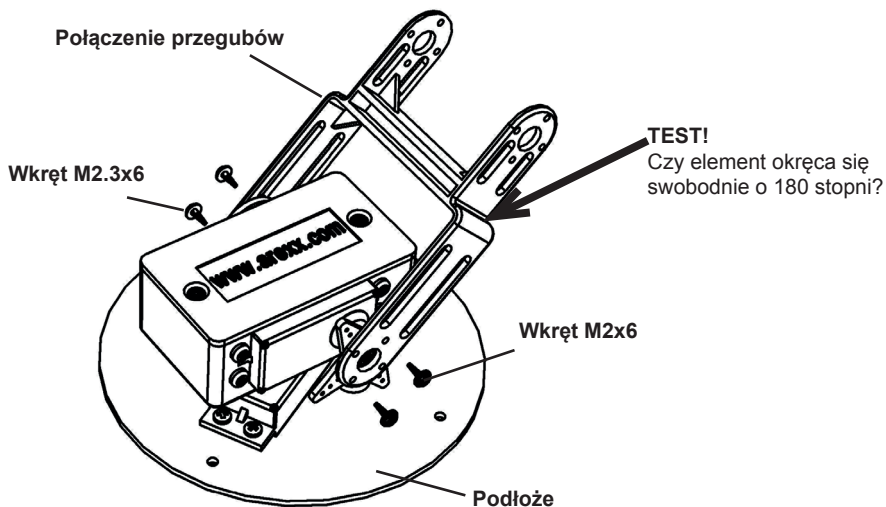
- 1x Podłoże z elementami
- 1x Orczyk
- 1x Duży wkręt do serwomotoru M2.3x6



Montaż połączenia przegubów:

Potrzebne będą następujące części:

- 1x Podłoże z elementami
- 1x Połączenie przegubów
- 1x Mały wkręt do serwomotoru M2x6
- 2x Duży wkręt do serwomotoru M2.3x6

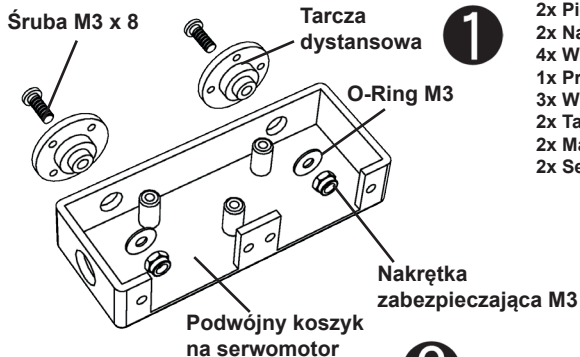


Zamontuj elementy zgodnie z powyższym rysunkiem!

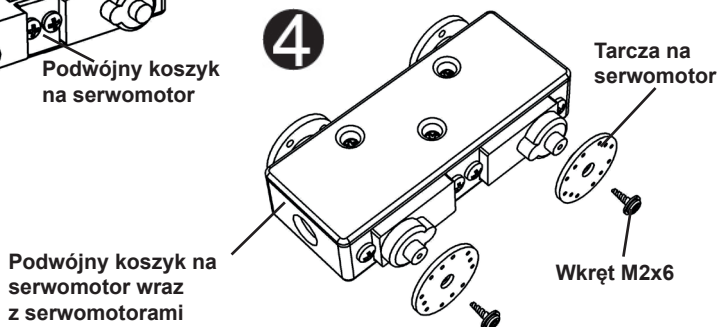
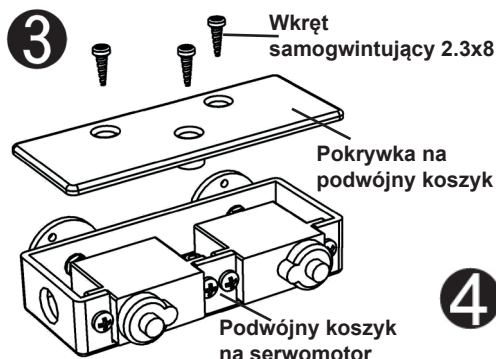
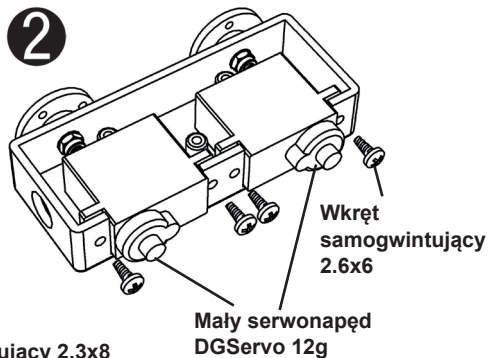
Montaż dwóch serwomotorów:

W ostatnim kroku montażu serwonapędów, potrzebne będą następujące części:

- 1x Podwójny koszyk na serwomotor
- 2x Tarcza dystansowa
- 2x Śruba M3 x 8
- 2x Pierścień M3
- 2x Nakrętka zabezpieczająca M3
- 4x Wkręt samogwintujący 2.6x6
- 1x Przykrywka
- 3x Wkręt samogwintujący 2.3x8
- 2x Tarcza na serwonapęd
- 2x Mały wkręt do serwomotoru M2x6
- 2x Serwonapęd DGServo 12g



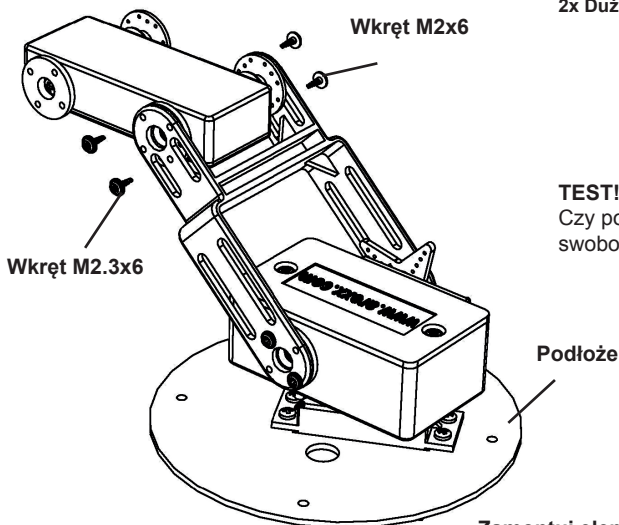
Zamontuj elementy zgodnie z powyższym rysunkiem!



Montaż podwójnego koszyka:

Potrzebne będą następujące części:

1x Podłoże z elementami
1x Podwójny koszyk na serwomotor
2x Mały wkręt do serwomotoru M2x6
2x Duży wkręt do serwomotoru M2.3x6



TEST!

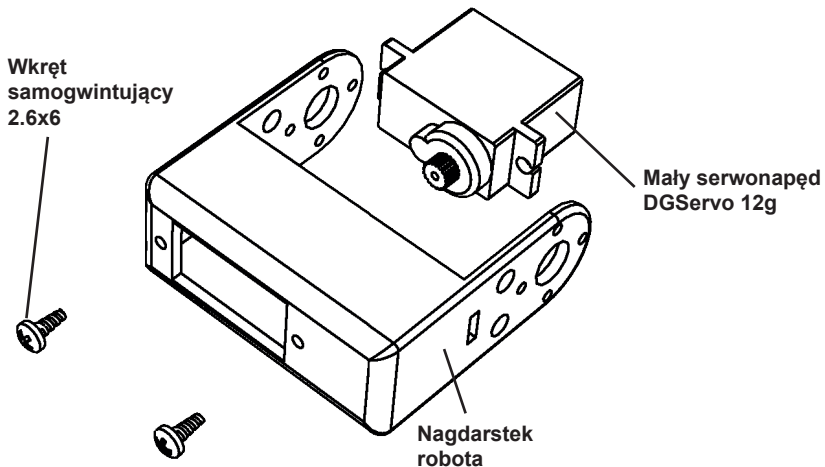
Czy podwójny koszyk obraca się swobodnie o 180 stopni?

Zamontuj elementy zgodnie z powyższym rysunkiem!

Montaż nadgarstka robota:

Potrzebne będą następujące części:

1x Mały serwonapęd DGServo 12g
1x Nadgarstek robota
2x Wkręt samogwintujący M2.6x6

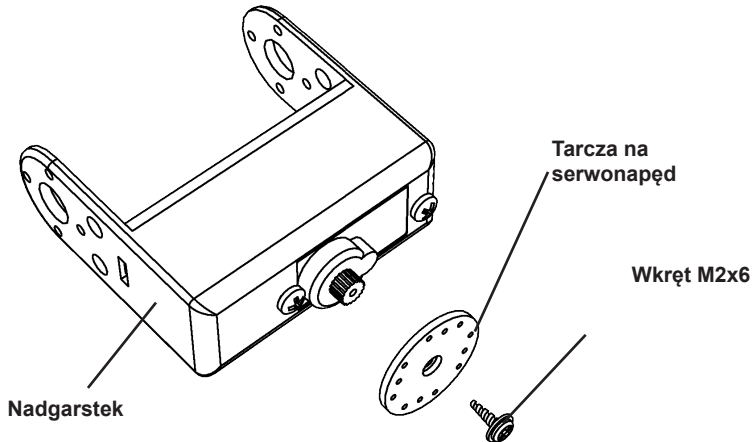


Zamontuj nadgarstek robota zgodnie z powyższym rysunkiem!

Końcowy etap składania nadgarstka:

Potrzebne będą następujące części:

- 1x Nadgarstek z elementami
- 1x Tarcza na serwonapęd
- 1x Mały wkręt do serwomotoru M2x6



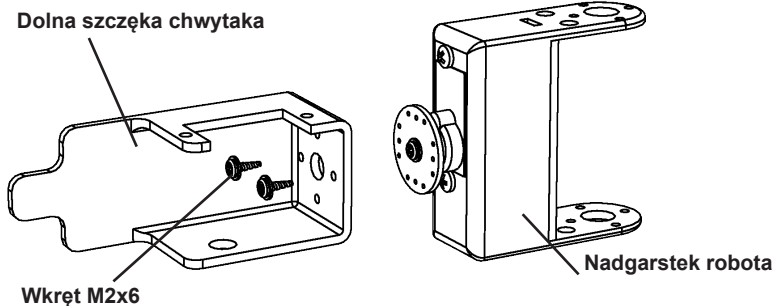
Zamontuj tarczę robota zgodnie z powyższym rysunkiem!



Montaż dolnej szczęki chwytaka:

Potrzebne będą następujące części:

1x Zmontowany nadgarstek
1x Dolna szczeka chwytaka
2x Mały wkręt do serwomotoru M2x6

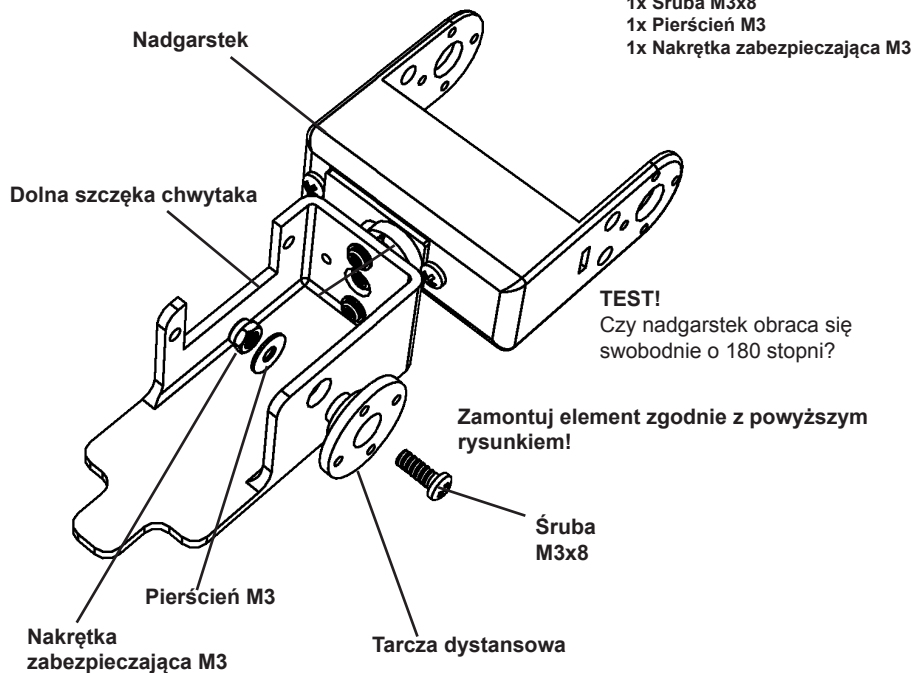


Zamontuj element zgodnie z powyższym rysunkiem!

Montaż serwonapędu w dolnej szczęki chwytaka:

Potrzebne będą następujące części:

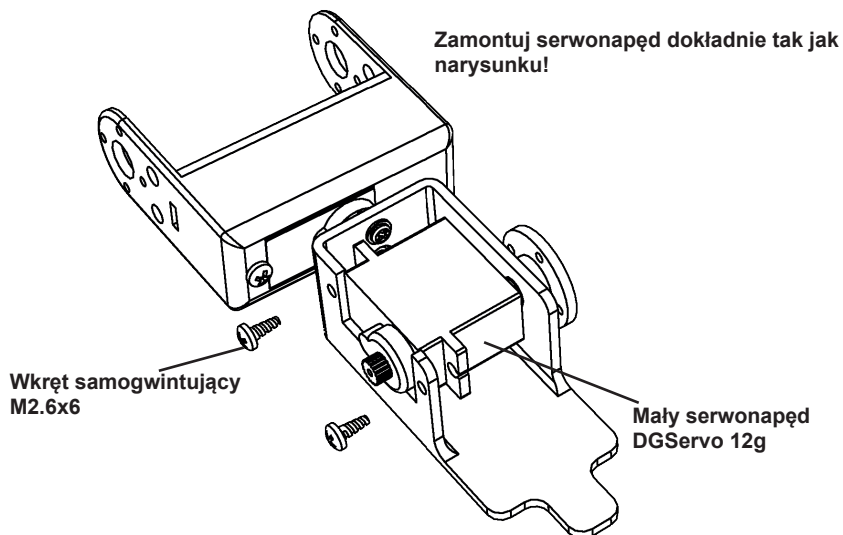
1x Zmontowany nadgarstek
1x Dolna szczeka chwytaka
1x Tarcza dystansowa
1x Śruba M3x8
1x Pierścień M3
1x Nakrętka zabezpieczająca M3



Montaż serwonapędu chwytaka:

Potrzebne będą następujące części:

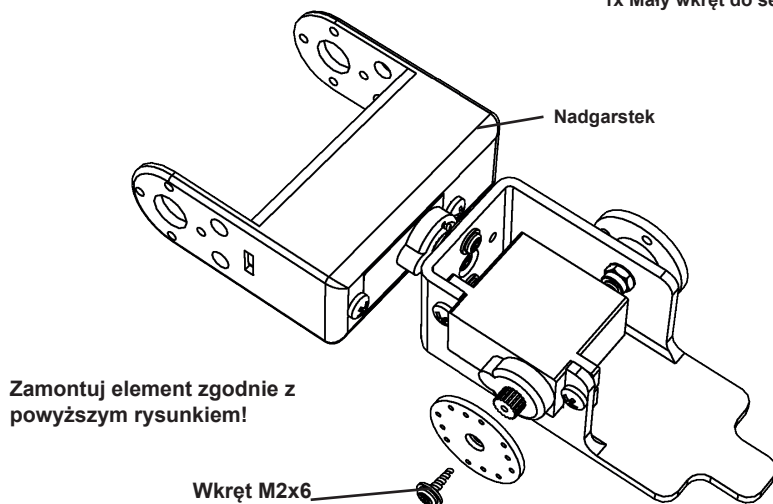
1x Zmontowany nadgarstek
1x Mały serwonapęd DGServo 12g
2x Wkręt samogwintujący M2.6x6



Montaż dolnej części chwytaka:

Potrzebne będą następujące części:

1x Zmontowany nadgarstek
1x Tarcza na serwonapęd
1x Mały wkręt do serwowymotora M2x6

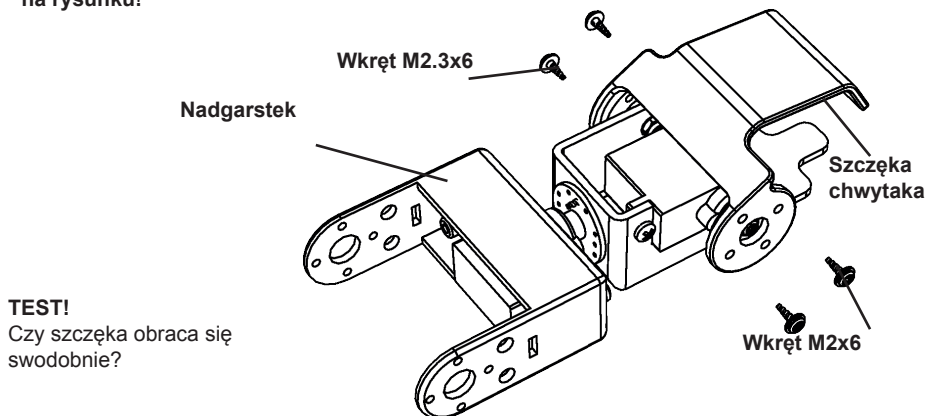


Montaż górnej szczęki chwytaka:

Potrzebne będą następujące części:

Zamontuj szczękę dokładnie tak jak na rysunku!

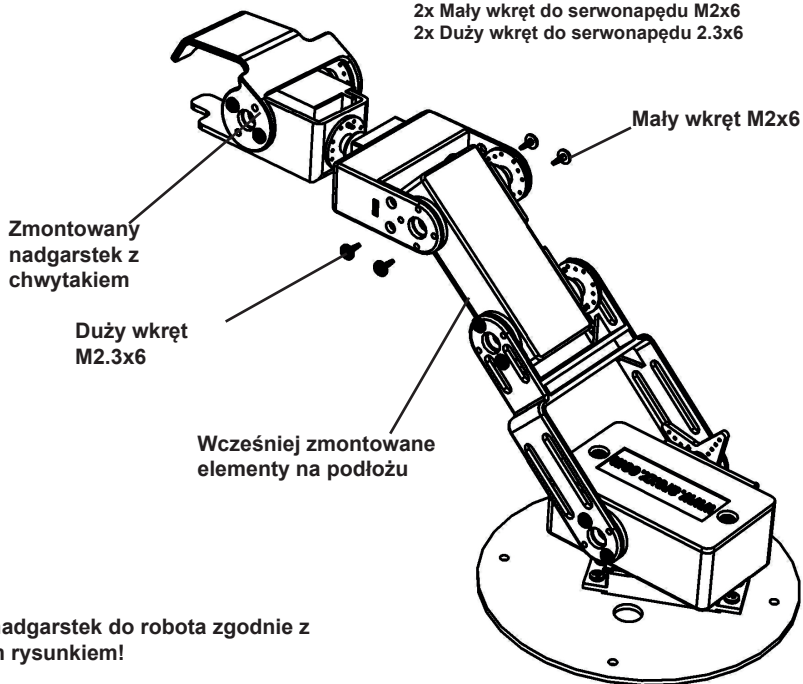
- 1x Zmontowany nadgarstek
- 1x Górna szczęka chwytaka
- 2x Mały wkręt do serwonapędu M2x6
- 2x Duży wkręt do serwonapędu M2.3x6



Końcowy etap montażu:

Potrzebne będą następujące części:

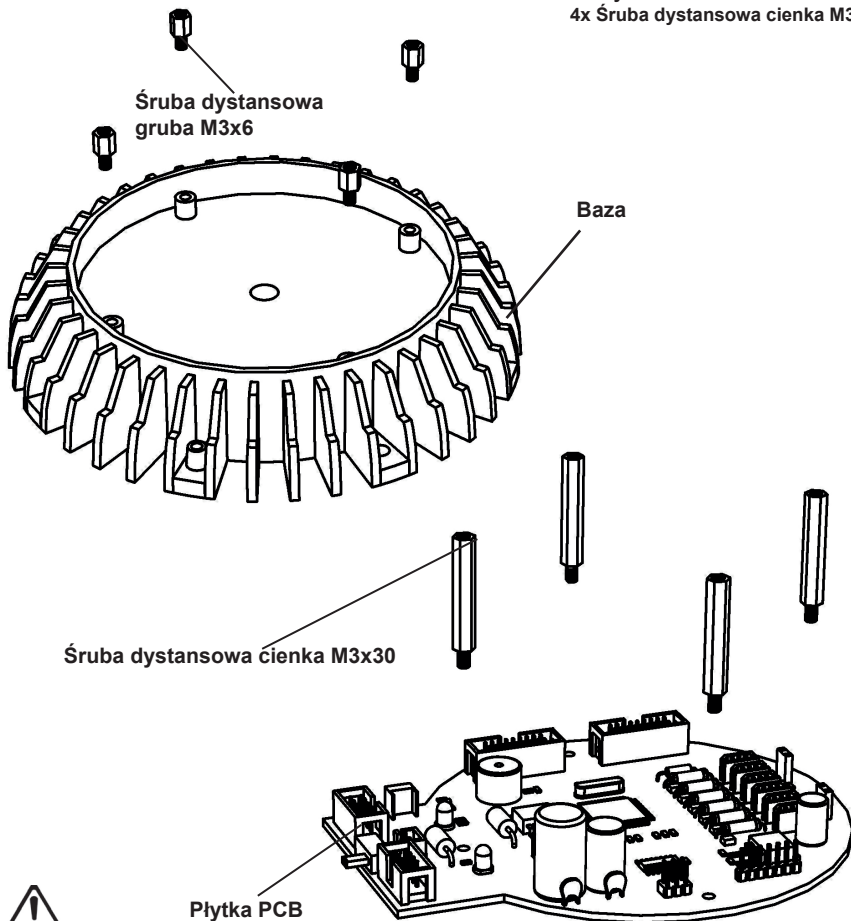
- 1x Zmontowany nadgarstek z chwytakiem
- 1x Zmontowane elementy na podłożu
- 2x Mały wkręt do serwonapędu M2x6
- 2x Duży wkręt do serwonapędu 2.3x6



Montaż bazy i płytki PCB:

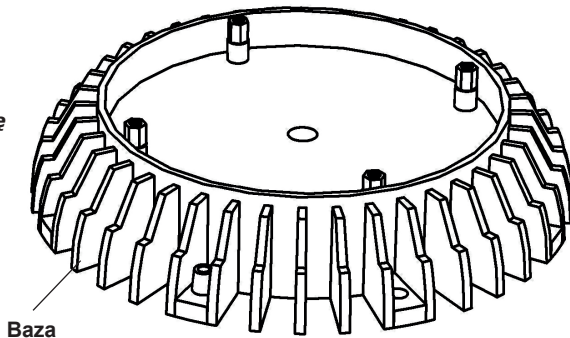
Potrzebne będą następujące części:

1x Metalowa baza
4x Śruba dystansowa gruba M3x6
1x Płytki PCB
4x Śruba dystansowa cienka M3x30



WAŻNE!

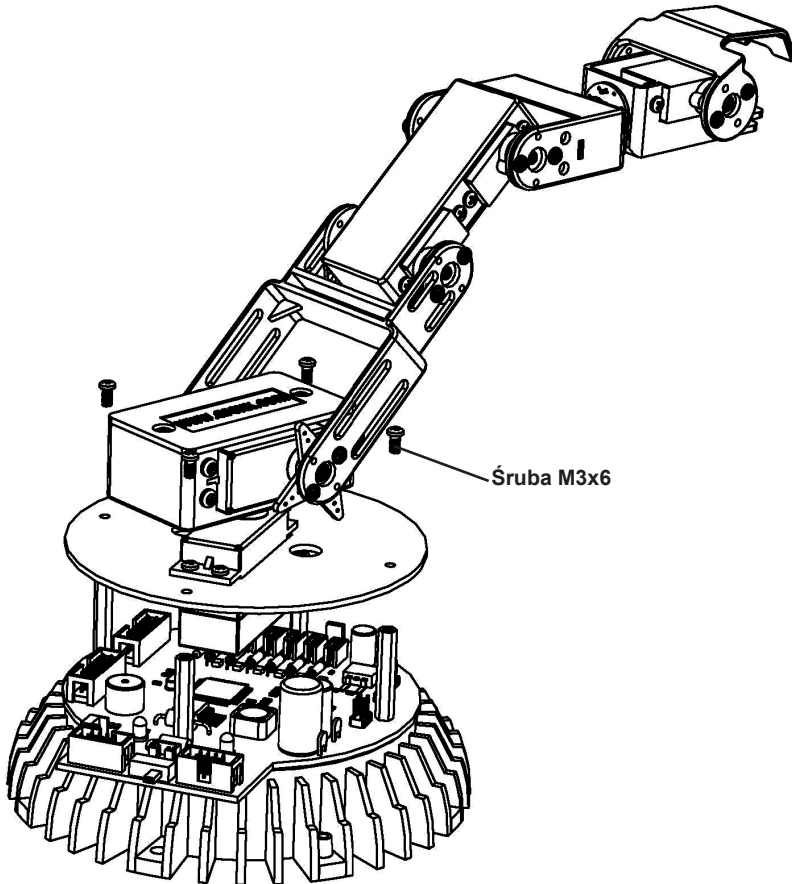
Unikaj bezpośredniego kontaktu z elementami na płytce! Staraj się podnosić ją za brzegi.



Końcowy montaż ramienia robota:

Potrzebne będą następujące części:

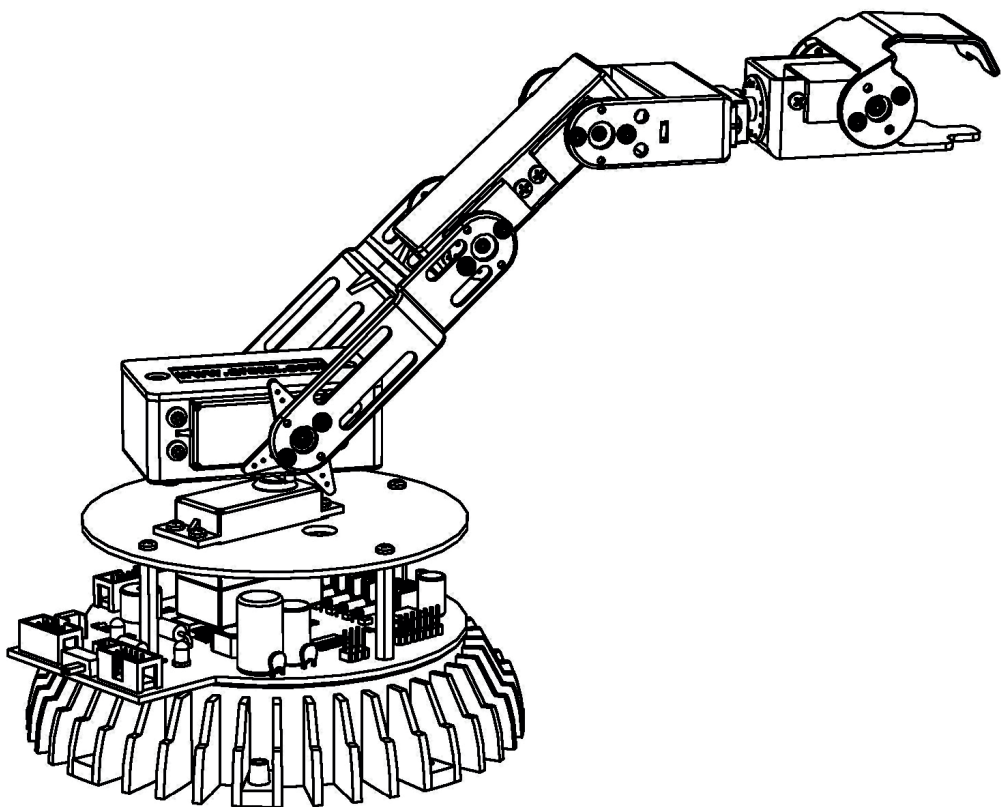
1x Baza z płytką
1x Zmontowane ramię
4x Śruba M3x6



WAŻNE!

Podłącz wszystkie przewody przed przystąpieniem do etapu z rysunku powyżej.
Odnieś się do schematów na stronie 25 i 51.

GOTOWE !



5. ELEMENTY ELEKTRONICZNE

5.1. Dane techniczne

Ten rozdział zawiera informacje związane z danymi technicznymi robota oraz pewien wstęp do podstawowych zagadnień, które pozwolą ci zaznajomić się z terminologią użytą w tym podręczniku. Większość zagadnień zostanie wyjaśniona w późniejszych rozdziałach.

Cechy elementów i dane techniczne ramienia robota:

- **Mikrokontroler o dużej mocy Atmel ATMEGA64 (8-Bitowy)**
 - ◇ Szybkość 16 MIPS (=16 milionów instrukcji na sekundę) przy częstotliwości zegara równej 16MHz.
 - ◇ Pamięć: 64KB Flash ROM, 4KB SRAM, 2KB EEPROM.
 - ◇ Programowalny w języku C (z użyciem WinAVR / avr-gcc)!

- **Elastyczny system rozszerzeń, oparty na magistrali I²C**
 - ◇ Wymagane jedynie dwa sygnały (TWI -> "Two Wire Interface").
 - ◇ Szybkość przesyłania danych do 400kBit/s.
 - ◇ Tryb Master->Slave (urządzenie Nadrzędne->Podrzędne).
 - ◇ Do magistrali może być podłączonych jednocześnie 127 urządzeń podrzędnych (Slave).
 - ◇ Bardzo popularny układ z magistralą. Na rynku dostępny jest szereg układów scalonych, czujników i innych elementów, które mogą zostać bezpośrednio podłączone.

- **Możliwość dołączenia modułów bezprzewodowych**
 - ◇ RP6 WIFI.
 - ◇ Bluetooth.
 - ◇ APC-220.

- **Interfejs USB PC do przesyłania programów z PC do mikrokontrolera**
 - ◇ Połączenie kablowe dla maksymalnej prędkości przesyłania danych. Program wysyłany jest z prędkością 500kBaud, wypełniając miejsce w pamięci (30KB, 2KB zarezerwowane dla Bootloadera) w kilka sekund.
 - ◇ Może być użyty do programowania dostępnych modułów rozszerzających dla ramienia robota z mikrokontrolerami AVR (moduł RP6-256-WIFI).
 - ◇ Może być użyty do komunikacji między modułami rozszerzającymi robota. Można zastosować, np. do analizy błędów kodu poprzez wysyłanie do PC wiadomości tekstowych i innych danych.
 - ◇ Dostępny jest również wirtualny port COM (VCP) dla wszystkich popularnych systemów operacyjnych: Windows 2K/XP/Vista, Linux i inne. VCP może być użyty w standardowych programach uruchamianych z terminala lub w spersonalizowanym oprogramowaniu.

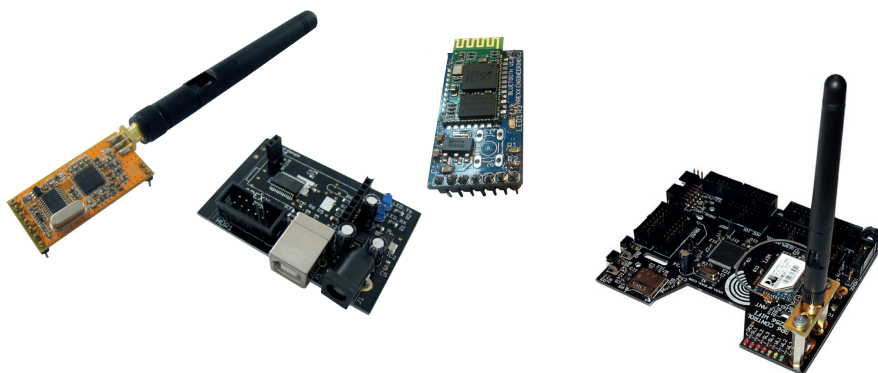
5.2. Modyfikacje między PCB wersji RA1 PRO i RA2-HOBBY

- Nowy konwerter TI TPS54332 DC/DC, 3500mA.
- Nowy stabilizator napięcia LDO dla 5-woltowych mikrokontrolerów.
- Dodatkowe złącze zasilające dla VCC i serwonapędu.
- 6 przedwzmacniaczy do pomiaru aktualnego prądu w serwonapędach poprzez rezystor bocznikowy.
- Przełącznik Reset jest teraz przełącznikiem Start/Stop - aby rozpocząć lub zakończyć pracę programu.
- Autostart (czyli automatyczne rozpoczęcie pracy programu po przełączeniu na Start) może zostać przez ciebie skonfigurowany poprzez RobotLoader.
- Dodatkowe piny wejścia/wyjścia mikrokontrolera ATMEGA64 na rozszerzenia.
- Połączenie magistralą I2C zgodne z modułami rozszerzającymi RP6 XBUS, na przykład moduł RP6-M256 WLAN.
- Gniazdo na zestaw komunikacji radiowej APC220 lub moduł Bluetooth.
- 4 niebieskie diody LED statusu zamiast pary diod: zielonej i czerwonej.
- Bezpiecznik 4A.
- Rezonator kwarcowy 16.000 MHz Crystal zamiast 16.384MHz.
- Rezystor pull-up magistrali I2C ulokowany jest teraz na płytce PCB.
- Przełącznik On/Off odłącza układ logiczny. Mikrokontroler steruje zasilaniem serwonapędu z przetwornicy DC/DC.
- Napięcie wejściowe wynosi 7 - 14 Volt (nieprzekraczalne max. 18 Volt).

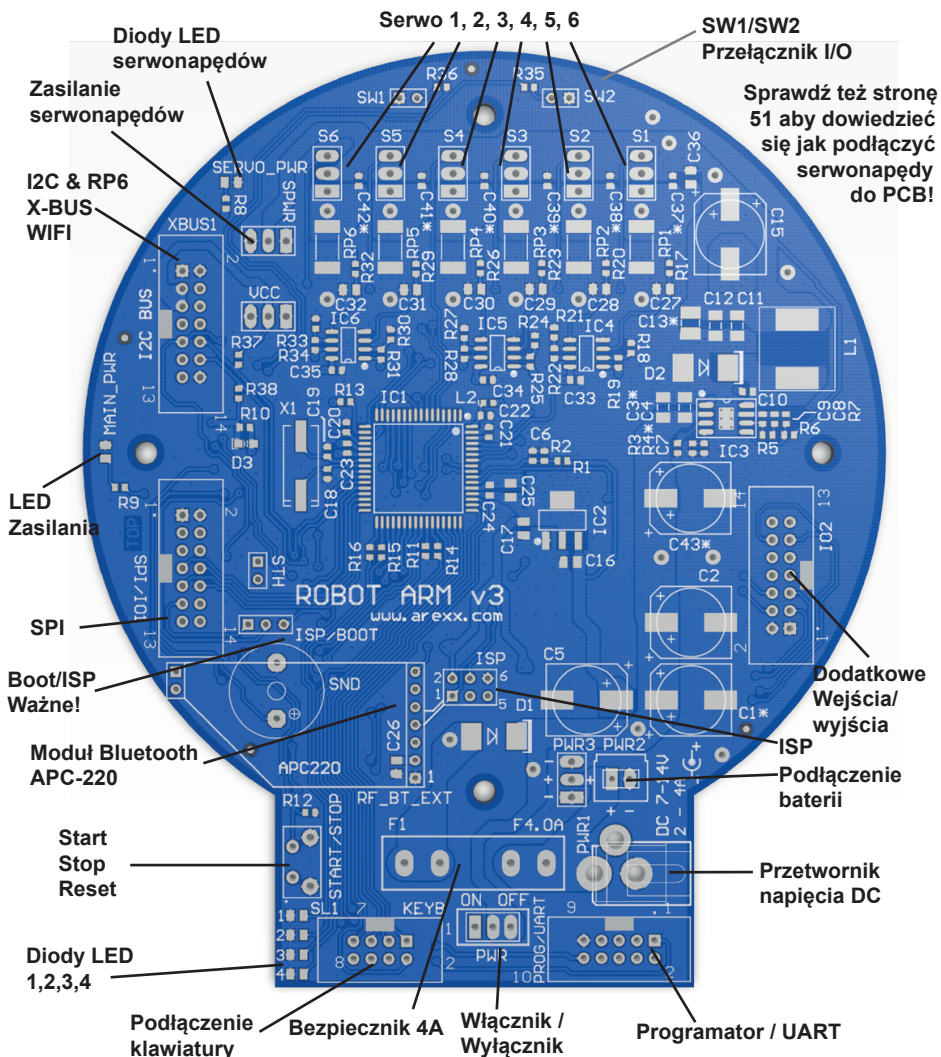
ROZSZERZENIA DLA RAMIENIA ROBOTA

- RP6v2-M256-WIFI: moduł do sterowania ramienia bezprzewodowo przez sieć
- ARX-APC220 for RACS 433 MHz: sterowanie bezprzewodowe
- ARX-BT03 dla systemu ANDROID: sterowanie bezprzewodowe bluetooth

Sprawdź też moduły AREXX APC-220, Android i WIFI!



5.3. Główna płytki PCB - najważniejsze elementy



5.4 DIODY LED

W tej sekcji zawarte jest wyjaśnienie niektórych diod LED znajdujących się na PCB. Żółta dioda "MAIN_PWR" pokazuje, czy przy wyłączonym zasilaniu robota dostarczane jest jakieś zewnętrzne napięcie.

Czerwona dioda "SERVO_PWR" zapala się tylko, gdy przetwornica DC/DC dla zasilania serwonapędu jest włączona, co kontrolowane jest z poziomu oprogramowania.

Na płytce są 4 niebieskie diody LED statusu "SL1-4" które sterowane są bezpośrednio z mikroprocesora. Można używać tych diod dowolnie przy programowaniu własnych aplikacji.

5.5. Początek pracy z robotem

1. Sprawdź montaż wszystkich elementów mechanicznych oraz elektronicznych robota i podłącz serwonapędy (str. 25 i 51).
2. Podłącz zasilanie 9 do 14V (**nie przekraczać 18 V!!**).
3. Włącz robot za pomocą przełącznika On/Off.

Źródło napięcia

Zasilacz

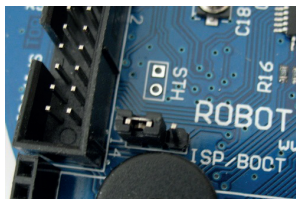
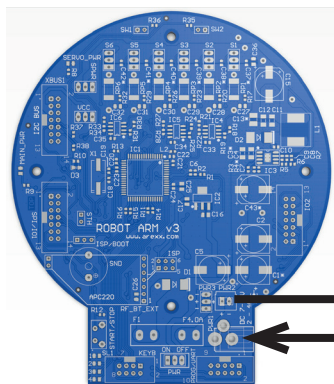
Istnieją dwa sposoby na zasilanie robota. Najprostszym z nich jest podłączenie do gniazdka zasilacza, którego napięcie wyjściowe wynosi 9-14V / 3-4 Amps (stałe - DC). W ten sposób sygnał jest podłączony na wejście regulatora napięcia.

Baterie

Innym sposobem jest podłączenie baterii (9-14V) w odpowiednim miejscu. Jeśli napięcie spadnie poniżej 6.7V, wyświetli się ostrzeżenie.

Ważne!

Zwarka "ISP/BOOT" powinna być ustawiona na lewo jak na obrazku poniżej.



Zaciski baterii
9 - 14 Volt

Zaciski DC
9 - 14 Volt

Jeśli robot jest podłączony do zasilania, serwonapędy lekko się poruszają i zapali się żółta dioda LED.

Zatem rozpoczęcie pracy nie jest takie trudne jak się wydaje! Teraz właśnie zacznie się ta trudniejsza część...!

Ale najpierw należy zainstalować oprogramowanie, co omówiono w rozdziale 6!

6. INSTALACJA OPROGRAMOWANIA

Zajmijmy się instalacją oprogramowania. Prawidłowe zainstalowanie jest bardzo ważne w każdym z następujących rozdziałów.

Zaloguj się w systemie jako administrator, ponieważ będziesz potrzebował jego uprawnień!

Zaleca się najpierw przeczytać cały rozdział, a później ponownie czytać krok po kroku i wykonywać polecenia.

Użytkownik musi znać podstawy obsługi systemu Windows lub Linux i być zaznajomiony z programami takimi jak: manager plików, przeglądarka internetowa, edytor tekstu, program do kompresji plików (WinZip, WinRAR, unzip etc.), powłoka systemowa Linux etc.! Jeśli twoja wiedza na temat komputerów jest dość uboga, poczytaj trochę na temat tych zagadnień zanim zaczniesz używać robota. Ten podręcznik nie jest przeznaczony do nauki podstaw informatyki, co jest pojęciem bardzo obszernym! Celem podręcznika jest ramię robota, potrzebne oprogramowanie i sposób programowania.

The Robot Arm CD-ROM

Już prawdopodobnie umieściłeś płytę w odtwarzaczu - jeśli nie, to zrób to teraz! W systemie Windows menu płyty powinno pojawić się chwilę po tym poprzez autostart. Jeśli nie, możesz to zrobić otwierając plik "start.htm" w przeglądarce internetowej, np. Firefox, w głównym katalogu CD poprzez manager plików. Pliki instalacyjne Firefox znajdują się też na płycie w katalogu:

```
<CD-ROM drive>:\Software\Firefox
```

Przeglądarka powinna być zaktualizowana przynajmniej do wersji 1.x (w przypadku Firefox) lub 6 (w przypadku Internet Explorer).

Po wyborze języka w menu CD, poza informacjami zawartymi w podręczniku, znajdziesz instrukcje (datasheet) oraz pomocne obrazki, a także pozycję w menu nazwaną "Software" (to wszystko jest również do pobrania z naszej strony domowej). Pozycja zawiera wszystkie narzędzia oprogramowania, sterownik USB oraz przykładowe programy z kodem źródłowym dla ramienia robota.

Zależnie od ustawień bezpieczeństwa twojej przeglądarki, możesz rozpocząć instalację programów bezpośrednio z CD!

Jeśli ustawienia bezpieczeństwa twojej przeglądarki internetowej nie pozwalają na bezpośrednią instalację z CD-ROM, musisz skopiować najpierw pliki do katalogu na twoim dysku twardym i rozpocznij instalację z tego miejsca. Aby uzyskać więcej szczegółów przeczytaj stronę w menu płyty. Inny sposób: przejdź do napędu CD poprzez manager plików i zainstaluj programy z płyty.

WinAVR - dla systemu Windows

Teraz rozpoczniemy instalację programu WinAVR. WinAVR jest - jak sama nazwa wskazuje - dedykowana jedynie dla systemu Windows!

Użytkownicy Linux mogą opuścić tę część.

WinAVR to zbiór wielu przydatnych i niezbędnych narzędzi do programowania mikrokontrolerów AVR w języku C. Oprócz GCC dla AVR (znane jako "AVR-GCC", o tym więcej później) WinAVR posiada bardzo wygodny edytor tekstu "Programmers Notepad 2", którego będziemy używać do programowania ramienia robota.

WinAVR jest dostępny w internecie za darmo. Aktualne wersje programu i dodatkowe informacje na jego temat znajdziesz na:

<http://winavr.sourceforge.net/>

Wsparcie programowe zapewnia również firma ATMEL z programem AVRStudio wraz z kompilatorem AVRGCC. Jednak uważamy, że Programmers Notepad lepiej odpowiada naszym celom. Plik instalacyjny WinAVR znajduje się na CD w folderze:

<CD-ROM drive>:\Software\AVR-GCC\Windows\WinAVR\

Instalacja WinAVR jest bardzo łatwa. Dla własnego użytku nie potrzebna jest zmiana ustawień, więc wystarczy klikać "Kontynuuj"!

Jeśli używasz Windows Vista lub Windows 7, musisz zainstalować najnowszą wersję WinAVR! Powinno również doskonale działać w Windows 2K i XP. Jeśli tak nie jest, spróbuj z jedną ze starszych wersji, które również znajdziesz na CD (zanim zaczniesz nową instalację WinAVR, musisz odinstalować wpięrw istniejącą wersję). Oficjalnie wersja systemu Win x64 (64-bitowa) nie jest jeszcze wspierana, ale w razie problemów płyta zawiera łątkę dla systemów Win x64! Więcej informacji na temat oprogramowania znajdziesz w menu płyty.

AVR-GCC, avr-libc uad avr-binutils - dla systemu Linux

(Użytkownicy Windows mogą przejść do następnej sekcji!)

Linux może wymagać większego wysiłku. Niektóre dystrybucje już zawierają wymagane paczki danych, ale to w większości wersje dość przestarzałe, dlatego powinieneś zainstalować nowsze wersje. Dystrybucji Linuxa jest tak wiele, że nie sposób wymienić je wszystkie. Są to m.in. SuSE, Ubuntu, RedHat/Fedora, Debian, Gentoo, Slackware, Mandriva etc., które też mają różne wersje, dlatego poradnik jest napisany bardziej ogólnie.

To samo tyczy się do innych sekcji systemu Linux w tym rozdziale!

Opisana procedura niekoniecznie musi działać dla twojej wersji. Zawsze lepiej więc wspomóc się zasobami internetu np. wyszukując "<LinuxDistribution> avr gcc". (Spróbuj inne kombinacje). To samo tyczy się do innych sekcji dla systemu Linux - zapewnimy pewne słowa kluczowe, które pomogą szukać! Jeśli napotkasz problemy w instalacji AVR-GCC, możesz zasięgnąć pomocy na naszym forum lub na jednym z licznych forum użytkowników Linux. Pierwszą czynnością jest odinstalowanie tej wersji avr-gcc, avr-binutils oraz avr-libc, które masz na dysku twardym, ponieważ są nieaktualne. Możesz je odnaleźć poprzez package manager lub szukając hasła "avr" i odinstaluj te trzy wcześniej wspomniane komponenty. Czy te pliki istnieją możesz łatwo sprawdzić wpisując w konsolę:

```
> which avr-gcc
```

Jeśli ścieżka się wyświetla, wersja jest zainstalowana. Wpisz zatem:

```
> avr-gcc --version
```

Jeśli dana wersja zawiera się w przedziale 3.4.6 - 4.1.0, spróbuj skompilować programy (w następnym rozdziale). Jeśli nie udało się, zainstaluj nowe narzędzia. W dalszej części instalować będziemy najnowszą wersję 4.1.1 (na marzec 2007) wraz z ważnymi łańkami.

W przypadku, gdy w package manager nie pojawia się ścieżka `avr-gcc`, wersja została odinstalowana. Należy jeszcze usunąć istotne pliki binarne: wyszukaj w katalogach `all/bin`, `/usr/bin` etc. plików zaczynających się na `"avr"` i usuń je (TYLKO te pliki i nic ponadto!). W końcu katalogi takie jak `/usr/avr` czy też `/usr/local/avr` również muszą zostać wymazane.

Ważne: Upewnij się, że narzędzia programistyczne takie jak GCC, make, binutils, libc, etc. są zainstalowane przed kompilacją oraz instalacją narzędzi! Najlepiej zrobić to poprzez package manager twojej dystrybucji. Każda z dystrybucji Linux powinna zawierać wymagane paczki danych z płyty CD lub z internetu, gdzie również są dostępne.

Upewnij się, że program `"texinfo"` jest zainstalowany. Jeśli nie, zainstaluj wymagane pakiety, bo inaczej nie zadziała.

Po wykonaniu tego kroku, możesz rozpocząć instalację.

Masz teraz dwa wyjścia: zrobić wszystko ręcznie lub użyć prostego w obsłudze skryptu instalacyjnego.

Zalecamy użyć najpierw skryptu. Jeśli nie zadziała, wciąż możesz zainstalować kompilator ręcznie.

Uwaga:

Powinieneś mieć wystarczająco wolnego miejsca na dysku twardym! Chwilowo wymagane jest więcej niż 400Mb. Po instalacji wymazane zostanie ponad 300Mb, jednak podczas instalowania potrzebne jest nieco więcej miejsca.

Wiele z następnych kroków instalacji wymaga specjalnych praw (**ROOT RIGHTS**), więc zaloguj się jako użytkownik uprzywilejowany `"su"` lub wykonaj `"sudo"` etc. (skrypt instalacyjny, `mkdir` w katalogu `/usr/local` i wykonaj instalację).

Zwróć uwagę na nazwy komend i wypisz je DOKŁADNIE tak, jak napisano dalej!

Każdy znak jest ważny i nawet, jeśli niektóre komendy wydają ci się dość dziwne, wszystkie są poprawne i upewnij się, że przy przepisywaniu nie popełniłeś błędu! (<CD-ROM-drive> ma być oczywiście zastąpione ścieżką stacji CD-ROM!).

Folder na CD:

```
<CD-ROM drive>:\Software\avr-gcc\Linux
```

zawiera wszystkie istotne pliki dla avr-gcc, avr-libc oraz binutils. Najpierw, należy skopiować wszystkie pliki instalacyjne do katalogu na twoim twardym dysku - **to się tyczy obu metod instalacji!** Użyjemy katalogu domowego (zazwyczaj skrótem do aktualnego katalogu domowego jest tylda: „~”):

```
> mkdir ~/Robot Arm  
> cd <CD-ROM drive>/Software/avr-gcc/Linux  
> cp * ~/Robot Arm
```

Po udanej instalacji możesz usunąć pliki w celu zaoszczędzenia miejsca!

Skrypt automatycznej instalacji

Jeśli raz zrobisz skrypt wykonywalnym za pomocą polecenia `chmod`, możesz zacząć od razu:

```
> cd ~/Robot Arm
> chmod -x avrgcc_build_and_install.sh
> ./avrgcc_build_and_install.sh
```

Na pytanie czy chcesz zainstalować w tej konfiguracji odpowiedz wciskając "y".

UWAGA: Kompilacja i instalacja zajmie trochę czasu (zależy on od mocy obliczeniowej twojego systemu - np. około 15 min. przy dwurdzeniowym procesorze 2GHz Core Duo Notebook. Naturalnie potrwa to dłużej dla wolniejszych systemów).

Skrypt zawiera również pewne łatki. Są to pliki o rozszerzeniu `.diff`. Jeśli instalacja zakończyła się sukcesem, wyświetli się następująca wiadomość:

```
(./avrgcc_build_and_install.sh)
(./avrgcc_build_and_install.sh) installation of avr GNU tools complete
(./avrgcc_build_and_install.sh) add /usr/local/avr/bin to your path to use the avr
GNU tools
(./avrgcc_build_and_install.sh) you might want to run the following to save disk
space:
(./avrgcc_build_and_install.sh)
(./avrgcc_build_and_install.sh) rm -rf /usr/local/avr/source /usr/local/avr/build
```

Zgodnie z poradami, możesz wykonać

```
rm -rf /usr/local/avr/source /usr/local/avr/build
```

Ta instrukcja wymarże wszystkie chwilowe pliki, które nie będą już potrzebne.

Możesz opuścić następny akapit i ustawić ścieżkę narzędzi `avr`.

Jeśli wykonanie skryptu się nie powiedzie, musisz uważnie przeczytać wiadomość z błędem (jeśli potrzeba, idź w górę okna korzystając z paska przesuwania). Najczęściej jest to kwestia brakującego programu, który powinien być zainstalowany wcześniej (taki jak np. plik `texinfo`). Zanim kontynuujesz po błędzie, zaleca się wymazać już wygenerowane pliki standardowej instalacji w katalogu `"/usr/local/avr"` – najlepiej cały katalog.

Jeśli nie wiesz co dokładnie poszło źle, zapisz linie komend w pliku oraz skontaktuj się ze wsparciem technicznym. Prosimy, żeby zawierać możliwie jak najwięcej informacji na temat niepowodzenia. W ten sposób łatwiej będzie rozwiązać twój problem.

GCC dla AVR

Kompilator GCC jest "połatany", skompilowany i zainstalowany trochę jak binutils:

```
> cd ~/Robot Arm> bunzip2 -c gcc-4.1.1.tar.bz2 | tar xf -
> cd gcc-4.1.1
> patch -p0 < ../gcc-patch-0b-constants.diff
> patch -p0 < ../gcc-patch-attribute_alias.diff
> patch -p0 < ../gcc-patch-bug25672.diff
> patch -p0 < ../gcc-patch-dwarf.diff
> patch -p0 < ../gcc-patch-libiberty-Makefile.in.diff
> patch -p0 < ../gcc-patch-newdevices.diff
> patch -p0 < ../gcc-patch-zz-atmega256x.diff
> mkdir obj-avr
> cd obj-avr
> ../configure --prefix=$PREFIX --target=avr --enable-
languages=c,c++ \
--disable-nls --disable-libssp --with-dwarf2
> make
> make install
```

Po znaku \ wciśnij po prostu Enter i kontynuuj pisanie. W ten sposób komendy są rozszerzone na kilka linijek, ale możesz też o tym zapomnieć.

AVR Libc

I ostatni, ale nie mniej ważny AVR libc:

```
> cd ~/Robot Arm
> bunzip2 -c avr-libc-1.4.5.tar.bz2 | tar xf -
> cd avr-libc-1.4.5
> ./configure --prefix=$PREFIX --build=`./config.guess` --host=avr
> make
> make install
```

Ważne: at `-build=`./config.guess`` upewnij się, że nie zapomniałeś o znaku ``` (na klawiaturze klawisz z tyldą) a nie zamiast niego zwykły znak apostrofu, ponieważ tak napisana instrukcja nie zadziała.

Ustaw ścieżkę

Musisz się upewnić, że katalog `/usr/local/avr/bin` jest zapisany w zmiennej ścieżce, bo inaczej niemożliwym będzie pobranie `avr-gcc` z konsoli albo z plików `makefile`. Na koniec musisz wprowadzić ścieżkę `/etc/profile` lub `/etc/environment` lub podobną (różne zależne od dystrybucji) – oddzielone dwukropkiem `“:”` od już istniejących pozycji. W pliku może to wyglądać na przykład tak:

```
PATH="/usr/local/bin:/usr/bin:/bin:/usr/X11R6/bin:/usr/local/avr/bin"
```

Teraz wprowadź w konsoli "avr-gcc --version" jak opisano powyżej. Jeśli zadziała, instalacja powiodła się!

Instalacja ręczna

Jeśli wolisz zainstalować kompilator ręcznie lub instalacja za pomocą skryptu nie przebiegła pomyślnie, podążaj na instrukcjami poniżej.

Opis powstał na bazie artykułu dostępnego pod linkiem:

http://www.nongnu.org/avr-libc/user-manual/install_tools.html

Jest on również dostępny na CD w formacie PDF w dokumentacji AVR Libc:

<CD-ROM drive>:\Software\Documentation\avr-libc-user-manual-1.4.5.pdf

Opis w podręczniku jest znacznie krótszy, ale zawiera kilka ważnych łątek, bez których kilka rzeczy może nie działać poprawnie.

Najpierw należy stworzyć katalog, w której zainstalowane zostaną wszystkie narzędzia. Powinien nim być: /usr/local/avr.

W konsoli również wprowadź (jako użytkownik uprzywilejowany):

```
> mkdir /usr/local/avr  
> mkdir /usr/local/avr/bin
```

To niekoniecznie musi być właśnie ten katalog. Po prostu należy stworzyć zmienną \$PREFIX dla tego katalogu:

```
> PREFIX=/usr/local/avr  
> export PREFIX
```

Musi to zostać dodane do zmiennej ścieżki:

```
> PATH=$PATH:$PREFIX/bin  
> export PATH
```

Binutils dla AVR

Teraz należy wypakować kod źródłowy binutils i dodać kilka łatek. W naszym przykładzie zakładamy, że skopiowałeś wszystko do katalogu domowego ~/Robot Arm:

```
> cd ~/Robot Arm
> bunzip2 -c binutils-2.17.tar.bz2 | tar xf -
> cd binutils-2.17
> patch -p0 < ../binutils-patch-aa.diff
> patch -p0 < ../binutils-patch-atmega256x.diff
> patch -p0 < ../binutils-patch-coff-avr.diff
> patch -p0 < ../binutils-patch-newdevices.diff
> patch -p0 < ../binutils-patch-avr-size.diff
> mkdir obj-avr
> cd obj-avr
```

Teraz wykonaj skrypt:

```
> ../configure --prefix=$PREFIX --target=avr --disable-nls
```

Ten skrypt wykrywa co jest w twoim systemie dostępne i generuje odpowiednie pliki makefile. Teraz binutils mogą być skompilowane oraz zainstalowane:

```
> make
> make install
```

Zależnie od mocy obliczeniowej twojego systemu, może to zająć kilka minut. Tyczy się to również do dwóch kolejnych sekcji, szczególnie do sekcji z GCC!

Java 6

RobotLoader (zobacz Info poniżej) został stworzony dla platformy Java i pasuje na systemy Windows i Linux (teoretycznie również na systemy operacyjne takie jak OS X, ale AREXX Engineering niestety jeszcze oficjalnie nie wspiera innych systemów). Aby działało, należy zainstalować aktualną wersję Java Runtime Environment (JRE). Zazwyczaj jest to już zainstalowane na komputerze, jednak musi być to wersja co najmniej 1.6 (= Java 6)! Jeśli na twoim komputerze nie ma JRE lub JDK, należy zainstalować JRE 1.6 z SUN Microsystems lub zamiennie pobierz nowszą wersję z:
<http://www.java.com> or <http://java.sun.com>.

Windows

JRE 1.6 dla Windows powinien być w poniższym folderze:

```
<CD-ROM drive>:\Software\Java\JRE6\Windows\
```

Instalacja Java w systemie Windows jest bardzo łatwa. Należy rozpocząć instalację i wykonywać instrukcje na ekranie - i gotowe. Możesz pominąć kolejny akapit.

Linux

W systemie Linux instalacja nie przysparza dużych problemów, jednak niektóre dystrybucje wymagają odrobiny pracy ręcznej.

W folderze:

```
<CD-ROM drive>:\Software\Java\JRE6\
```

znajdziesz JRE 1.6 jak RPM (SuSE, RedHat etc.) i jako samorozpakowujące się archiwum "bin". Pod systemem Linux lepiej będzie poszukać pakietów Java w package manager twojej dystrybucji (słowa kluczowe np. „java”, „sun”, „jre”, „java6” ...) i użyć pakietów twojej dystrybucji niż zawartych na CD-ROM! Upewnij się jednak, że zainstalowana jest wersja Java 6 (=1.6) lub nowsza!

Na Ubuntu lub Debian, archiwum RPM nie działa bezpośrednio. Będziesz musiał użyć pakietu package manager swojej dystrybucji, aby znaleźć odpowiedni pakiet instalacyjny. RPM powinien jednak działać dobrze również w innych dystrybucjach, np. RedHat/Fedora i SuSE. Jeśli nie, wciąż możesz rozpakować JRE (np. do /usr/lib/Java6) z archiwum samorozpakowującego się (.bin) i ustawić ręcznie ścieżki do JRE (PATH oraz JAVA_HOME etc.).

Odnieś się do instrukcji instalacyjnych z Sun, które również znajdziesz w wyżej wymienionym katalogu oraz na stronie Java (zobacz powyżej).

Możesz sprawdzić czy Java zainstalowała się poprawnie wpisując komendę "java-version" w oknie konsoli. Wiadomość wyjściowa powinna być przybliżona do poniższej:

```
java version "1.6.0"  
Java(TM) SE Runtime Environment (build 1.6.0-b105)  
Java HotSpot(TM) Client VM (build 1.6.0-b105, mixed mode, sharing)
```

Jeśli otrzymana wiadomość całkowicie się nie zgadza, być może zainstalowana została nieodpowiednia wersja lub inna JavaVM została już zainstalowana w twoim systemie.

RobotLoader

RobotLoader został przygotowany, aby w szybki i prosty sposób przesyłać nowe programy i wszystkie rozszerzenia do robota (dopóki moduły są kompatybilne z bootloaderem). Zawiera ponadto kilka przydatnych funkcji, np. prosty program terminala.

Instalacja RobotLoadera nie jest konieczna. Po prostu skopiuj program do folderu na dysku twardym.

```
<CD-ROM drive>:\Software\RobotLoader\RobotLoader.zip
```

Wypakuj program gdzieś na dysku twardym, np. w nowym folderze C:\Programme\RobotLoader. Folder zawiera plik RobotLoader.exe, który uruchamia się, gdy klikniesz na ikonę dwa razy.

Sam program RobotLoader jest w archiwum Java (JAR) RobotLoader_lib.jar. Zamiennie można rozpocząć poprzez wprowadzenie komendy:

Windows:

```
java -Djava.library.path=".\\lib" -jar RobotLoader_lib.jar
```

Linux:

```
java -Djava.library.path="./lib" -jar RobotLoader_lib.jar
```

Opcja wywoływana poprzez -D jest konieczna aby pozwolić JVM na znalezienie wszystkich użytych bibliotek. Windows tego nie wymaga i możesz po prostu rozpocząć wybierając plik .exe. Linux żąda skryptu powłoki systemowej "RobotLoader.sh". Może to być konieczne, by skrypt dało się wykonać (chmod -x ./RobotLoader.sh). Po tym możesz rozpocząć w konsoli poprzez "./RobotLoader.sh".

Rozsądnie jest stworzyć skrót na pulpicie lub w menu start, aby uruchamianie Robot Loadera było wygodniejsze. W systemie Windows kliknij prawym przyciskiem myszy na pliku RobotLoader o rozszerzeniu .exe, a następnie kliknij na "Pulpit (utwórz skrót)" w menu "Wyślij do".

Biblioteka Robot Arm, biblioteka Robot Arm CONTROL oraz przykładowe programy

Biblioteka Robot Arm i związane z nią przykładowe programy są spakowane w archiwum zip na płycie:

```
<CD-ROM drive>:\Software\Robot Arm Examples\Robot ArmExamples  
[MINI].zip
```

Dla swojej wygody po prostu rozpakuj je bezpośrednio do katalogu na dysku twardym. Zaleca się rozpakowanie przykładowych programów do folderu "My files" w podfolderze "Robot Arm\Examples\", a w systemie Linux w katalogu domowym. Zależy to od ciebie.

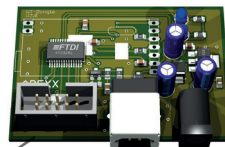
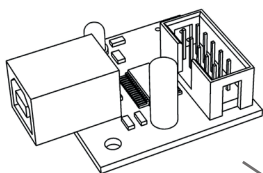
W rozdziale dotyczącym programów będą opisane osobno przykładowe programy!

7. Programator i Loader

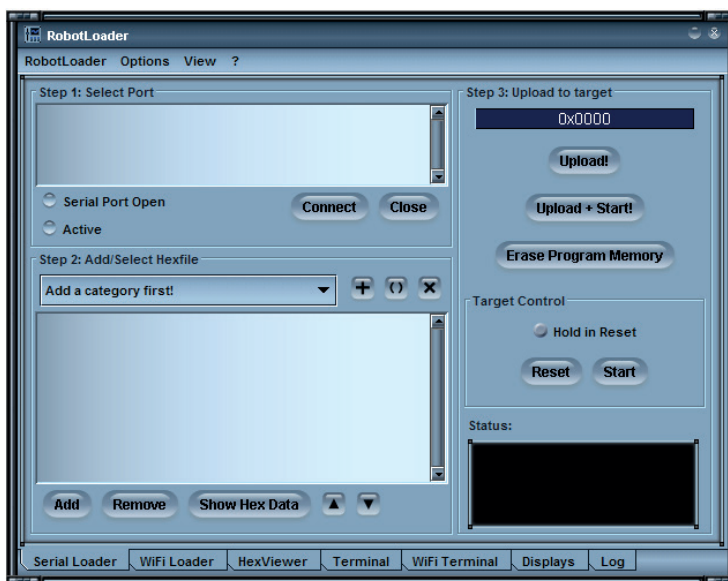
Do wgrania programu HEX z komputera do robota użyty zostanie konwerter USB i nasz program RobotLoader.

Zawarty w zestawie został konwerter USB, który jest elementem pozwalającym na połączenie niezgodnych ze sobą urządzeń. Należy z jednej strony podłączyć go do portu USB w komputerze, a z drugiej strony do portu Prog/UART ramienia robota na płytce PCB.

Aktualnie wgrany do robota program automatycznie usuwa poprzedni program.



Konwerter USB



Program RobotLoader

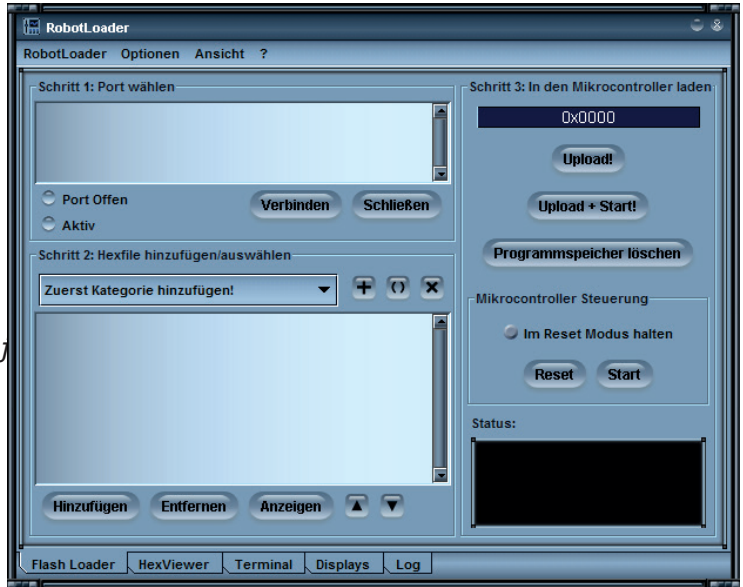
7.1. RobotLoader

RobotLoader został stworzony, by móc wgrywać nowe programy do ramienia robota i wszystkich innych naszych robotów w bardzo prosty sposób! (pod warunkiem, że bootloader jest kompatybilny).

RobotLoader

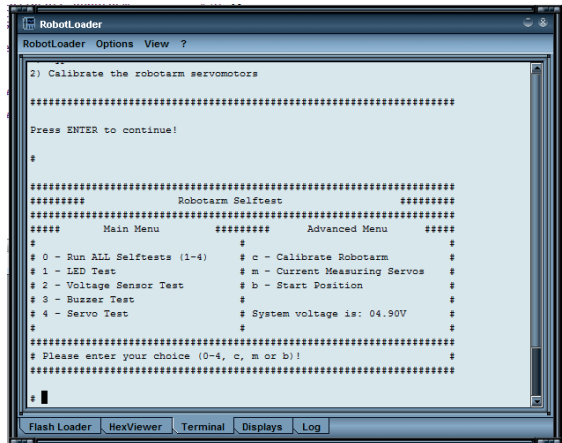


Jeśli napięcie spadnie poniżej < 6.7 V, wyświetla się ostrzeżenie.



Wbudowane są pewne dodatkowe funkcje, jak na przykład prosty program terminala.

Okno terminala



Aby RobotLoader pracował nie musi być zainstalowany. Skopiuj jedynie program do nowego folderu na dysku twardym.

7.2. Podłączenie interfejsu USB – Windows

Użytkownicy systemu Linux mogą przejść do kolejnej sekcji!

Jest kilka sposobów na zainstalowanie interfejsu USB, lecz najprostszym z nich będzie zainstalować **sterownik ZANIM podłączony zostanie sprzęt!**

CD zawiera program do instalacji drivera.

Dla wersji **32 i 64 Bit systemów Windows 7, XP, Vista, Server 2003 i 2000:**

```
<CD-ROM drive>:\Software\USB_DRIVER\Win2k_XP\CDM_Setup.exe
```

Dla systemów **Win98SE/Me**, tak łatwy sposób niestety nie istnieje. Należy zainstalować sterownik starszej wersji po podłączeniu sprzętu (zobacz poniżej).

Uruchom program instalacyjny. Pojawi się jedynie krótkie potwierdzenie pomyślnej instalacji i to wszystko.

Teraz możesz podłączyć USB do komputera. **JESZCZE NIE PODŁĄCZAJ KONWERTERA DO ROBOTA!** Podłącz konwerter jedynie do komputera kablem USB. Pamiętaj o bezpiecznym trzymaniu płytki PCB interfejsu USB za brzegi lub za wejście USB (przeczytaj instrukcje bezpieczeństwa dla elementów elektronicznych)! Aby uniknąć wyładowań elektrostatycznych **NIE** dotykaj żadnych elementów na płytce PCB, miejsc lutu ani złącz standardu IDE jeśli nie jest to absolutnie konieczne!

Sterownik zainstalowany w poprzednim kroku będzie używany automatycznie przez urządzenie bez twojej pomocy. Pod systemem Windows XP/2k pojawią się powiadomienia nad zasobnikiem paska zadań. Ostatnia z wiadomości powinna brzmieć: "The device has been successfully installed and is ready for use!" (urządzenie zostało zainstalowane i jest gotowe do użycia).

Jeśli podłączyłeś interfejs USB przed instalacją (lub pracujesz na systemie Win98/Me) – nie jest to aż tak ważne. Windows sam spyta o sterownik. Taka metoda instalacji jest możliwa, ponieważ sterownik także znajduje się na CD w nierozpakowanym formacie.

W tym przypadku pojawi się dialog (w Windows), aby zainstalować nowy driver. Należy wskazać ścieżkę, w której się driver znajduje. Pod Windows 2k/XP należy najpierw wybrać instalację ręczną. Na CD sterownik znajduje się w katalogu wymienionym na poprzedniej stronie.

Po prostu wprowadź katalog swojej wersji Windows i być może kilku innych plików, których system sam automatycznie nie odnajdzie (wszystkie są wymienione poniżej!).

W systemie Windows XP i wersjach późniejszych zazwyczaj pojawia się wiadomość o tym, że sterowniki FTDI nie są zweryfikowane przez Microsoft. Nie jest to ważne i bez problemu można potwierdzić komunikat.

Obsługa

W systemach 32 i 64 Bit Windows 7/8, XP, Vista, Server 2003 oraz 2000:

```
<CD-ROM drive>:\Software\USB_DRIVER\Win2k_XP\FTDI_CDM2\
```

Dla starszych systemów **Windows 98SE/Me:**

```
<CD-ROM drive>:\Software\USB_DRIVER\Win98SE_ME\FTDI_D2XX\
```

Po instalacji sterownika na starszych wersjach systemu Windows (np. Win98SE) może być konieczny restart komputera! UWAGA: pod Win98/Me działa tylko jeden ze sterowników: Virtual Comport lub D2XX dla FTDI! Niestety żaden sterownik nie zapewni pracy ich obu.

Sprawdź połączenie z urządzeniem

Aby sprawdzić czy sterownik został poprawnie zainstalowany możesz użyć Managera urządzeń: kliknij prawym przyciskiem myszy na: Mój Komputer --> Właściwości --> Sprzęt --> Manager urządzeń

LUB zamiennie: Start --> Ustawienia --> Panel sterowania --> Wydajność i konserwacja --> System --> Sprzęt --> Manager urządzeń i sprawdź czy pod "Porty (COM i LPT)" znajduje się "USB-Serial Port (COMX)" - gdzie X odpowiada numerowi portu, lub czy pod "Kontrolery uniwersalnej magistrali szeregowej" jest rozwinięcie "Konwerter szeregowy USB" !

Jeśli pewnego dnia zechcesz odinstalować sterownik

Jeśli nadarzy się taka sytuacja (nie teraz - to tylko wskazówka na przyszłość): Jeśli użyłeś programu instalacyjnego z CD ROM, możesz odinstalować to bezpośrednio poprzez menu Start --> Ustawienia --> Panel sterowania --> Dodaj lub Usuń programy. Na liście programów odszukaj pozycję "FTDI USB Serial Converter Drivers" – wybierz i kliknij "odinstaluj".

Jeśli sterownik został zainstalowany ręcznie, możesz uruchomić program ""FTUNIN.exe" w katalogu poświęconemu sterownikowi USB! Ostrzeżenie: do adaptorów USB-->RS232 z chipem FTDI również często używa się tego sterownika!

7.3. Podłączenie interfejsu USB – Linux

Użytkownicy Windows mogą opuścić tę sekcję!

System Linux z jądrem 2.4.20 lub wyższy już zawiera wymagany sterownik (przynajmniej dla poprzedniego kompatybilnego modelu chipu FT232BM naszego modelu interfejsu USB, którym jest FT232R). Sprzęt rozpoznawany jest automatycznie i nie musisz robić nic więcej. W przypadku jakichkolwiek problemów, możesz pobrać sterowniki na system Linux (oraz wsparcie i być może nowsze sterowniki) bezpośrednio ze strony FTDI:

<http://www.ftdichip.com/>

Jeśli już sprzęt zostanie połączony, możesz to sprawdzić pod systemem Linuxem poprzez:
`cat /proc/tty/driver/usbserial`

jeśli port szeregowy USB został zainstalowany poprawnie. Zazwyczaj nic więcej nie musisz robić.

Warto wspomnieć, że RobotLoader pod systemem Windows używa sterowników D2XX i pełny opis USB pojawi się w liście portów (np. "USB0 | Robot USB Interface | serialNumber"). Podczas gdy pod Linuxem opisy wirtualnych comportów pojawiają się jako `/dev/ttyUSB0`, `/dev/ttyUSB1` etc.. Zwykle porty com wyświetlają się tak samo "dev/ttyS0" etc.. W tym przypadku musisz próbować aż znajdziesz poprawny port.

Niestety Linux nie posiada tak wygodnego sterownika, który robi obie rzeczy. Dlatego też lepiej jest użyć sterowników Virtual Comport, które i tak są już zawarte w jądrze systemu. Instalacja sterownika D2XX może wymagać sporo pracy ręcznej...

Zakończenie instalacji oprogramowania

Instalacja oprogramowania i interfejsów USB jest zakończona! Musisz jedynie skopiować najważniejsze pliki z CD na dysk twardy (przede wszystkim cały folder "Documentation" oraz, jeśli do tej pory nie były skopiowane, przykładowe programy). Dzięki temu nie będziesz musiał wciąż sięgać po CD, gdy będziesz potrzebował plików.

Jeśli kiedyś CD się zapodzieje, najważniejsze pliki (jak ten podręcznik, RobotLoader, przykładowe programy) możesz pobrać ze strony domowej AREXX. Możesz tam również odnaleźć linki do paczek innych programów, których będziesz potrzebować.

7.4. Test interfejsu USB i uruchamianie RobotLoadera

Następnym krokiem jest test wgranego programu poprzez interfejs USB. Podłącz interfejs USB do PC (zawsze najpierw podłącz stronę PC!), a drugim końcem 10-pinowym kablem taśmowym do modułu "PROG/UART", który z kolei podłącza się do robota. (Robot Arm MUSI BYĆ WYŁĄCZONY!) Taśma jest mechanicznie zabezpieczona przed odwróceniem polaryzacji. Dopóki nie ma wymuszenia, nie ma możliwości na złe podłączenie kabla.



Następnie włącz RobotLoadera. W zależności od wybranego języka, nazwy menu mogą nieco się różnić. Poprzez menu "Opcje->Preferencje" ("Options->Preferences") pod "Language /Sprache" możesz wybrać język (spośród angielskiego i niemieckiego) i wybór zatwierdź klikając OK.

Jeśli wybrany został język, należy zrestartować program RobotLoader w celu zatwierdzenia zmian!

Otwarcie portu - Windows



Wybierz port USB. Dopóki żaden inny konwerter szeregowy USB z chipem FTDI jest podłączony do PC, zobaczysz tylko jedną wybraną pozycję.

Jeśli pojawia się więcej portów, możesz je zidentyfikować poprzez nazwę "Robot USB Interface" (lub „FT232R USB UART”). Za nazwą portu widnieje numer seryjny.

Jeśli żaden z portów się nie wyświetla, należy odświeżyć listę portów klikając w menu "RobotLoader-->Refresh Portlist" !



UWAGA!

Jeśli napięcie spadnie poniżej < 6,7 V, wyświetla się ostrzeżenie.

7.5. Otwieranie portu – Linux

Linux traktuje szeregowy adaptor USB jako zwykły port com. Instalacja sterownika D2XX z FTDI może nie być taka prosta jak instalacja pod Linuxem, a sterowniki normalnych wirtualnych portów com (VCP) i tak są już zawarte w jądrach systemów Linux. Działa to niemal tak samo jak pod Windowsem. Musisz jedynie znaleźć nazwę interfejsu USB ramienia robota i upewnić się, że port nie zostanie odłączony od PC dopóki połączenie nie zostanie otwarte (inaczej będzie trzeba zrestartować RobotLoader i połączyć się ponownie). Pod Linuxem nazwy wirtualnych portów com to `"/dev/ttyUSBx"`, gdzie x to numer, np. `"/dev/ttyUSB0"` lub `"/dev/ttyUSB1"`. Nazwy zwykłych portów com pod Linuxem nazwane są: `"/dev/ttyS0"`, `"/dev/tty- S1"` itd.. One również pokazują się na liście portów tak długo, jak istnieją.

Jeśli jest kilka portów, RobotLoader zamięta który był używany ostatnio i wybiera go automatycznie przy starcie programu (większość ustawień dokonanych w programie jest zachowywana).

Teraz należy kliknąć na przycisk "Connect" (połącz)! RobotLoader otworzy port i sprawdzi, czy komunikacja z bootloaderem działa poprawnie. Czarne pole u dołu ("Status") powinno pokazywać wiadomość "Connected to: Robot Arm ..." lub podobną wraz z informacją na temat aktualnie zmierzonego napięcia. Jeśli tak się nie dzieje, spróbuj ponownie! Jeśli jednak dalej nie działa, popełniony został gdzieś po drodze błąd! Natychmiast wyłącz robota i rozpocznij poszukiwanie błędu.

Jeśli napięcie jest za niskie, pojawia się ostrzeżenie. Należy naładować akumulatory (najlepiej zrobić to jeszcze zanim napięcie spadnie poniżej 6,7V)!

7.6. AUTOTEST

Żółta dioda LED zapala się, gdy ramię robota jest włączone.

Dioda statusu LED mignie, gdy wgrany zostanie plik HEX.

Jeśli zadziała, możesz uruchomić niewielki program autotestu, aby sprawdzić działanie wszystkich układów robota. W tym celu należy kliknąć guzik "Add" u dołu okna RobotLoadera i wybrać plik RobotArmExamples [MINI], „Example_11_Selftest\RobotArm_Self-test.hex” w katalogu przykładowych programów. Plik zawiera program autotestu w formacie hexadecymalnym - właśnie dlatego ten typ plików z programem nazywa się "plikiem hex". Wybrany plik pokaże się później na liście. W ten sposób możesz dodać inne pliki hex programów, które napisałeś sam, a także z programów przykładowych (sprawdź zrzuty ekranu, gdzie dodano już pliki hex). RobotLoader jest w stanie wczytać z kilkoma rodzajami plików hex.

Pomaga to w segregowaniu plików, np. jeśli zamontowane są do robota inne programowalne moduły rozszerzeń, które mogą używać innych wersji programów. Lista zapisuje się automatycznie na końcu programu. Oczywiście zapisują się jedynie ścieżki do tych plików hex, a nie sam plik hex. Jeśli pracujesz nad programem, plik hex dodajesz i wybierasz tylko raz. Później po każdym re-kompilowaniu możesz załadować nowy program do mikrokontrolera (można również użyć skrótów klawiszowych [STRG+D] lub [STRG+Y], aby rozpocząć program zaraz po przesłaniu go do mikrokontrolera). Nazwy ścieżki są oczywiście zupełnie inne w różnych systemach operacyjnych. Mimo to RobotLoader odpowiada obu systemom, Windows i Linux, bez dokonywania żadnych zmian, ponieważ istnieje osobna lista dla Windowsa i Linuxa.

Możesz teraz kontynuować z innymi przykładowymi programami ramienia robota lub rozpocząć programowanie samemu.



UWAGA!

Jeśli napięcie spadnie poniżej < 6,7 V, wyświetla się ostrzeżenie.

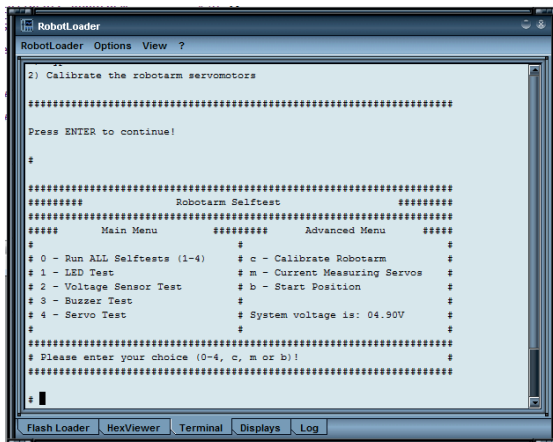
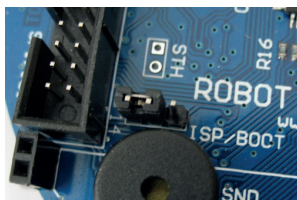
Należy wybrać z listy plik "RobotArm_Selftest.hex" i kliknąć na przycisk "Upload!" (wgraj), który znajdziesz na górze tuż pod pas-kiem postępu.

Teraz program zostanie wgrany do procesora MEGA64 na płytce PCB ramienia robota. Powinno to zająć nie więcej jak kilka sekund (dla programu autotestu max. 5 sekund).

Przełącz na zakładkę "Terminal" (na dole okienka)! Inny sposób na wybranie terminala: poprzez opcję "View" w menu.

WAŻNE!

Zworka "ISP/BOOT" powinna być ustawiona na lewą stronę jak na obrazku poniżej!



Teraz możesz wykonać autotest oraz kalibrację ramienia robota. Aby rozpocząć program przełącz switch Start/Stop Reset na robocie. Można to zrobić również poprzez menu RobotLoadera --> Start lub kombinacją klawiszy [STRG]+[S]. Jednak tym razem sprawdź, czy przełącznik działa poprawnie!

Jeśli w autoteście wystąpi błąd, odłącz natychmiast robota i zacznij szukać błędów.

ZALECA SIĘ ROZPOCZĘCIE OD KALIBRACJI RAMIENIA ROBOTA! ZOBACZ NA STRONIE 50.



UŻYWAJ JEDYNIIE ODPOWIEDNIEGO OPROGRAMOWANIA (DLA V3 PRO LUB MINI)!

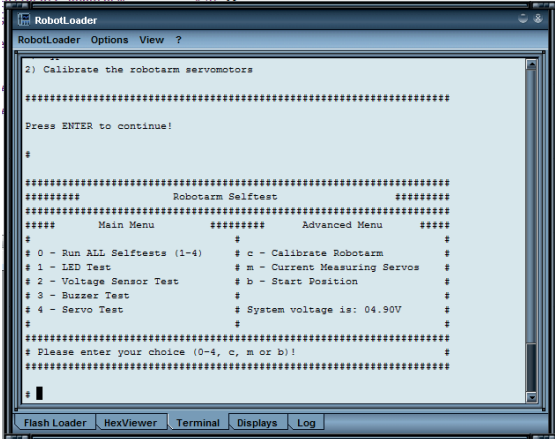
7.7. Kalibracja

Uruchom program do kalibracji, aby skalibrować ramię robota.

W tym celu należy kliknąć przycisk "Add" u dołu okna RobotLoadera i wybrać plik RobotArmExamples [MINI], „Example_11_Selftest\RobotArm_Selftest.hex” w katalogu plików przykładowych.

Plik zawiera program autotest w formacie heksadecymalnym. Wybrany plik pojawi się później na liście (jak na zrzucie poniżej).

Wybierz C
(C - calibrate)
w programie
do kalibracji, aby zacząć
kalibrację.

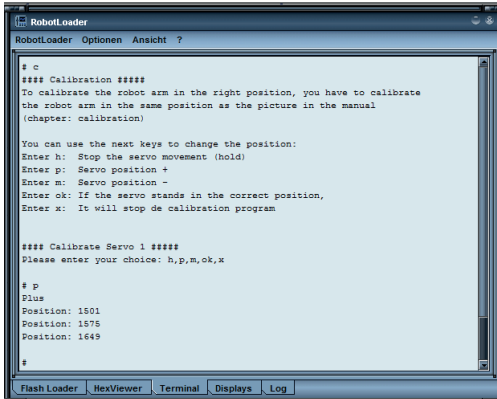


```
RobotLoader
RobotLoader Options View ?

2) Calibrate the robotarm servomotors

*****
Press ENTER to continue!
#

***** Robotarm Selftest *****
***** Main Menu ***** Advanced Menu *****
# # #
# 0 - Run ALL Selftests (1-4) # c - Calibrate Robotarm #
# 1 - LED Test # m - Current Measuring Servos #
# 2 - Voltage Sensor Test # b - Start Position #
# 3 - Burner Test # #
# 4 - Servo Test # System voltage is: 04.90V #
# # #
*****
# Please enter your choice (0-4, c, m or b)! #
*****
#
```



```
RobotLoader
RobotLoader Optionen Ansicht ?

# c
#### Calibration ####
To calibrate the robot arm in the right position, you have to calibrate
the robot arm in the same position as the picture in the manual
(chapter: calibration)

You can use the next keys to change the position:
Enter h: Stop the servo movement (hold)
Enter p: Servo position +
Enter m: Servo position -
Enter ok: If the servo stands in the correct position,
Enter x: It will stop de calibration program

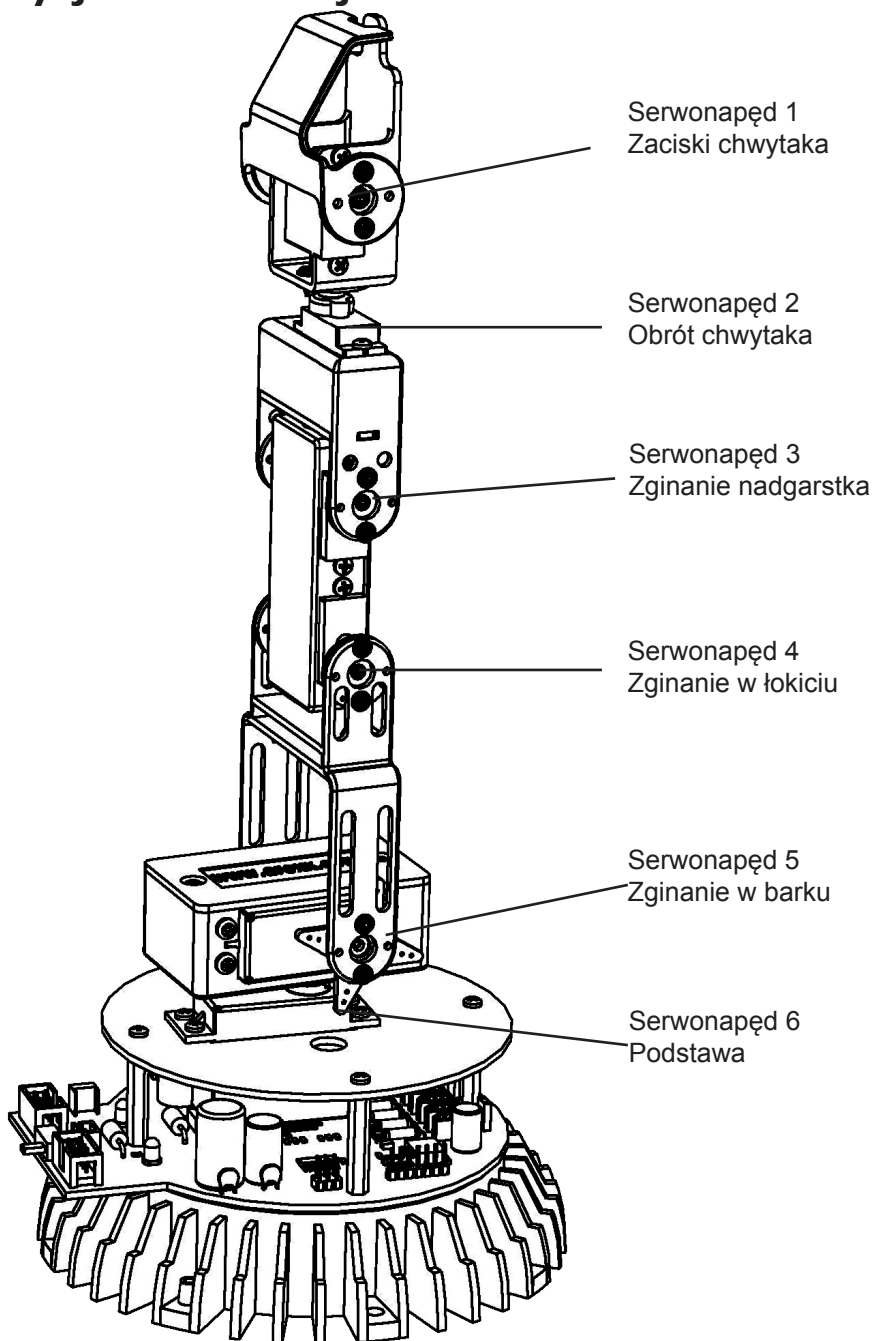
#### Calibrate Servo 1 ####
Please enter your choice: h,p,m,ok,x

# p
Plus
Position: 1501
Position: 1575
Position: 1649
#
```

Ustaw wszystkie serwonapędy w środkowej pozycji tak, aby robot wyglądał jak na stronie 51. Serwomotory od 2 do 6 są w przybliżeniu w pozycji środkowej, a chwytak (serwonapęd nr 1) jest niemal zamknięty.

Kiedy kalibracja będzie zakończona, robot może rozpocząć wykonywanie autotestu. Wynik kalibracji zapisuje się w mikrokontrolerze.

Pozycja do kalibracji



7.8. Test klawiatury

W zestawie zawarta jest klawiatura, którą można podłączyć do ramienia robota. Użyć jej można, aby w łatwy sposób zaprezentować działanie robota, a także, aby poćwiczyć sposób sterowania.

Klawiatura zawiera 6 przycisków do sterowania i 4 specjalne przyciski do późniejszych rozszerzeń.

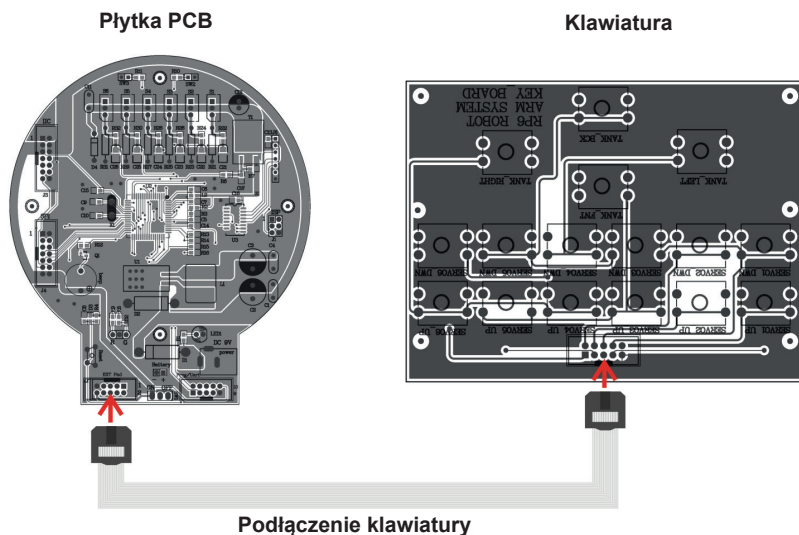
Aby przetestować ramię robota klawiaturą, należy wgrać odpowiedni plik programu hex do mikroprocesora.

Należy w tym celu kliknąć przycisk "Add" u dołu okna RobotLoadera oraz wybrać plik RobotArmExamples, "RobotArm_Key_Board.hex".

Wybierz z listy plik „RobotArm_Key_Board.hex” i przyciśnij przycisk "Upload!" na górze nad paskiem postępu.

Jeśli i ten krok zostanie wykonany, można już sterować ramieniem robota używając klawiszy klawiatury.

UŻYWAJ POPRAWNEGO OPROGRAMOWANIA (V3 PRO LUB MINI)!



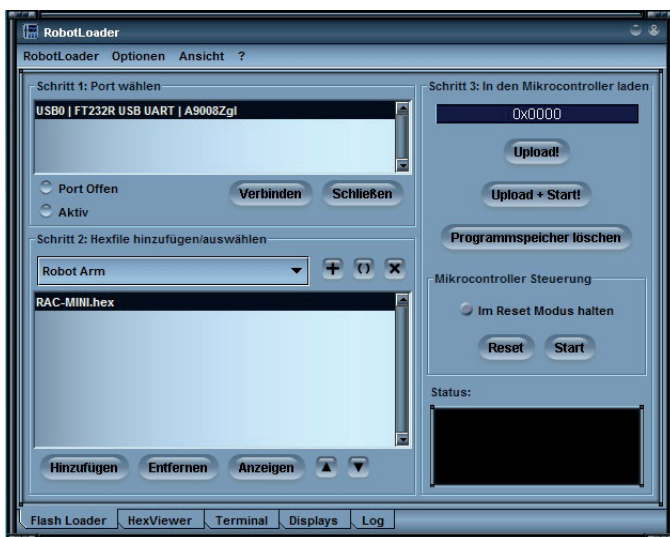
8.0. Oprogramowanie RACS

RACS (Robot Arm Control Software) to najprostszy sposób na sterowanie oraz programowanie ramienia robota. Do programowania poprzez RACS wymagane jest użycie programu RobotLoadera oraz adaptoru USB.

Przed użyciem robota do pamięci Flash procesora należy wgrać program RAC-MINI.hex.

Podłącz przewód z USB do komputera i otwórz program Loadera. Powinno pokazać się następujące okno:

Fig. 1



Jeśli na liście "Step 1: Select a port" (wybór portu) nie pojawia się żaden port, upewnij się, że przewód jest podłączony i sterowniki programera są zainstalowane. Możesz wywołać listę portów jeszcze raz odświeżając ją z menu: **RobotLoader -> Refresh port list**. Wybierz port i kliknij "Connect" (połącz).

Wybierz odpowiedni plik .hex z listy "Step 2"

– Kliknij "Add": **RACV3-MINI.HEX**

**UŻYWAJ POPRAWNEGO OPROGRAMOWANIA
(V3 PRO LUB MINI)!**

W kroku 3 kliknij przycisk "Upload" (wgraj, załaduj), aby wgrać program.

Jeśli chcesz pracować na ramieniu, odłącz RobotLoader w polu "Step 1" klikając przycisk "Close" (zamknij). Jeśli zamkniesz program, połączenie zostanie przerwane automatycznie.

Upewnij się, że nie ma połączenia między programem Loadera a ramieniem robota, bo inaczej robot nie może być sterowany z programu RACS.

7.1. Instrukcje dla RACS

Ramię robota steruje się łatwo poprzez oprogramowanie RACS.

Nawiązywane jest połączenie i silniki ramienia robota zaczynają reagować na zmianę pozycji suwaków, które zmienia się za pomocą myszy. Aktualne pozycje serwonapędów mogą być zapisane, zmieniane i wymazywane z listy w dolnej części interfejsu użytkownika. Generuje się lista zawierająca poszczególne pozycje, które mogą zostać zapisane w pliku na komputerze poprzez kliknięcie przycisku "Save". Lista kroków może zostać wgrana w każdej chwili.



Uwaga! Ważne !!!

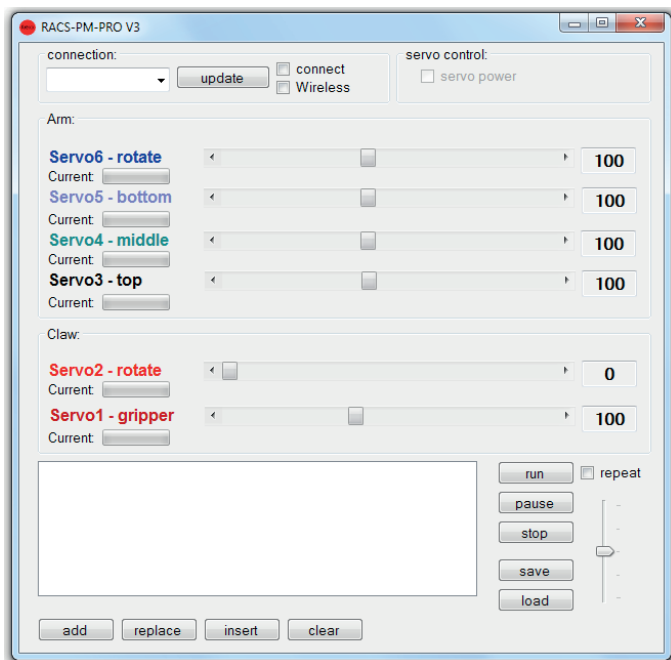
Robot monitoruje prąd każdego z serwomotoru z osobna. Jeśli wartość progowa serwonapędu wykracza poza granice, np. podczas kolizji lub przy przeciążeniu - tekst w oprogramowaniu RACS zaczyna migać. W tym przypadku, robot musi zostać wrócić do swojej poprzedniej pozycji jak najszybciej lub musi zostać odłączone zasilanie dla danego serwonapędu w programie RACS (odznaczyć pole "servopower").

Inaczej robot może zostać uszkodzony nieodwracalnie!!!

7.2. RACS - Połączenie

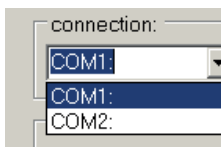
1. Po włączeniu programu do sterowania, powinno się ukazać poniższe okno:

Fig. 2



2. W rozwijanym menu wymienione są wszystkie interfejsy szeregowo:

Fig. 3



3. Podłącz wejście programatora USB
4. Kliknij przycisk "Update". Kiedy spojrzysz na menu rozwijane ponownie, zobaczysz dodatkowe interfejsy. Interfejs został zainicjalizowany poprzez podłączenie fizycznego programera USB.

Uwaga: Nazwy interfejsu dla różnych komputerów różnią się od siebie!

5. Wybierz nowy interfejs

Fig. 4



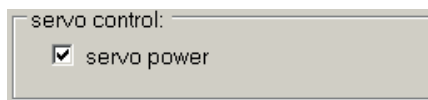
6. Zaznacz pole "Connect"

Fig. 5



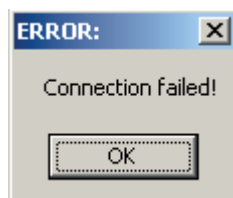
7. Zaznacz pole "servo power"

Fig. 6



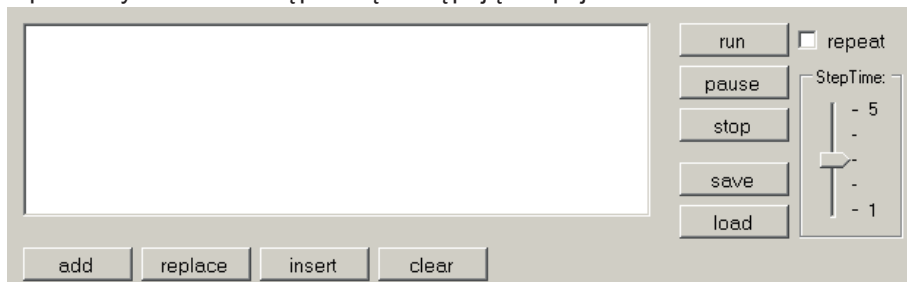
8. Poruszaj suwakami, aby zmienić pozycje serwonapędów. Jeśli podczas nawiązywania połączenia wystąpi błąd, pojawi się poniższy komunikat. Należy nawiązać połączenie ponownie (powtórz kroki 2-6 i spradz interfejs).

Fig. 7



7.3. RACS – automatyczna kontrola pozycji

W poniższym oknie dostępne są następujące opcje:



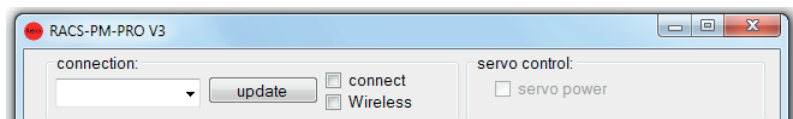
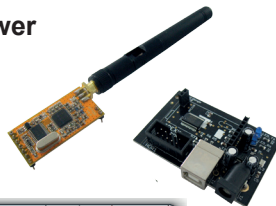
- Add:** ten przycisk dodaje aktualną pozycję suwaka do listy
- Replace:** wybrany z listy element zostaje zastąpiony przez aktualne pozycje paska przesuwania
- Insert:** aktualne pozycje suwaka zostają umiejscowione pod wybrane elementy listy
- Clear:** wybrane elementy z listy zostają wymazane
- Save:** elementy z listy zostają zapisane
- Load:** elementy z listy zostają wgrane z pliku
(Uwaga, aktualna lista elementów zostaje wymazana!)
- Run:** Elementy z listy wykonywane są w sekwencji od góry do dołu. Jeśli opcja "Repeat" została zaznaczona, ramię robota będzie wykonywał instrukcje w pętli.
- Step Time:** Step time to czas w sekundach, który definiuje jak długo robot będzie czekał zanim wykona kolejną instrukcję z listy. Jeśli lista zawiera jedynie krótkie przemieszczenia, wybrany czas może być krótki. Jeśli natomiast zawiera duże przemieszczenia, np. ruchy serwo o 180°, wybrany czas musi być dłuższy, ponieważ robot może nie zdążyć wykonać danego ruchu i w połowie drogi zacząć wykonywać kolejny krok.
- Pause:** Przerwa w wykonywaniu instrukcji.
- Stop:** Wykonywanie instrukcji jest zatrzymane.

7.3.1 RACS – bezprzewodowe

Z oprogramowaniem RACS i zestawem AREXX ARX-APC-220, można sterować ramieniem robota także bezprzewodowo. Poniżej opisane jest krok po kroku jak to działa:

- Podłącz programator RP6v2 i APC-220 do PC, dokładnie tak, jak opisano to na płycie.
- Podłącz APC do płytki PCB ramienia robota (zobacz na stronie 16).
- Załaduj Wireless Racs HEX plik do procesora ramienia robota.
- Wybierz poprawny port COM w programie RACS.
 - * **Wciśnij przycisk Start (na płycie robota)**
 - * **Enable --> Wireless - Connect i Servo Power**

Teraz można sterować ramieniem robota bezprzewodowo.



7.3.2 ANDROID – PROGRAM

Nasz zestaw ARX-BT03 umożliwia sterowanie ramienia robota poprzez Bluetooth i aplikację ANDROID.

Podążaj za następującymi instrukcjami:

- Podłącz moduł Bluetooth do płytki PCB ramienia robota (str. 16).
- Załaduj dane z ANDROID HEX do procesora ramienia robota.
- Załaduj dane z ANDROID APK do swojego smartfona lub na tablet. Pliki te możesz znaleźć na CD oraz w Sklepie Google Play.



8.0. PROGRAMOWANIE RAMIENIA ROBOTA

Stopniowo dochodzimy do programowania robota.

Ustawienie edytora tekstu źródłowego

Na początku należy ustawić środowisko do programowania. Tak zwany kod źródłowy dla naszego programu w języku C musi w jakiś sposób znaleźć się w komputerze!

W tym celu z pewnością nie użyjemy programów do edycji tekstu takich jak OpenOffice czy Word! Żart ten wynika z tego, że nie dla wszystkich może to być oczywiste. Tekst źródłowy to czysty tekst bez formatowania. Kompilator nie dba o kolor ani rozmiar czcionki...

Dla człowieka formatowanie ma znaczenie, ponieważ łatwiej mu odnaleźć w tekście kluczowe słowa. Niektóre z tych funkcji i inne są zawarte w edytorze Programmers Notepad 2 (w skrócie "PN2"), który będziemy używać (UWAGA: Pod systemem Linux należy użyć innego edytora, który oferuje te same funkcje jak PN2. Zazwyczaj takie narzędzia są już w systemie (np. kate, gedit, exmacs)). Dodatkowo wyróżniane są słowa kluczowe ("podświetlanie składni"), a także oferuje elementarne zarządzanie projektem. Pozwala to na organizowanie kilku plików tekstu źródłowego w jednym projekcie oraz wyświetlanie ich w liście, której elementy podpite są do jednej jednostki projektowej. Ponadto, można łatwo naprawić programy takie jak AVR-GCC w PN2 i otrzymać programy skompilowane. Zazwyczaj AVR-GCC to program zawierający szereg linii komend bez żadnego interfejsu graficznego.

Więcej ostatnich wersji programu Programmers Notepad znajdziesz na stronie domowej projektu: <http://www.pnotepad.org/>

Najnowsze wersje WINAVR już nie wymagają ustawiania elementów menu!

ZAUWAŻ:

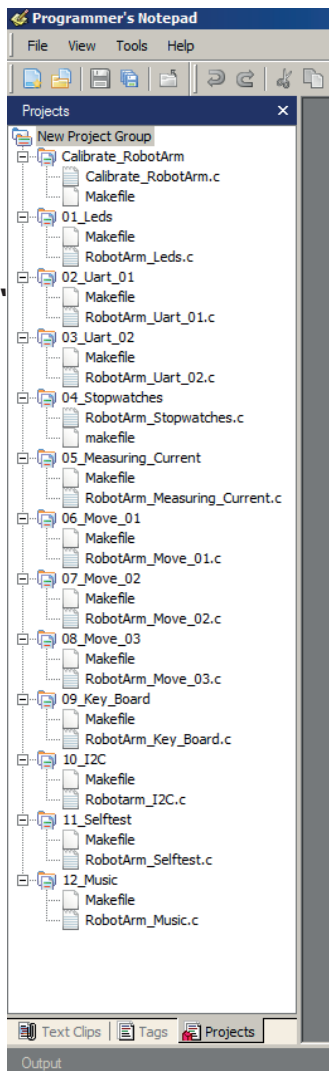
W tej sekcji nie wyjaśniamy więcej jak należy ustawić elementy menu w PN2, ponieważ nowa wersja WINAVR robi to za ciebie!

Zobacz na stronie 56 jak otworzyć i skompilować przykładowy projekt!

Jeśli otworzyłeś przykładowy projekt, powinien on wyglądać mniej więcej jak na tym zrzucie:

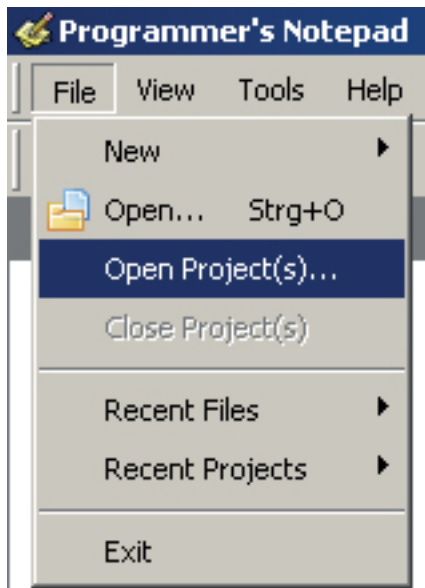
Plik "Robot ArmExamples.ppg"

To są grupy projektów dla PN2, w których są wszystkie przykładowe programy oraz Biblioteka Robot Arm są wrzucone w jedną listę projektową ("Projects").



Po lewej stronie pokazują się przykładowe projekty, a po prawej tekst źródłowy programów (z podświetlaniem składni) i na dole narzędzia wejścia (w tym przypadku wejściem jest kompilator). W programie PN2 można znaleźć wiele użytecznych właściwości i funkcji.

Otwieranie i kompilowanie przykładowego projektu



Teraz należy sprawdzić czy wszystko działa poprawnie i otworzyć przykładowy projekt:

Wybierz w menu "File" opcję "Open Project(s)".

Pojawi się zwykła sekcja dialogowa. Wyszukaj "Robot Arm_Examples [MINI]" w folderze, w którym zapisałeś przykładowe programy.

Otwórz plik "Robot ArmExamples.ppg". Jest to grupa projektowa dla PN2, która ładuje wszystkie przykładowe programy, tak jak Biblioteka Robot Arm, na listę programu ("Projects").

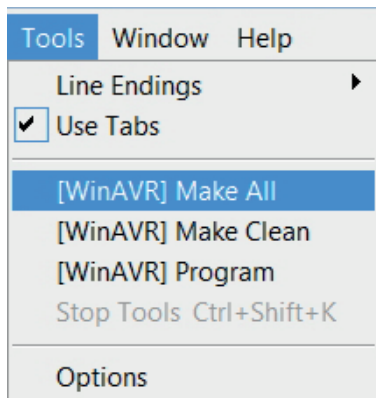
Teraz wszystkie przykładowe projekty są na wyciągnięcie ręki, jeśli zechcesz odnieść się do któregoś z nich, np. oglądając na początku przykładowe funkcje w Bibliotece Robot Arm itd.

Otwórz pierwszy z tych programów na górze listy oraz wybierz plik "01_Leds". Kliknij dwa razy na "01_Leds.c"! Edytor tekstu źródłowego wyświetla się w oknie wewnątrz programu.

Obszar wyjścia powinien pojawić się na dole okna programu PN2. Jeśli tak się nie stanie, należy włączyć ten obszar klikając w menu "View" --> "Enable output" LUB jeśli obszar jest zbyt mały, zwiększ rozmiar okna rozciągając brzegi okienka myszką (po najechaniu na brzeg okna kursor zmienia się w podwójną strzałkę).

Możesz przyjrzeć się programowi, który właśnie został otwarty, ale nie musisz teraz dokładnie rozumieć o co tam chodzi. Jednakże pierwsza wiadomość brzmi: zielony tekst oznacza komentarze, które są tylko informacją dla programisty i nie są naprawdę częścią programu (tzn. kompilator ich nie widzi).

Funkcje wyjaśnimy szczegółowo nieco dalej (istnieje również wersja programu BEZ komentarzy, abyś mógł zauważyć jak krótki tak naprawdę ten program jest. Komentarze dodają objętości kodowi, jednak są niezbędne do zrozumienia kodu. Niezakomentowana wersja jest jednak przydatna do skopiowania jako baza dla twoich własnych programów!).



Na początku należy jedynie przetestować odpowiednie działanie kompilatora.

W menu Tools na górze powinny się pojawić świeżo zainstalowane elementy menu (zobacz zrzut).

Kliknij teraz "MAKE ALL"!

PN2 tworzy teraz wyżej wymieniony plik wsadowy "make_all.bat". Plik ten z kolei będzie tworzył program "make". Więcej na temat programu "make" wyjaśnione zostanie później.

Przykładowy program nie skompiluje się. Wygenerowany plik hex zawiera program w wersji przetłumaczonej tak, aby mikrokontroler rozumiał i może być załadowany i później wykonany. Kompilacja to proces, w którym tworzy się wiele tymczasowych plików (np. takie o rozszerzeniach: .o, .lss, .map, .sym, .elf, .dep). Nie są dla ciebie istotne. Nowa opcja "make clean" wszystkie je usunie. Interesuje nas jedynie plik hex, którego funkcja "make clean" nie wymaże.

Po aktywowaniu opcji z menu MAKE ALL, powinno wyświetlić się następujące wyjście (poniżej przedstawiona skrócona wersja! Niektóre linie mogą oczywiście wyglądać nieco inaczej):

```
> "make.exe" all
```

```
----- begin -----
```

```
avr-gcc (WinAVR 20100110) 4.3.3
```

```
Copyright (C) 2008 Free Software Foundation, Inc.
```

```
This is free software; see the source for copying conditions. There is NO  
warranty; not even for MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.
```

```
Size before:
```

```
AVR Memory Usage
```

```
-----
```

```
Device: atmega64
```

```
Program: 3074 bytes (4.7% Full)
```

```
(.text + .data + .bootloader)
```

```
Data: 68 bytes (1.7% Full)
```

```
(.data + .bss + .noinit)
```

```
EEPROM: 14 bytes (0.7% Full)
```

```
(.eeprom)
```

```
Compiling C: Robot Arm_Leds.c
```

```
avr-gcc -c -mmcu=atmega64 -I.
```

```
-gdwarf-2 -DF_CPU=16000000UL -Os -funsigned-char -funsigned-bitfields -fpack-  
struct -fshort-enums -Wall
```

```
-Wstrict-prototypes -Wa,-adhlns=./Robot Arm_Leds.lst -std=gnu99 -MMD -MP -MF  
.dep/Robot Arm_Leds.o.d Robot Arm_Leds.c -o Caterpillar_Leds.o
```

```
Linking: Robot Arm_Leds.elf
```

```
avr-gcc -mmcu=atmega16 -I. -gdwarf-2 -DF_CPU=16000000UL -Os -funsigned-char -funsigned-  
bitfields
```

```
Creating load file for Flash: Robot Arm_Leds.hex
```

```
Creating load file for EEPROM: Robot Arm_Leds.eep
```

```
avr-objcopy -j .eeprom --set-section-flags=.eeprom="alloc,load" \
```

```
--change-section-lma .eeprom=0 --no-change-warnings -O ihex Robot Arm_Leds.elf
```

```
Robot Arm_Leds.eep || exit 0
```

```
Size after:
```

```
AVR Memory Usage
```

```
-----
```

```
Device: atmega64
```

```
Program: 3074 bytes (4.7% Full)
```

```
(.text + .data + .bootloader)
```

```
Data: 68 bytes (1.7% Full)
```

```
(.data + .bss + .noinit)
```

```
EEPROM: 14 bytes (0.7% Full)
```

```
(.eeprom)
```

```
----- end ----->
```

```
Process Exit Code: 0
```

```
> Time Taken: 00:04
```

Linia "Process Exit Code: 0" na końcu ma największe znaczenie. Oznacza ona, że nie wystąpił żaden błąd kompilacji. Jeśli pojawia się coś innego, kod źródłowy zawiera błąd, na temat którego kompilator zazwyczaj zwraca wiadomość, aby przybliżyć powód jego pochodzenia. Błąd ten musi zostać znaleziony i poprawiony.

Zauważ jednak, że komunikat "Process Exit Code: 0" nie zawsze jest gwarancją na całkowicie bezbłędny program! Kompilator nie znajdzie nieprawidłowego procesu myślowego, który zaimplementowałeś, np. nie może zapobiec programowi, który sprawi, że robot uderzy w ścianę. ;-)

UWAGA: Ostrzeżenia i inne błędy także możesz znaleźć w logu na dole strony. Nie przejmuj się jednak ostrzeżeniami! PN2 zaznacza błędy i ostrzeżenia innym kolorem, aby łatwiej je było znaleźć. Kompilator pokaże nawet która linijka mu się nie podoba. Jeśli klikniesz na wiadomość błędu, PN2 pokaże ci wadliwą linię kodu.

Wskaz na końcu, czyli "AVR Memory Usage" również jest przydatny.

Size after:

AVR Memory Usage

Device: atmega64

Program: 3074 bytes (4.7% Full)

(.text + .data + .bootloader)

Data: 68 bytes (1.7% Full)

(.data + .bss + .noinit)

Dla procesora Atmega64 oznacza to, że nasz program ma rozmiar 3074 bajtów oraz 68 bajtów pamięci RAM jest zarezerwowane na zmienne statyczne (należy dodać też dynamiczne zakresy stosu, ale to już bardziej zaawansowane... po prostu zawsze staraj się mieć conajmniej kilkaset bajtów wolnej pamięci). Procesor dysponuje 64kb (65536 bajtów) pamięci Flash ROM oraz 2kb (2028 bajtów) pamięci RAM. Wśród 64kb, 2k jest zajęte przez bootloader - więc właściwie można używać tylko 62kb. Zawsze się upewnij, że program mieści się w limicie pamięci, jaki narzuca mikrokontroler (RobotLoader nie prześle za dużego programu!)

Oznacza to, że przykładowy program powyżej ma jeszcze zapas 60414 bajtów wolnej pamięci. Stosunkowo krótki program o nazwie Example_01_Leds.c ma taki duży rozmiar, bo zawiera bibliotekę Robot ArmBaseLibrary! Więc nie przejmuj się, jest wystarczająco dużo miejsca na twoje programy, a niewielkie programy zazwyczaj nie wymagają dużo miejsca. Sama biblioteka wymaga zaledwie kilka kb pamięci flash, ale znacznie ułatwia pracę i dzięki niej twoje programy będą znacznie mniejsze w porównaniu do biblioteki Robot ArmBaseLibrary.

Świeżo skompilowany program może być teraz wgrany do robota poprzez RobotLoader. Aby to zrobić, należy dodać wygenerowany plik hex do listy w RobotLoaderze przyciskiem "Add", wybierz go oraz kliknij na "Upload" dokładnie tak jak przy programie autotestu. Później możesz przełączyć na terminal i spojrzeć co dzieje się na wyjściu programu. Należy oczywiście uruchomić program. Najlepiej poprzez wciśnięcie kombinacji klawiszy [STRG]+[S] w zakładce terminala lub używając menu (lub wpisz "s" - po resecie musisz chwilę poczekać na wiadomość "[READY]" z terminala!). Kombinacja klawiszy [STRG]+ [Y] jest również bardzo wygodna. Jeśli aktualnie wybrany program jest załadowany do ramienia robota, kombinacja sprawia, że program od razu zaczyna bieg. Dzięki temu można uniknąć klikania na zakładkę "Flash Loader" lub używania menu.

Przykładowy program jest bardzo prosty: zapala on jedynie diody LED i wyświetla tekst na wyjściu.

W ramach podsumowania

Mamy nadzieję, że nasze roboty wprowadziły cię do świata robotów. Tak jak nasi japońscy przyjaciele wierzymy, że roboty staną się następną rewolucją technologiczną tak jak to było z komputerami i telefonią komórkową! I będzie to rewolucja, która przyniesie nowe ekonomiczne możliwości.

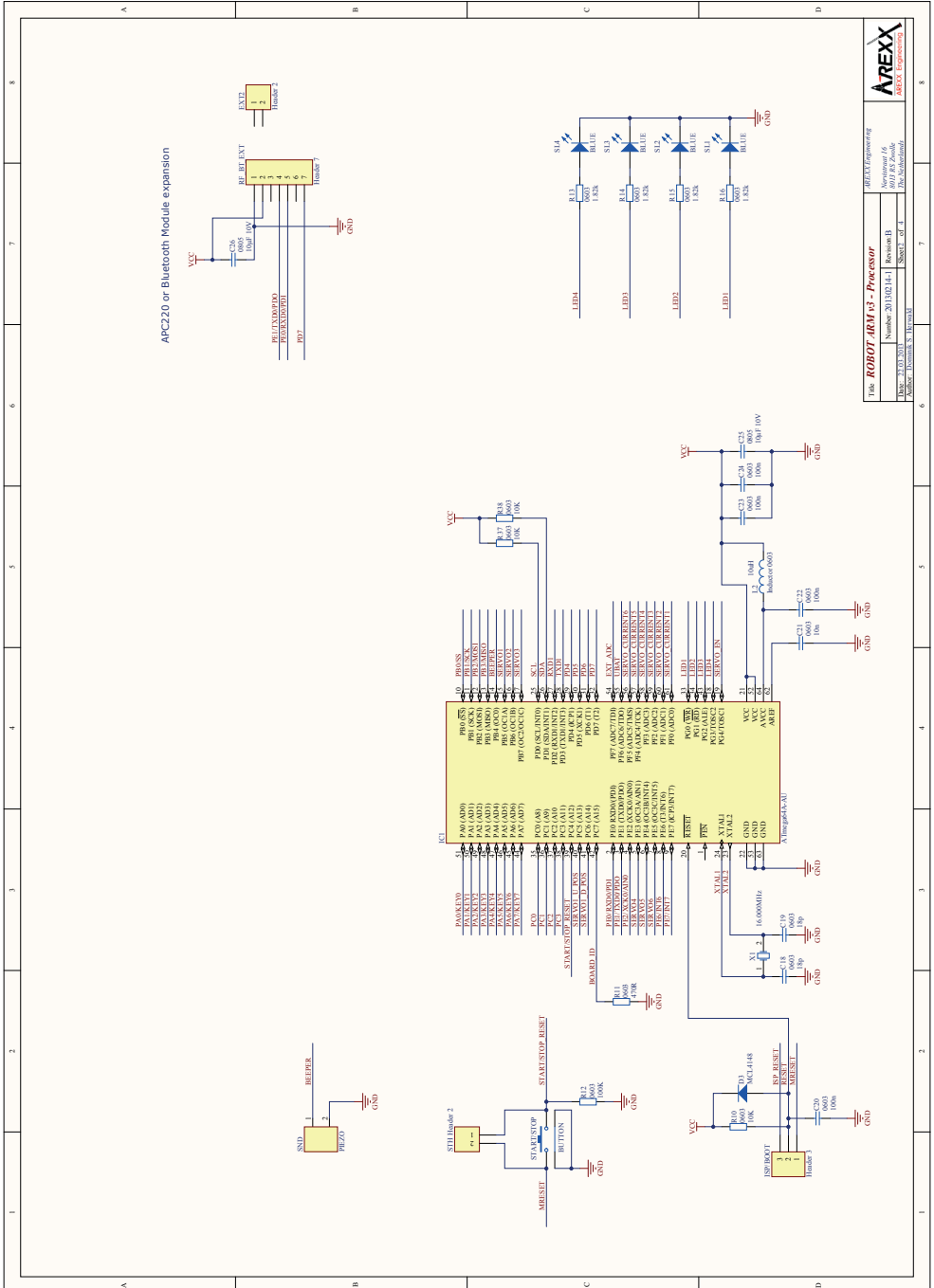
Niestety w tym polu Japonia, inne szybko rozwijające się kraje Dalekiego Wschodu oraz USA znacznie wyprzedzają Europę. W przeciwieństwie do Europy, na Dalekim Wschodzie uczeń rozpoczyna swoją techniczną edukację już w szkole podstawowej, a nauki techniczne odgrywają kluczową rolę w tamtejszych systemach edukacji.

Mamy cel rozwinąć nasze roboty ASURO, YETI, Caterpillar i Ramię Robota, ponieważ:



DODATEK

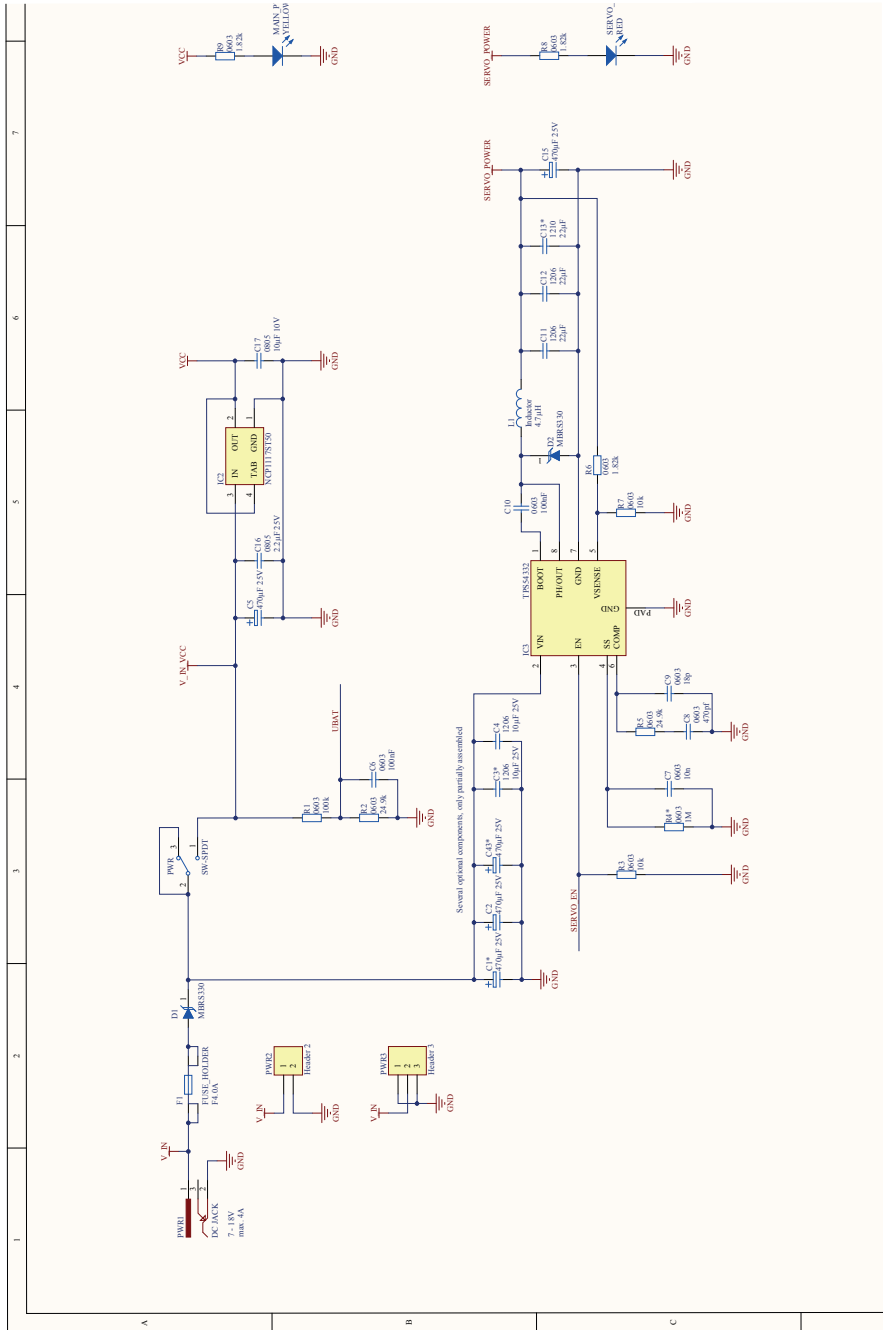
A. SCHEMAT ELEKTRYCZNY ROBOTA RA2-HOBBY V3



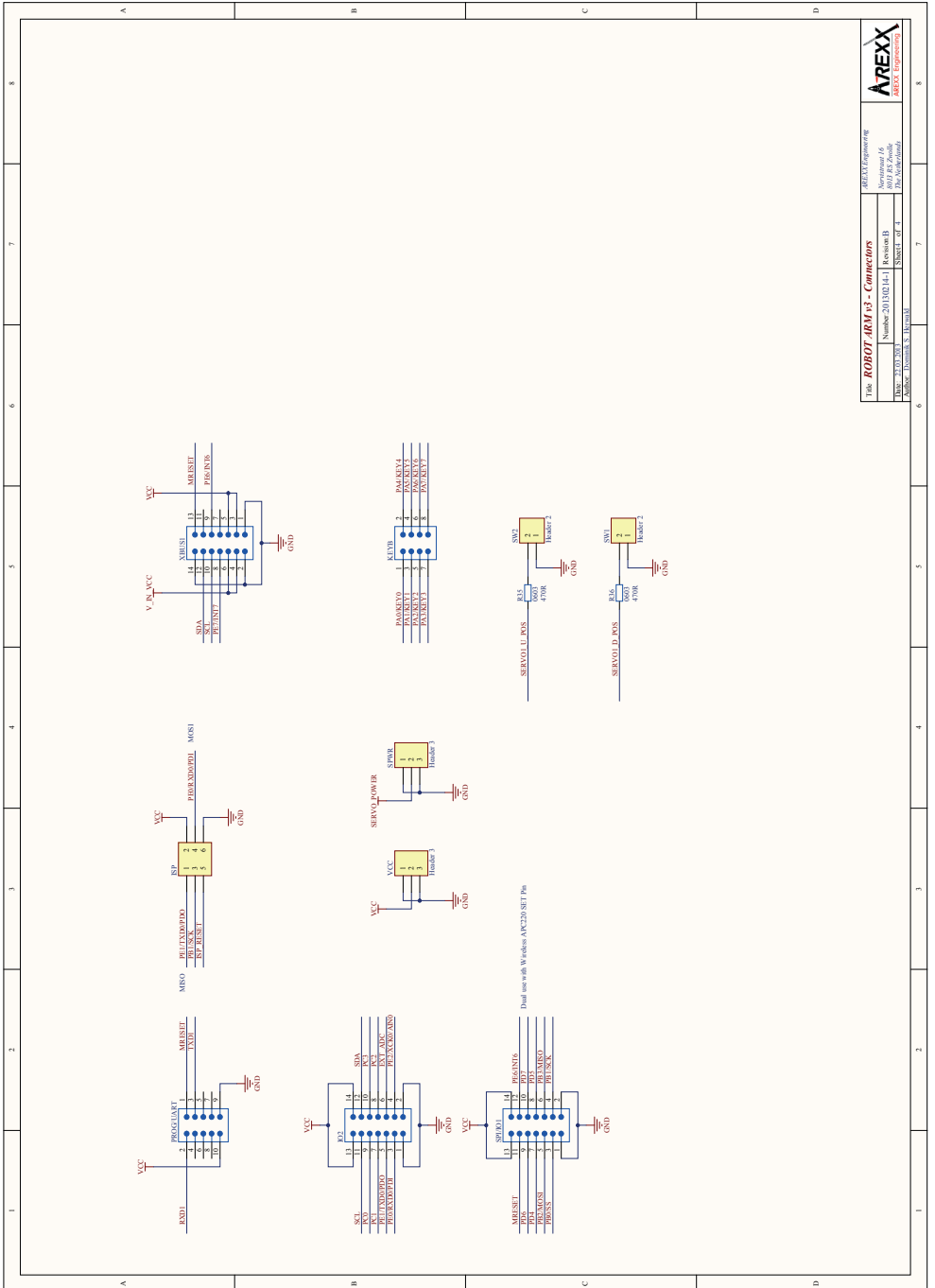
Tytuł: **ROBOTA RA2 - Procesor**
 Nazwa pliku: Robotarium 16
 Numer katalogowy: 160718-001
 Data wydania: 16.05.2017
 Nazwa: Robotarium K. Hryniewski

AREXX
 Robotarium
 00 22 62 22 22

B. SCHEMAT UKŁADU ZASILANIA ROBOTA RA2-HOBBY V3



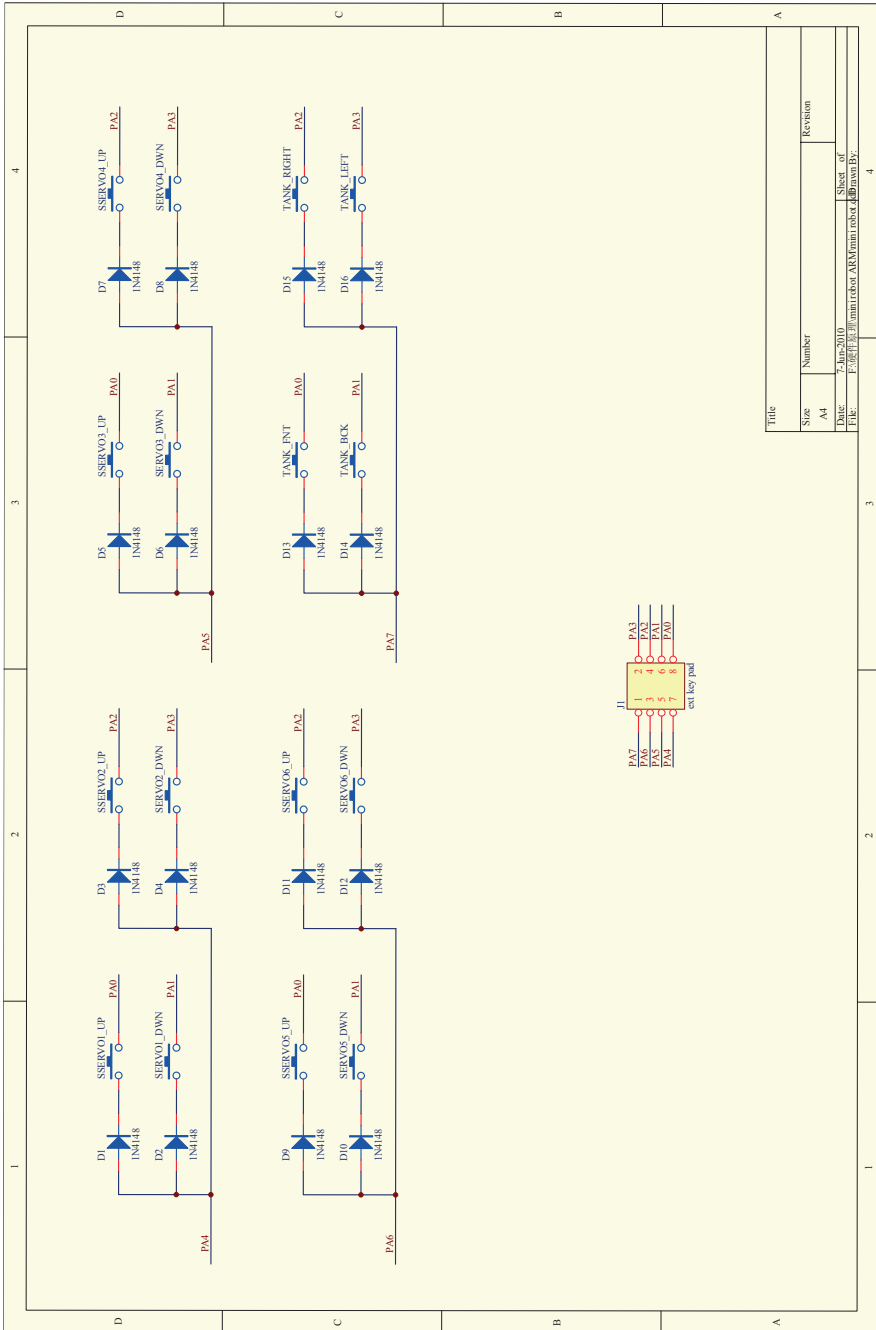
C. SCHEMAT ZŁĄCZY ROBOTA RA2-HOBBY V3



AREXX
AREXX Engineering
Number 2013102 (1-1) | Revision 1 |
Sheet of 1

Title: **ROBOT-ARCV3 - Connectors**
Author: **AREXX S. Kowalski**

D. SCHEMAT KLAWIATURY ROBOTA RA2-HOBBY V3



E. PŁYTKA PCB RAMIENIA ROBOTA RA2-HOBBY V3

