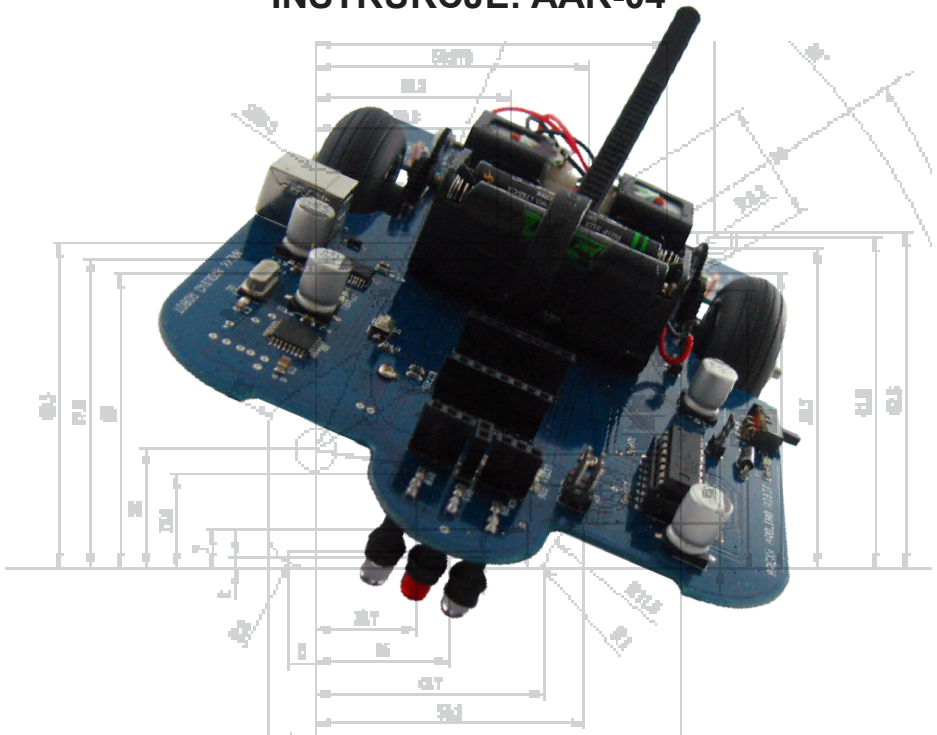




ROBOT ARDUINO AAR

INSTRUKCJE: AAR-04



Contents

1. OPIS PRODUKTU AAR	
1.1 Rodzina robotów ARDUINO	5
1.2 Specyfikacja techniczna	5
2. OGÓLNY OPIS ARDUINO	7
3. ROBOT ARDUINO	12
3.1. Opis LED robota AAR	12
3.2. Układ płytki PCB robota AAR	13
4. ROBOT ARDUINO (AAR)	16
4.1. Pobranie i instalacja oprogramowania	16
4.2. Język Arduino	16
4.3. Instalacja sterownika USB	16
4.4 Montaż koszyka bateryjnego	17
4.5 Programowanie z programami Arduino	18
4.6 Wybór programu Arduino	18
4.7 Wybór poprawnego portu COM	19
4.8 Wgrywanie programu do Robota Arduino	20
4.9 Rozwiązywanie problemów	21
4.10 Autotest AAR	22
5. Moduły rozszerzające AAR	23
6. Poboczne informacje na temat obwodów mostka H	27
7. Odometria	30
8. Programowanie w Boot-loader	33

DODATEK

A. Lista elementów	35
B. Główna płytki PCB - rzut z góry	37
C. Główna płytki PCB - rzut z dołu	38
D. Schematy AAR	39
E. 3D PCB AAR	40

UWAGA!

AAR to znak towarowy zastrzeżony.

Wszystkie prawa zastrzeżone.

Każda reprodukcja lub adaptacja całości lub części podręcznika bez pisemnej zgody jest zabroniona. Specyfikacja, forma oraz wszystkie treści tego produktu mogą być zmienione bez wcześniejszego uprzedzenia.

Producent:

AREXX Engineering
Zwolle, The Netherlands



© 2014 AREXX Engineering

Wsparcie techniczne dostępne na:

www.arexx.com

www.roboternetz.de

<p>Impressum</p> <p>©2013 AREXX Engineering Nervistraat 16 8013 RS Zwolle The Netherlands</p> <p>Tel.: +31 (0) 38 454 2028 Fax.: +31 (0) 38 452 4482</p> <p>E-Mail: Info@arexx.nl</p>	<p>Wszystkie prawa zastrzeżone. Każda reprodukcja lub adaptacja całości lub części podręcznika wymaga pisemnej zgody.</p> <p>Specyfikacja produktu oraz dostarczane treści mogą podlegać zmianom. Treść podręcznika może ulec zmianie bez wcześniejszego uprzedzenia.</p> <p>Darmowe aktualizacje podręcznika można znaleźć na:</p> <p>http://www.arexx.com/</p>
<p>"AAR and AREXX" 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!</p>	
<p>Informacje dotyczące ograniczonej gwarancji oraz odpowiedzialności</p> <p>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.</p> <p>Gwarancja nie dotyczy bezpośrednio uszkodzonych elementów podczas użytkowania. Takie założenie wyklucza wszelkie roszczenia, które podlegają odpowiedzialności prawnej.</p> <p>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!</p> <p>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".</p> <p>Proszę zwrócić uwagę na umowy licencyjne oprogramowania na CD-ROM!</p> <p>WAŻNE!</p> <p>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.</p> <p>Ważne zalecenia dotyczące bezpieczeństwa</p> <p>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 płytce PCB. Nigdy nie przeciążaj silników.</p>	

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 serwo mechanizmów.
- 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 płycie 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 AAR

1.1. Rodzina robotów ARDUINO

Arduino to platforma open source stworzona do budowania prototypów elektronicznych. Umożliwia to mikrokontroler, który zawiera wszystkie dodatkowe interfejsy oraz potrzebne do tego celu oprogramowanie.

Koncept Arduino został zaprojektowany aby uczyć nowoczesnej elektroniki dla robotyki, sterowanie oprogramowaniem oraz czujników w najprostszy możliwy sposób.

Jako następcę robota ASURO, który można programować w języku C, zaprojektowaliśmy robota Arduino. Nowy robot przypomina swojego poprzednika, jednak różni je sposób programowania, który w przypadku układu Arduino jest znacznie prostsze.

1.2. Specyfikacja techniczna:

Silniki	2 silniki DC (3 Volt)
Procesor	ATmega328P-AU
Język programowania	ARDUINO
Napięcie zasilające	4 x baterie typu AAA 4,8 - 5,5 Volt
Prąd zasilający	Min. 10 mA Max. 600 mA
Sposób komunikacji	kabel USB
Rozszerzenia	kompatybilne z rozszerzeniami ASURO
Wysokość	40 mm
Szerokość	120 mm
Długość	180 mm



Ostrzeżenia

- * 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 Robot nie będzie używany dłużej niż tydzień.

1.3. Środki ostrożności

1. Uwaga!

Należy przeczytać ten podręcznik przed podłączeniem robota do zasilania! Niepoprawne podłączenie może trwale uszkodzić cały sprzęt lub jego część.

2. Uwaga!

Należy obejrzyć dokładnie wszystkie schematy połączeń! Bądź ostrożny w podłączeniach schematów. Błąd w połączeniach może uszkodzić moduły. Pamiętaj o poprawnej polaryzacji baterii (źródła zasilania). Odwrotnie podłączone zasilanie także może uszkodzić sprzęt.

3. Uwaga!

Nie należy używać napięcia zasilającego wyższego niż podane na poprzedniej stronie wartości!

Używaj tylko ustabilizowanego i przefiltrowanego napięcia zasilającego, aby uniknąć przepięć.

4. Uwaga!

Układ nie posiada żadnej ochrony wodoodpornej.

Należy używać układu tylko w suchym środowisku.

5. Uwaga!

Należy unikać zwarcia na jakiegokolwiek metalowej powierzchni oraz używania nadmiernej siły oraz wagi na płytce lub wtykach.

6. Uwaga!

Należy unikać wyładowań elektrostatycznych (ESD - z ang. Electrostatic Discharge).

2. Ogólny opis ARDUINO

2.1. Czym i kim jest ARDUINO?

Arduino to płytką open source z 8-bitowym mikroprocesorem, która umożliwia łatwe programowanie mikrokontrolera oraz pracę nad projektami z obiektami interaktywnymi dla artystów, projektantów, hobbystów i innych.

Platforma Arduino oparta jest na mikrokontrolerach firmy Atmel - ATmega168 lub ATmega328. Układ zapewnia użytkownikom kanały wejść analogowych oraz cyfrowych, które pozwalają na otrzymanie i wysłanie sygnału z/do środowiska.

Na rynku dostępnych jest kilka płytek Arduino takich jak: Arduino Uno, Arduino LilyPad i Arduino Mega 2560. Każda z nich została zaprojektowana do innych celów. Użytkownik sam może wybrać płytkę dopasowaną do swojego projektu.

Sygnały wejściowe mogą być doprowadzane poprzez przełączniki, czujniki światła, czujniki prędkości i przyspieszenia, czujniki odległości oraz temperatury. Dodatkowo, jest możliwość otrzymania instrukcji z dowolnego źródła internetowego. Sygnały wyjściowe mogą być użyte do sterowania silnikami, pompami czy też informacjami na ekranach.

Układ został wyposażony w kompilator dla języków programowania według standardów oraz w boot-loader. Język programowania został oparty na tzw. wiring language, który przypomina język C++.

Projekt Arduino rozpoczęty został w roku 2005 w mieście Ivrea, we Włoszech. Głównym celem projektu było wsparcie studentów w ich projektach, gdzie prototypy powinny być dużo tańsze i równie wydajne jak większość standardowych metod.

Grupa pod opieką Massimo Banzi i Davida Cuartielles zdecydowała się nazwać projekt po historycznym charakterze 'Arduina z Ivrei'. "Arduino" to włoska wersja tego imienia.

2.2 Mikrokontrolery

2.2.1. Aplikacje

Mikrokontroler (oznaczany często jako μC , uC lub MCU) to mały komputer w małym układzie scalonym, który zawiera rdzeń procesora, pamięć oraz programowalne wejścia/wyjście (I/O). Dodatkowo w układzie znajduje się pamięć programu oraz niewielka ilość pamięci danych (RAM).

Mikrokontrolery stosuje się w urządzeniach sterowanych automatycznie, np. samochodowe systemy sterowania silnikiem, piloty, urządzenia biurowe, implanty medyczne, zabawki i inne. Dzięki mikrokontrolerom dochodzi do minimalizacji rozmiaru oraz kosztów projektów w porównaniu do urządzeń, które używają osobnego mikroprocesora, pamięci oraz wejść/wyjść. Dzięki mikrokontrolerom dąży się do tego, że tworzenie urządzeń jest bardziej ekonomiczne, ponieważ może on sterować cyfrowo dużą ilością urządzeń oraz procesów.

W rozwiniętym kraju jest bardzo prawdopodobne, że typowy dom posiada kilku mikroprocesorów ogólnego użycia oraz wiele mikrokontrolerów. Typowe auto średniej klasy ma trzydzieści lub więcej mikrokontrolerów. Można je również znaleźć w wielu urządzeniach elektronicznych, np. zmywarki do naczyń, mikrofalówki czy telefony.

2.3. Pobór mocy oraz prędkość

Niektóre mikrokontrolery mogą pracować w niskich częstotliwościach zegara (np. 4 kHz) w celu poboru małej mocy (rzędu miliwatów lub mikrowatów). Będą mogły utrzymać funkcjonalność w momencie oczekiwania na wydarzenie, np. naciśnięcie przycisku; pobór mocy w stanie uśpienia (zegar procesora i inne urządzenia są wyłączone) może być nawet w nanowatach, co umożliwi oszczędność baterii.

Inne mikrokontrolery mogą zachowywać się jak cyfrowy przetwornik sygnałów, jednak w tym przypadku pobór mocy znacznie wzrasta, ponieważ wymagają one większej częstotliwości zegara.

Swoje zastosowanie w układzie Arduino ma 8-bitowy mikrokontroler Atmel ATmega328P o częstotliwości 16MHz, z wbudowaną programowalną 32-kilobajtową pamięcią flash. Napięcie zasilające dla tego mikrokontrolera zostało zaprojektowane dość ogólnie, tzn. w zakresie DC 7-12V, który zapewnia stabilne i bezpieczne warunki dla chipu oraz izolowane linie mocy aż do 2A dla zespołu obwodów elektrycznych silników.

2.4 Programy mikrokontrolera

Programy mikrokontrolera muszą mieścić się w dostępnej pamięci programowej na chipie, ponieważ dodatkowa pamięć zewnętrzna byłaby kosztowna. Kompilery i assembly (programy tłumaczące kod programu na "język komputera") są używane do konwertowania języka wysokiego poziomu na kod maszynowy do zasobu pamięci mikrokontrolera. W zależności od urządzenia, pamięć programu może być stała, tylko do odczytu, która może zostać zapełniona jedynie raz, fabrycznie lub pamięć programu może być programowalna przez użytkownika lub też tylko do odczytu, ale dająca się wymazać.

Mikrokontrolery na początku były programowane tylko w języku assemblera, ale wiele rodzajów programowalnych językami wysokiego poziomu stały się teraz głównym celem uC. Te języki zostały zaprojektowane specjalnie do wykorzystania ogólnego, np. język C. Sprzedawcy mikrokontrolerów zazwyczaj tworzą narzędzia ogólniedostępne, aby umożliwić łatwiejsze zastosowanie ich sprzętu.

Układ Arduino zapewnia nam około 32 kilobajtów pamięci flash na programy, które mogą być programowane w języku C.

2.5. Architektura interfejsu

Mikrokontrolery zazwyczaj zawierają od kilku do kilkudziesięciu uniwersalnych pinów wejścia/wyjścia (GPIO - general purpose input/output). Piny GPIO można konfigurować programując je, aby były wejściami lub wyjściami. Jeśli piny GPIO zostają określone jako piny wejściowe, mogą być używane do odczytywania sygnałów z zewnątrz, np. z czujników. Skonfigurowane jako wyjścia mogą sterować zewnętrznymi urządzeniami, np. diodami elektroluminescencyjnymi lub silnikami.

Wiele systemów wbudowanych odczytuje dane z czujników, które wytwarzają sygnały analogowe. W tym miejscu swoje zastosowanie znajdują przetworniki analogowo-cyfrowe (ADC). Odkąd stworzono procesory możliwe jest interpretowanie i przetwarzanie danych w postaci cyfrowej. Przedtem nic nie dało się zrobić z wysłanymi przez urządzenia danymi analogowymi. Zatem konwertery ADC są używane do konwersji przychodzących danych do postaci, które procesor jest w stanie rozpoznać. Mniej popularną opcją w niektórych uC są konwertery cyfrowo-analogowe (DAC), które pozwalają procesorowi wysłać analogowe sygnały lub poziom napięcia.

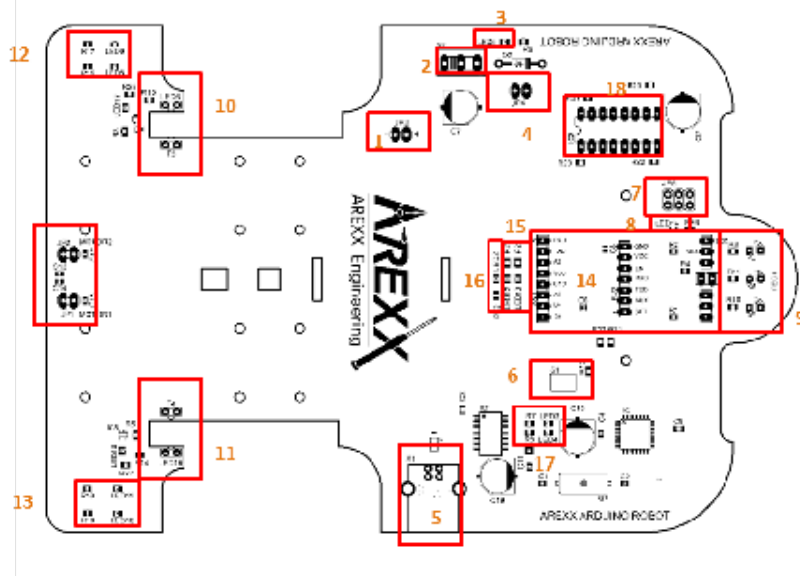
Dodatkowo, w wielu mikroprocesorach znajdują się różne rodzaje timerów. Jednym z najbardziej znanych typów timerów są tzw. PIT (z ang. Programmable Interval Timer). Ten rodzaj timerów liczy od pewnej wartości do 0 i kiedy je osiągnie wysyła do procesora informację o zakończeniu liczenia. Jest to przydatne w urządzeniach takich jak termostaty, które cyklicznie sprawdzają temperaturę otoczenia, aby sprawdzić, czy należy włączyć klimatyzację czy grzejnik.

Universal Asynchronous Receiver/Transmitter (UART) umożliwia wysyłanie i odbieranie danych za pomocą tylko jednej linii, co nie obciąża mocno procesora. Sprzęt na chipie również przeznaczony został do komunikacji z innymi urządzeniami (chipami) w formie cyfrowej, np. magistrała I2C oraz SPI (Serial Peripheral Interface).

Układ Arduino zapewnia nam aż 14 cyfrowych linii wejść/wyjść (I/O) oraz 7 linii wejść/wyjść analogowych.

3. ROBOT ARDUINO

Rys.: Płytki PCB AAR



3.1 OPIS LED ROBOTA AAR

LED 1.	Czerwona	Czujnik linii (na spodzie)
LED 2.	Niebieska	Dowolnie programowana
LED 3.	Czerwona	USB DATA
LED 4.	Zielona	USB DATA
LED 5.	Niebieska	Dioda zasilania
LED 6.	IR	Dioda podczerwona, sensor lewego koła
LED 7.	NA	NA
LED 8.	Czerwona	Wskaźnik silnika: w lewo, w przód
LED 9.	Czerwona	Wskaźnik silnika: w lewo, w tył
LED 10.	Czerwona	Wskaźnik silnika: w prawo, w tył
LED 11.	Czerwona	Wskaźnik silnika: w prawo, w przód
LED 12.	Zielona	linia TX (wysyłanie)
LED 13.	Czerwona	linia RX (odbieranie)
LED 14.	Niebieska	Dowolnie programowana
LED 15.	NA	NA
LED 16.	Czerwona	Czujnik impulsu prawego koła
LED 17.	Czerwona	Czujnik impulsu lewego koła
LED 18.	IR	Dioda podczerwona, sensor prawego koła

LED 14 Miga jedynie z bootloaderem

LED 14 Wyłączona bez bootloadera

LED 14 miga krótko po włączeniu zasilania, kiedy program jest załadowany.

3.2 UKŁAD PŁYTKI PCB ROBOTA AAR

Zobacz numery na Rys.: Płytką PCB AAR

1. Wejście na koszyk bateryjny.
(Uważaj przy podłączaniu! Sprawdź czy polaryzacja jest poprawna!)
2. Przełącznik On/Off Robota.
3. Dioda statusu: sygnalizuje czy Robot został zasilony ze źródła.
4. Jeśli używasz baterii wielokrotnego ładowania możesz podłączyć to podwójne wejście, który zasili robota poprawnie.
5. Podłączenie USB: aby programować robota za pomocą oprogramowania Arduino.
6. Przycisk Reset: aby ręcznie zresetować robota.
7. Podłączenie ISP: umożliwi zainstalowanie innego programu bootloader.
8. LED 14: dostępny do celów programistycznych, jeśli zamiga, oznacza to, że bootloader został uruchomiony.
9. Linefollower: ten moduł jest do twojej dyspozycji programistycznej, ponieważ pozwoli robotowi podążać za linią.
10. Czujnik koła lewego: ten moduł generuje impulsy zgodnie z tym jak obraca się lewe koło.
11. Czujnik koła prawego: ten moduł generuje impulsy zgodnie z tym jak obraca się prawe koło.
12. Diody statusu dla lewego silnika: Te diody wskazują czy silnik obraca się do przodu.
13. Diody statusu dla prawego silnika: Te diody wskazują czy silnik obraca się do tyłu.
14. Miejsce na rozszerzenia, gdzie, np. moduły bezprzewodowe mogą być przyłączone do układu Arduino.
15. Diody statusu dla interfejsu szeregowego RS232.
16. Dioda statusu 2: tę diodę można dowolnie programować.
17. Diody statusu dla komunikacji danych USB.
18. Kontrola silnika.

Na CD zawarty jest program autotestu AAR o nazwie AAR_SelftestV3.

Za jego pomocą możesz testować podstawowe funkcje robota ARR.

Użyj programu do monitoringu w twoim Arduino Software, nie zapomnij o instalacji bibliotek - zobacz na stronie 22.

3.3 AAR - DODATKOWE INFORMACJE

Dostępny jest interfejs fizyczny do podłączenia USB wraz z chipem FT232 IC. Chip ten przekształca sygnał z USB na sygnał RS232 UART, który później może być już przetworzony przez procesor ATMEGA328P (znajdujący się po prawej stronie na przodzie).

Po drugiej stronie umieściliśmy przełącznik ON/OFF, który zawiera złącze JP3 potrzebne do podłączenia zasilania oraz kontroler silnika IC2. Tylna część płytki drukowanej (PCB) zawiera oba silniki oraz czujniki kół.

Czujniki przy kołach to czujniki fotoelektryczne. W zębatkach kół co 90° umieszczone zostały otwory, czyli na zębatce są cztery takie dziury. W momencie, gdy światło dostaje się przez otwór do czujnika, wysłana zostaje do procesora o tym informacja w postaci impulsu. Dodatkowo obwody elektryczne włączają diody 16 i 17. Wysyłane impulsy pozwalają nam kontrolować prędkość każdego z tylnych kół napędowych.

Na przodzie znajdują się także złącza dla różnych rozszerzeń, a na spodzie płytki PCB można znaleźć czujniki na potrzeby obwodu line followera, czyli robota mobilnego podążającego za torem określonym przez linię.

Robot typu line follower używa diody, która emituje wiązkę światła na jego spodzie. Tu obok diody znajdują się dwa czujniki podczerwieni aby monitorować odbite światło ze spodu. Płytką PCB zawiera elementy (diody elektroluminescencyjne, rezystory i kondensatory), które razem dopełniają działanie modułu, który później można nazwać line followerem.

Robot używa układu Arduino, który może być porównany z płytką Arduino Duemilanove. Mikrokontroler ATMEGA328P jest rdzeniem układu, który ma aż 14 cyfrowych portów wejścia/wyjścia (I/O), wśród których sześć można konfigurować jako wyjściowe kanały PWM (z ang. Pulse-Width-Modulated). Dodatkowo robot zawiera jeszcze sześć analogowych kanałów wyjściowych, oscylator kryształowy 16MHz oraz złącze USB do programowania i sterowania. Kolejnym elementem jest złącze ISP, które umożliwi doświadczonym hobbystom programowanie własnego programu bootloadera. Robot zaprojektowano na napięcie zasilające 5V i może być też zasilany z wtyczki USB. Ta opcja jest wygodniejsza do testów oraz programowania.

Bardzo wygodną funkcją projektu tego robota są wtyki, które umożliwiają ci dodawanie twoich własnych modułów rozszerzeń serii modułów rozszerzeń ASURO.

3.4 OPROGRAMOWANIE ARDUINO - DODATKOWE INFORMACJE

Oprogramowanie Arduino to kategoria oprogramowania open source, które jest dostępne uniwersalnie do wszystkiego, np. kody źródłowe są ujawnione do wszystkich platform programistycznych.

Platforma programistyczna Arduino została zaopatrzona w edytor tekstu, okno powiadomień oraz konsolę. Platforma może komunikować się bezpośrednio z AAR, co pozwala na łatwe przesyłanie programów do procesora.

Programy napisane w języku Arduino nazywane są „sketch” (z ang. szkice). Zwykły edytor tekstu służy do rozbudowania i edytowania tych programów. Pliki „sketch” zapisują się na dysku twardym komputera. Są one rozpoznawalne poprzez rozszerzenie „.ino”. Zapisywane działania plików „sketch” są wyświetlane w oknie powiadomień, w którym również pojawiają się błędy wykryte w kodzie źródłowym. W prawym dolnym rogu okna znajdują się obecnie aktywne płytki Arduino oraz interfejs szeregowy. Podstawowa płytka Arduino dostarcza biblioteki z dodatkową funkcjonalnością. Biblioteka definiuje funkcje, które mogą zostać użyte podczas pisania programu bez konsumpcji dodatkowego miejsca, ani dodatkowego nakładu pracy programisty.

Program Arduino można podzielić na trzy sekcje:

- struktura,
- dyrektywy (dla zmiennych stałych),
- funkcje.

Struktura Arduino składa się z `setup()` oraz funkcji `loop()`. Część `setup` służy do inicjalizacji zmiennych, definicji pinów („Pin-Modes”) oraz definicji użytych bibliotek.

Funkcja `Loop` powtarzana będzie w nieskończonej pętli, która pozwala programowi wykonywać się nieprzerwanie, dopóki układ nie zostanie odłączony.

W programie używa się zmiennych, aby przechowywać w nich dane edytowalne, podczas gdy wartości stałe pozostają niezmiennie i zapisuje się w nich, np. funkcjonalność pinów I/O.

4. ROZPOCZĘCIE PRACY

4.1. Pobranie i instalacja oprogramowania

Zainstaluj wersję 1 oprogramowania Arduino z płyty. Później możesz pobrać najnowszą wersję oprogramowania ze strony Arduino.



WAŻNE:

Użycie różnych wersji oprogramowania Arduino i różnych wersji aplikacji może stworzyć pewne problemy. Bywa, że dla aktualnej wersji oprogramowania należy też zmienić aplikacje, bo bez tego może nie zadziałać!

4.2. Język Arduino

Struktura języka Arduino jest dokładnie opisana na oficjalnej stronie Arduino. Naucz się rozumieć charakterystyczne punkty języka do poziomu, który jest tobie potrzebny do pracy.

4.3 Instalacja sterownika USB

Kiedy podłączysz płytkę, Windows powinien rozpoznać nowe urządzenie i rozpocząć instalację sterownika (jeśli nigdy wcześniej nie używałeś na tym komputerze Arduino). Na wersjach Windows Vista i późniejszych, sterownik powinien sam się pobrać i zainstalować.

Wybierz urządzenie szeregowe z menu Narzędzia > Porty szeregowe. Będzie to zapewne port COM3 lub wyższy (COM1 oraz COM2 są zazwyczaj zarezerwowane dla sprzętu).

Aby to sprawdzić możesz odłączyć płytkę Arduino i otworzyć menu ponownie: port, który zniknie, to właśnie ten, pod który przypisane jest Arduino. Podłącz płytkę i wybierz port szeregowy.

4.4 Montaż baterii

Projekt robota wymaga czterech baterii "paluszków", każda po 1,5V. Jeśli wolisz użyć baterii wielokrotnego ładowania, należy zamontować zworkę JP4 jako mostek, aby przygotować system na niższe zasilanie z tych baterii (zobacz Rys. 1, numer 4).



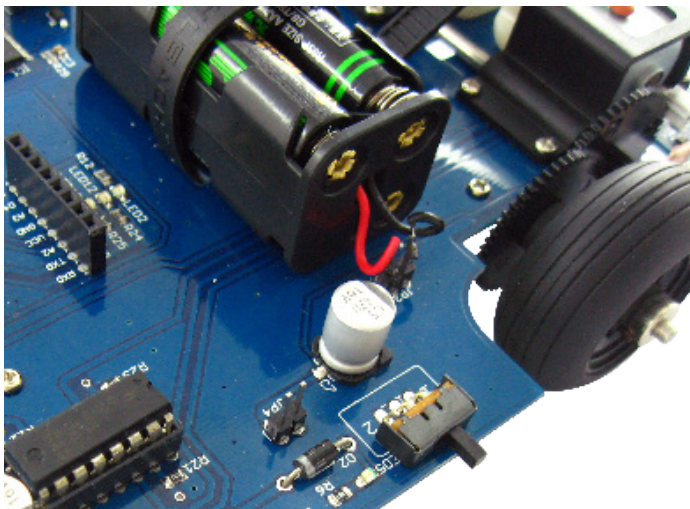
UWAGA!

Montaż zworki JP4 może wpłynąć na polaryzację baterii. Sprawdź, czy wszystko zostało umieszczone poprawnie. Dla bezpieczeństwa umieszczona została tam też dioda prostownicza. Bądź ostrożny! Błędy w podłączeniu zasilania za pomocą zworki JP4 mogą poważnie uszkodzić robota.

Koszyk bateryjny zamontuj jak poniżej (Rys. 2).



Rys. 2:
Montaż baterii.



Teraz możesz włączyć robota za pomocą przełącznika ON/OFF. Leżąca w pobliżu przełącznika dioda zasilania (LED5) powinna się zaświecić.

4.5 Programowanie z programami Arduino

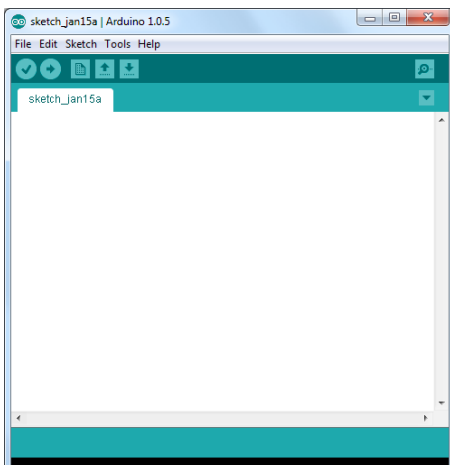
Podłącz robot do komputera za pomocą kabla USB. Jeśli układ Arduino jest podłączony do portu USB, robot nie potrzebuje baterii - czerpie zasilanie z komputera.



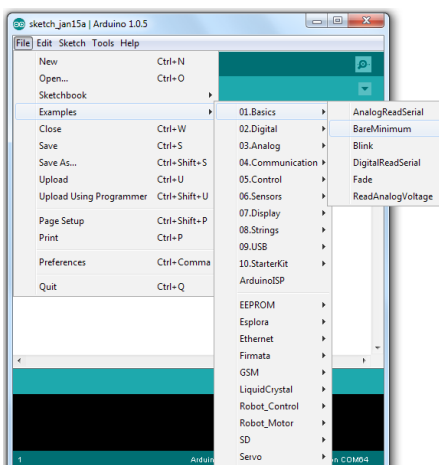
UWAGA:

Robot zawsze będzie się aktywował jak tylko podłączony zostanie do komputera. Przełącznik ON/OFF oraz dioda zasilania LED5 będą aktywne tylko w przypadku operacji wykonanych z baterią.

Teraz otwórz oprogramowanie Arduino (zobacz Rys. 3a).



Rys. 3a Oprogramowanie Arduino.



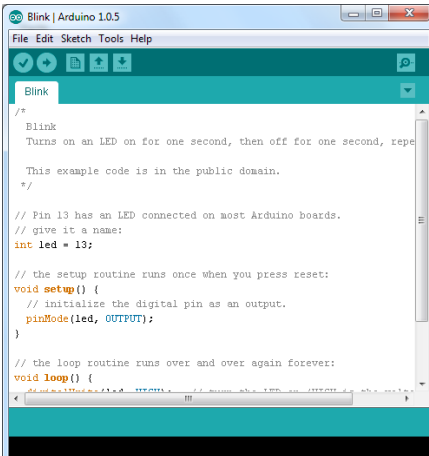
Rys. 3b Otwieranie pustego programu

4.6 Wybór programu Arduino

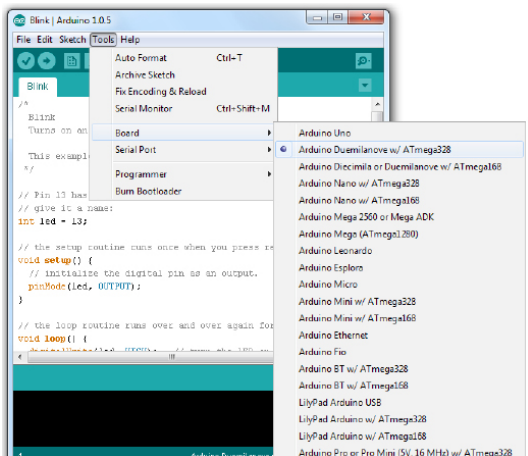
Zacznijemy od załadowania do robota prostego programu próbnego nazwanego „blink”. Program sprawi, że dioda 1 (LED1) zapali się wielokrotnie.

Żałduj program poprzez wyszukanie i kliknięcie programu w menu oprogramowania:

File > Examples > 1. Basics > Blink (zobacz Rys. 3b), który po kliknięciu wyświetli się w oknie (Rys. 4a).



Rys. 4a Program Blink.



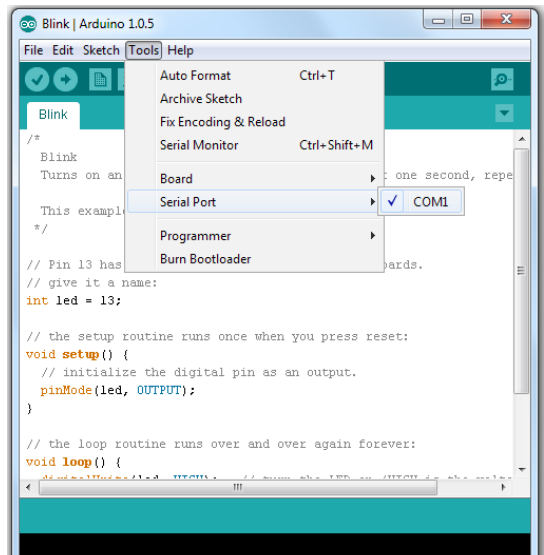
Rys. 4b Wybór płytki.

Na tym etapie należy jeszcze wybrać odpowiednią płytkę Arduino w menu: **Tools > Board > Arduino Duemilanove LUB Nano w/Atmega328** (jak na Rys. 4b).

4.7 Wybór poprawnego portu COM

W następnym kroku należy zdefiniować poprawny port COM dla interfejsu Arduino. Będzie to zapewne COM3 lub wyższy (COM1 i COM2 są już w użyciu).

Aby wybrać port COM otwórz menu: **Tools > Serial Port > COM x**. (zobacz Rys. 5)



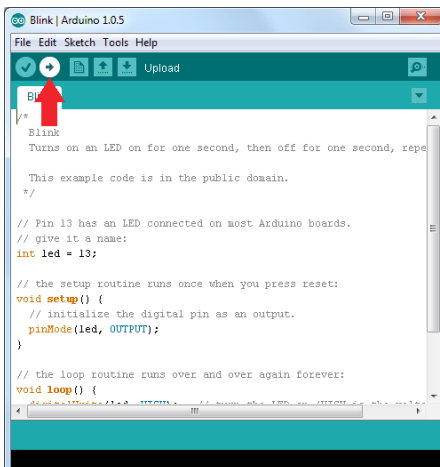
Rys. 5 Wybór poprawnego portu COM.

4.8 Wgrywanie programu do Robota Arduino

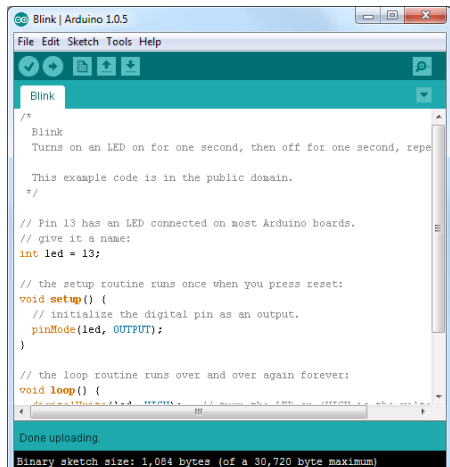
Kliknij przycisk, który zaznaczony został czerwoną strzałką (lub zamiennie wybierz z menu: „File > Uploading to I/O board”), aby wgrać wybrany program do podłączonego robota Arduino (Rys. 6a).

Okno statusu na bieżąco wyświetla jak przebiega proces kompilacji programu i gdy zakończy się on z sukcesem, rozpocznie się przesyłanie programu do układu.

Na koniec wgrywania programu w oknie pojawi się komunikat: „Done uploading” (jak na Rys. 6b), a także dioda 12 (LED12) zaświeci się krótko.



Rys. 6a) Wgrywanie programu.



Rys. 6b) Skończone wgrywanie.

Teraz możesz usunąć kabel USB, aby odłączyć go od komputera. Podłącz baterię i włącz robota.

Aby zaczerpnąć dalszych informacji oraz pobrać inne produkty odwiedź jedno z forum na stronach:

www.arexxx.com
www.roboternetz.de

4.9 Rozwiązywanie problemów

NAJPIERW sprawdź czy robot otrzymuje wystarczająco zasilania z portu USB. Da się to sprawdzić usuwając baterie i zamykając zworkę JP4 - powinna zapalić się dioda zasilania.



PROBLEM; nie ma połączenia z robotem AAR.

Sprawdź czy diody 3 i 4 migają krótko kiedy podłączasz robot do komputera kablem USB.

Czy robot jest podłączony z komputerem sprawdź w managerze urządzeń (panel sterowania).

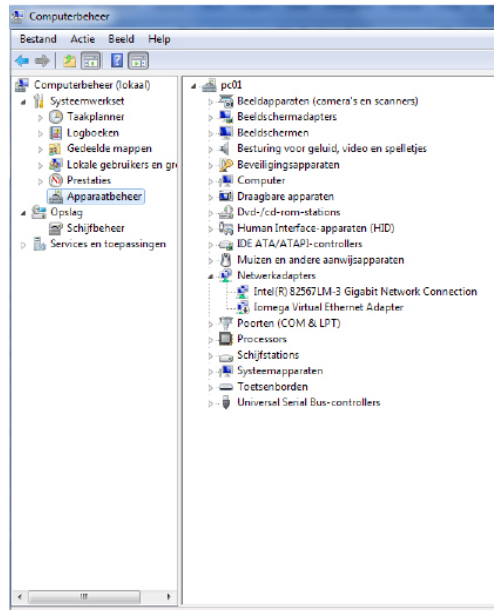
Rys. 6c)

Manager urządzeń (na zrzucie ekranu duńska wersja WIN7).

Port Robot FTDI jest zwykle pokazany w sekcji portów jako **“USB SERIAL PORT”**.

Sprawdź czy sterownik dla poprawnego FTDI został zainstalowany.

To jest właśnie najczęstszy problem. Pobierz i zainstaluj najnowszą wersję sterownika FTDI na komputer.



PROBLEM; nie da się wgrać pliku .INO do robota.

Sprawdź czy wybrany został odpowiedni port COM. Będzie to port różny od COM1 i COM3.

Sprawdź czy wybrałeś odpowiedni typ płytki Arduino z ATmega328.

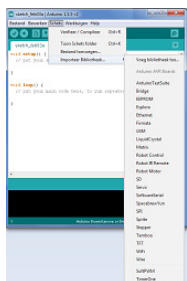
Sprawdź czy plik sketch posiada zainstalowane wszystkie niezbędne biblioteki. Wejdź w **Menu entry "sketch-->Library -->** (zobacz na stronie 22).

4.10 Autotest AAR

Z płyty CD otwórz program AAR_selftest i wgraj plik .ino do robota AAR. Aby wgrać wybrany program do podłączonego robota Arduino kliknij opcję, która na Rys. 7 została oznaczona czerwoną strzałką (lub zamiennie wejdź w menu "File > Uploading to I/O" board).

Ładowanie bibliotek

Ładowanie bibliotek wykonuje się poprzez menu "sketch"> "Library" > **SoftPWM** oraz **TimerOne** z CD (przykłady).

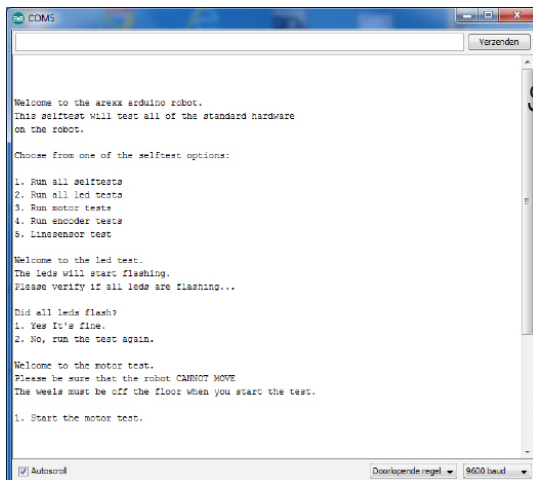


Rys.6d) biblioteki

Otwórz biblioteki (wersja duńska)



Rys. 7 Wgrzywanie programu.



SV1

Rys. 8 Serial monitor.

Otwórz serial monitor tak jak pokazano na Rys. 8.

Pilnuj, by połączenie z robotem kablem USB nie zostało przerwane! Upewnij się, że baterie są na miejscu i są naładowane. Włącz zasilanie robota przełącznikiem ON/OFF.

Rozpocznij pierwszy test i podążaj za krokami w serial monitor.

5 MODUŁY ROZSZERZAJĄCE AAR

5.1 Moduły rozszerzające ASURO dla Robota Arduino

Robot AREXX Arduino posiada wiele różnych połączeń dla modułów rozszerzających. Piny złączy AAR są również zgodne z robotem AREXX ASURO (zestaw programowalny w języku C). Oznacza to, że można używać tych samych rozszerzeń zarówno do zestawu AREXX ASURO, jak również i do niniejszego robota Arduino.

Na płycie znajduje się kilka przykładowych programów Arduino, abyś mógł rozpocząć pracę z rozszerzeniami od razu.

Robot ASURO również posiada ogólny zestaw do eksperymentowania (ARX-EXP2), którego można używać do swoich własnych rozszerzeń Arduino.



Poniżej znajduje się rozkład PINÓW złączy zestawu rozszerzeń AAR. Na płytce złącza te są oznaczone SV8, SV9 oraz SV10.

SV1

1	2	3	4	5	6	7	8
---	---	---	---	---	---	---	---

1. D8
2. D4
3. A3
4. GND
5. VCC
6. A2
7. TXD
8. RXD

SV8, SV9 oraz SV10

1	2	3
4		
5		
6	7	8

1. SCK/PB5
2. SCL/ ADC5/PC5
3. WS_RIGHT/INT0/PD2
4. RGND
5. PWM1_L/AIN0/OC0A/PD6
6. MOSI/OSC2A/PB3
7. MISO/PB4
8. SDA/ ADC4/PC4

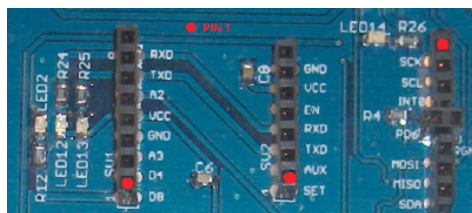
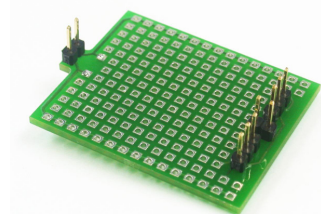
SV2

1	2	3	4	5	6	7
---	---	---	---	---	---	---

1. SET
2. AUX
3. TXD
4. RXD
5. EN
6. VCC
7. GND

ARX-EXP2

AREXX PŁYTKA DO
FKSPRYMENTÓW



Rys. 9. Styki dla rozszerzeń
Czerwona kropka to pin 1

5.2 Konfiguracje rozszerzeń ASURO

ARX-ULT10

AREXX ULTRA SOUND KIT

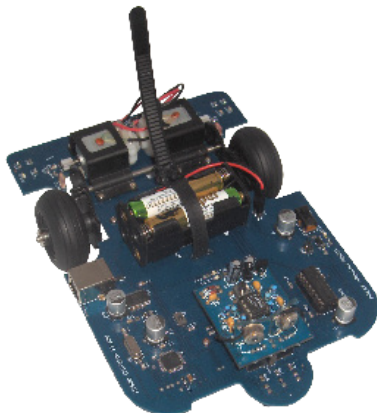
Użyj przykładowego programu Arduino **AAR_ULTRA** z CD. Z tym programem robot cofa się, gdy napotka przeszkodę.

ARX-SNK20

AREXX SNAKE VISION (PYRO) KIT

Użyj przykładowego programu Arduino **AAR_SNAKE** z CD.

Z tym programem robot będzie podążał za ciepłym obiektem.



ARX-DSP30

AREXX DISPLAY KIT

Użyj przykładowego programu Arduino **AAR_DISPLAY** z CD.

Z tym programem na wyświetlaczu pokazuje się tekst, a przyciski można dowolnie zaprogramować.

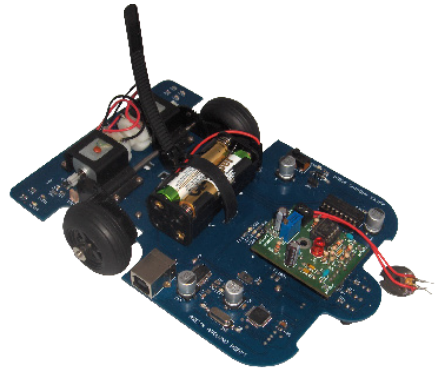


ARX-MNSP55

AREXX MINE SWEEPER KIT

Użyj przykładowego programu Arduino **AAR_MINE** z CD.

Z tym programem robot będzie wykrywał metal.



ARX-APC220

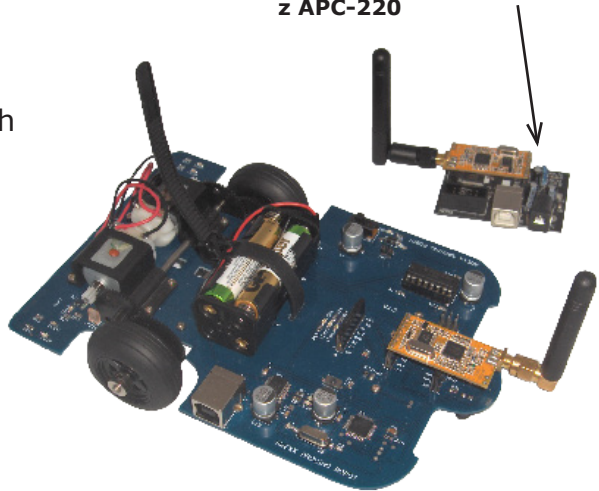
AREXX WIRELESS KIT

Zobacz program **AAR_APC220** spośród przykładowych plików na CD.

Użyj przykładowego programu Arduino **AAR_SRC**.

Potrzebny będzie też program Visual Basic na komputerze (**AAR_VB**).

RP6v2 USB Programmer z APC-220



Z zestawem AREXX ARX-APC-220 można sterować robotem AAR bezprzewodowo.

Poniżej opisane zostało, jak to działa:

- Podłącz RP6v2 USB Programmer oraz APC-220 do komputera.
- Podłącz inny APC do robota AAR (środkowe złącze).
- Załaduj dane **AAR_SRC** do procesora robota AAR.
- Zainstaluj oprogramowanie **AAR_VB** na komputerze.
- Wybierz odpowiedni port COM w oprogramowaniu AAR-VB.

ANDROID – PROGRAM

ARX-BT3

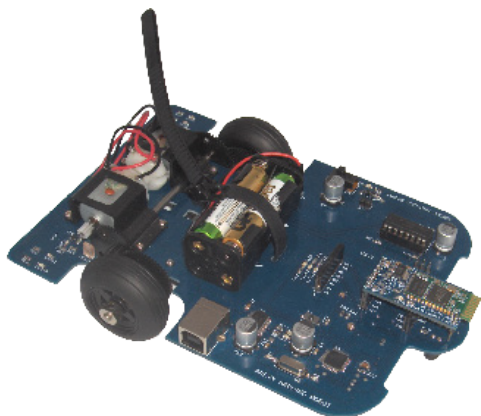
AREXX BLUETOOTH KIT

Zestaw ARX-BT03 umożliwia ci sterowanie robota AAR poprzez Bluetooth i aplikację na ANDROID.

Otwórz plik **AAR_Android app** z płyty i zainstaluj aplikację **BLUETOOTH_AAR.APK** na swoim urządzeniu. Z tego samego pliku zainstaluj plik AAR_SRC_Bluetooth na robocie AAR.

Podążaj za instrukcjami poniżej:

- Podłącz moduł Bluetooth do płytki PCB robota.
- Załaduj dane Bluetooth ANDROID do procesora robota.
- Załaduj dane ANDROID APK do swojego smartfona lub tabletu.



Bluetooth®

Możesz znaleźć też aplikacje AREXX na płycie pod nazwą **AAR_Android app** lub w sklepie GOOGLE PLAY.



6. POBOCZNE INFORMACJE NA TEMAT OBWODÓW MOSTKA H

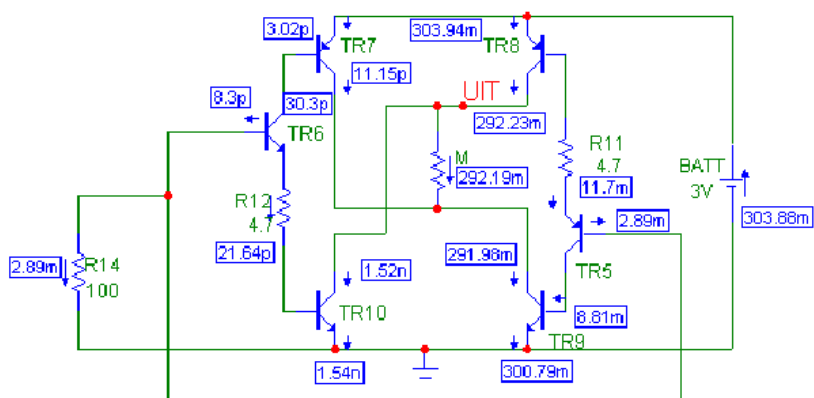
Mostek H to obwód elektroniczny, który odwraca polaryzację urządzenia (np. silnika prądu stałego - DC) poprzez sterowanie czterema przełącznikami. Bardzo często używa się go w robotyce do sterowania obrotami silnika w dwie przeciwne strony. Nowoczesne systemy używają układów scalonych do sterowania silnikiem, ale aby poznać podstawowe problemy związane z napięciem zasilającym kluczowym może się okazać użycie archaicznego układu do sterowania.

6.1 Mostek H dla napięcia zasilającego 3 Volt

Obwód sterujący w robotach Hyper Peppy zawierają dwa tranzystory PNP TR7 oraz TR8, a także tranzystory NPN TR9 oraz TR10. W tym projekcie pracę zezwalamy jedynie dwóm tranzystorom, aby przewodziły prądy do silnika M jednocześnie:

przez TR7 i TR10 lub zamiennie przez TR8 i TR9.

Symulator Microcap (darmowa wersja testowa tego programu) to program, który pozwala na łatwą i wygodną symulację układu DC oraz odczyt wartości z okna schematu:



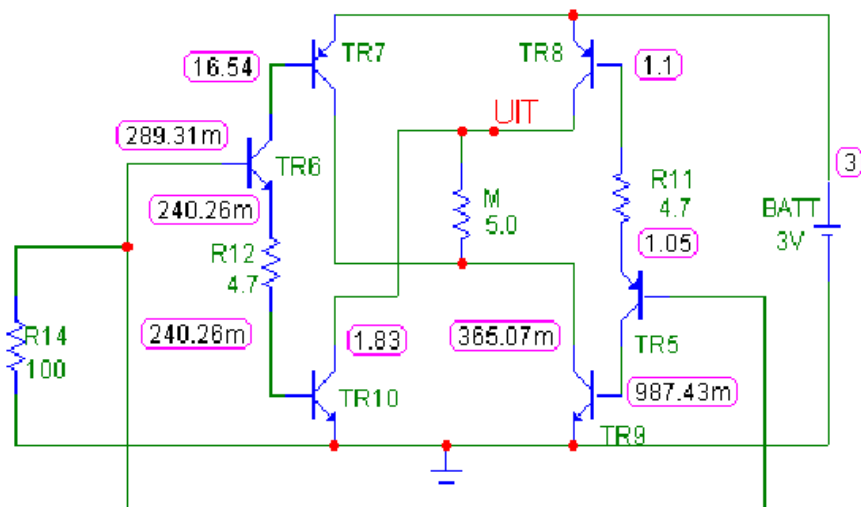
Rys. 10: Symulacja mostka H robota Hyper Peppy.

Przyjrzyjmy się silnikowi prądu stałego M. W obwodzie rezystor R14 imituje działanie przedwzmacniacza. Ten rezystor ma później wpływ na wyprowadzenia tranzystorów nazywane bazą, a konkretnie na tranzystory TR6 i TR5, których napięcie bazy będzie równe 0V. Z tego powodu prawa gałąź przewodzi istotne prądy.

Tranzystory TR8, TR5 i TR9 przewodzą, a inne są w tym czasie zablokowane. Jak tylko R14 dostanie napięcie dodatnie, prawa gałąź zostanie zablokowana i prąd na silniku zostanie odwrócony.

Symulator Microcap oblicza prądy dla wszystkich komponentów oraz odczytuje wartości z okna schematów. Cały prąd zasilania 3-woltowej baterii będzie wynosił około 300mA (mili amperów).

Tak niezwykle niskie napięcie zasilające tego obwodu zależy od kombinacji krzemu tranzystorów PNP i NPN, które pracują przy napięciu przewodzenia równym 0.7V. Jednakże silnik umieszczony został pomiędzy dwoma pinami tranzystora nazwanymi kolektorami, które jedynie w stanie nasyconym przewodzą przy napięciu 0.3V. Dla silnika M te przełączenia napięcia następują przy szacowanym napięciu 1.5V. Wartości obliczone przez Microcap widnieją na Rys. 11.



Rys. 11: Ustawienia silnika DC dla mostka H w robocie Hyper Peppy.

3-woltowe napięcie zasilające jest idealnym rozwiązaniem dla robota, który swoje napięcie czerpie z tylko dwóch paluszków baterii. Trudno jednak zawrzeć tranzystory PNP w układzie scalonym takim jak L293D. Układy scalone (IC - ang. integrated circuit) mają wiele plusów: niezawodność, ochrona przed nieodpowiednim budowaniem obwodów, a także mały rozmiar i waga. Z tego powodu zdecydowaliśmy się użyć chipu L293D z obwodem podwójnego mostka H, aby móc jednocześnie sterować dwoma silnikami prądu stałego.

6.2 Mostek H dla 4,5 Volt

Chip L293D (Rys. 10) pozwala na sterowanie prądami wyjściowymi aż do 600mA na każdy kanał (prąd maksymalny: 1.2A). Napięcie zasilające sterownika (VCC2) może wahać się między 4.5V a 36V, co czyni chip L293D jednym z najlepszych do sterowania silników DC.

Minimalne napięcie zasilające (VCC2) wynosi właśnie 4.5V, co poniekąd zmusza nas do wybrania aż czterech baterii jako źródło zasilania. Takie wymogi zwiększają również wagę całkowitą robota. Taką cenę płaci się za użycie nowoczesnych rozwiązań dla obwodów.



Rys. 12 Obwód z mostkiem H oraz chipem L293D.

7. Odometria

Ten rozdział poświęcony jest interesującym pomysłom na aplikację dla robota AAR dla osób, które są zainteresowane pogłębieniem tematu projektów związanych z robotyką.

7.1 Reakcja na światło i kolor

Sensory światłoczułe pozwolą nam programować robota, aby reagował na linię oraz kolory. W pierwszym z przedstawionych przykładów robot powinien jeździć po torze w kształcie ósemki. Program napisany został w pętli, czyli robot jest zmuszony do nieprzerwanej pracy.

Drugi i trzeci przykład uczy robota unikać źródeł czerwonego światła oraz jednocześnie Ignąć do zielonego. Takie zachowanie jest znane z życia, m.in. owadów, które dzięki temu potrafią przetrwać.

7.2 Robot meloman

Interesującym można odnaleźć także wzór zachowań, które zależą od hałasu w jego środowisku. Niespokojnego robota wyposażonego w wrażliwy mikrofon można "nauczyć" jak unikać ostrej muzyki z basem i jednocześnie Ignąć do wysokiej muzyki fletowej. Jego zainteresowanie wysokimi częstotliwościami dźwięku mogą nawet być ważniejsze niż jego strach przed tymi niskimi. W ten sposób robot będzie miał za zadanie uciekać od źródła muzyki heavy metalowej i kierować się w stronę wysokiej muzyki fletu.

Wzory zachowań, które zależą od wysokich i niskich częstotliwości, światła oraz kolorów, wymagają jedynie kilku czujników, dwóch filtrów częstotliwości i kilku czujników światła wyposażonych w filtry koloru.

7.3 Złożone roboty typu line follower

Roboty, które podążają za światłem lub po torze krzywej wymagają źródła światła, np. diody lub większej ilości czujników światła. Takie urządzenia pozwalają układowi na identyfikację linii, po której robot ma podążać. Pierwszym krokiem w programie powinny być instrukcje pozwalające robotowi na początkowe odnalezienie linii, która tworzy wzór, po którym robot ma się poruszać. Taki wzór może się składać z instrukcji, gdzie wzorem jest, np. spirala, czyli okrąg o zwiększającym się promieniu. Robot będzie podążał za linią toru wyznaczonym przez linię dopóki czujnik tę linię będzie wykrywał. Pisanie takich programów należy do osobnej kategorii oprogramowania i rozwiązuje problemy podążania za każdym, nawet trudnym wzorem, np. z ostrymi zakrętami itd.

7.3.1 Złożone zachowania (jako zadanie programistyczne)

Projekt może być spomplikowany, np. jeśli musi pracować wśród skomplikowanego, wielokolorowego otoczenia oraz linii, a za zadanie ma podążanie za czerwoną linią, która zaprowadzi go do bezpiecznego i cichego "garażu".

Jak tylko hałas ustanie, robot może opuścić "garaż" i zacząć szukać zielonej linii prowadzącej do kolejnego, pełnego muzyki.

Jeśli muzyka zawiera w sobie wysokie częstotliwości, których robot nie lubi, opuści to miejsce w poszukiwaniu drogi do domu, czy pierwszego bezpiecznego "garażu".

Doświadczeni programiści zdają sobie sprawę ze złożoności oraz wymogów projektowych programu, który wymaga wielu różnych zależności od kolorów i od dźwięków. Programiści stoją wtedy przed zadaniem stworzenia programu, który opierać się będzie na hierarchicznie wykonujących się funkcjach. Stworzenie takiego konceptu pozwala na napisanie złożonego i rzetelnego oprogramowania, które wykonywać będzie konkretne zadanie. Złożoność oprogramowania może sprawić, że programista zacznie podziwiać złożoność małych organizmów żywych, które przecież korzystają z takich mechanizmów w codziennym życiu podczas poszukiwania pożywienia lub schronienia. Natura sama wie najlepiej, jakie mechanizmy są najdoskonalsze i ulepsza je, by przetrwać mogli nawet teoretycznie najłabsi!



8. Programowanie w Boot-loader

UWAGA!

Opisane w tym rozdziale opcje wymagają dużego doświadczenia programistycznego.



Możesz załadować do mikrokontrolera Arduino-bootloader przy pomocy STK500.

Aby wgrać do mikrokontrolera Atmega jakikolwiek program, który został napisany w języku Arduino, procesor Atmega musi być wyposażony w specjalny Arduino-bootloader. Ten boot-loader zadba o poprawną lokalizację wszystkich użytych znaków wewnątrz pamięci Atmega.

Aby zainstalować boot-ladera, potrzebne będą:

- * płytką programowalną AVR (na przykład płytką STK500)
- * 12-woltowe napięcie zasilające (dla STK500)
- * Robot AAR zawierający na płytce złącze ISP (Rys. 7)
- * PC z fizycznym portem COM (lepiej, żeby nie było to konwerter USB-RS232, który może powodować błędy o przekroczeniu limitu czasu).

Należy zainstalować aktualną (lub zaktualizować istniejącą) wersję programu Arduino, który można znaleźć na stronie internetowej www.arduino.cc. Pobrany plik będzie miał rozszerzenie .ZIP lub .RAR. Rozpakuj te pliki i umieść gdzieś na dysku twardym.

Aby załadować Arduino-bootloader do robota użyj np. programu WINAVR.



Rys. 12: Złącze ISP

Uwaga!

Oprogramowanie ARDUINO jest typu freeware i od czasu do czasu może nie współpracować prawidłowo z bootloaderni Arduino!

Jeśli doświadczysz problemów tego typu lub podobne, możesz poszukać pomocy na wielu dostępnych forum i stron internetowych Arduino, gdzie być może ktoś miał już podobny problem.

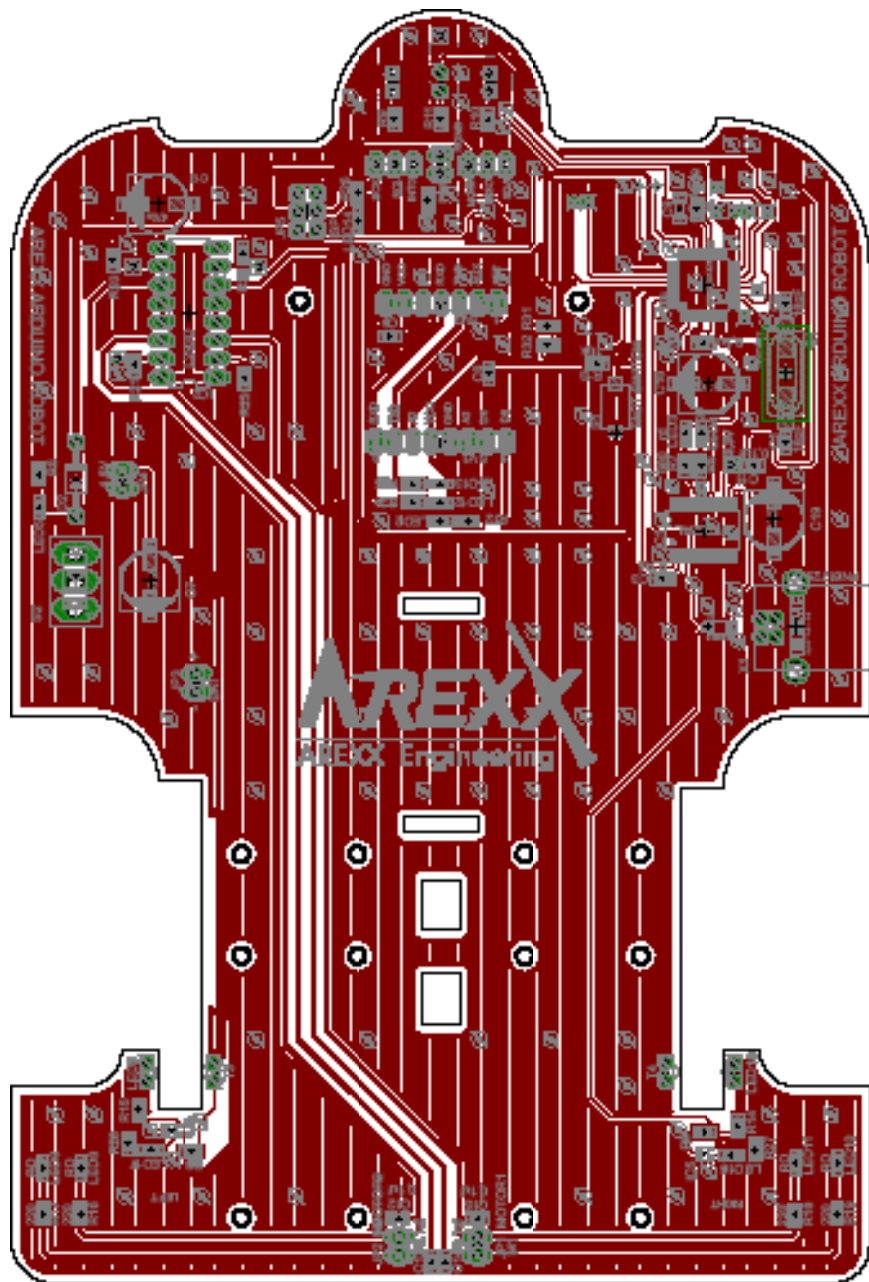
DODATEK

A. LISTA ELEMENTÓW

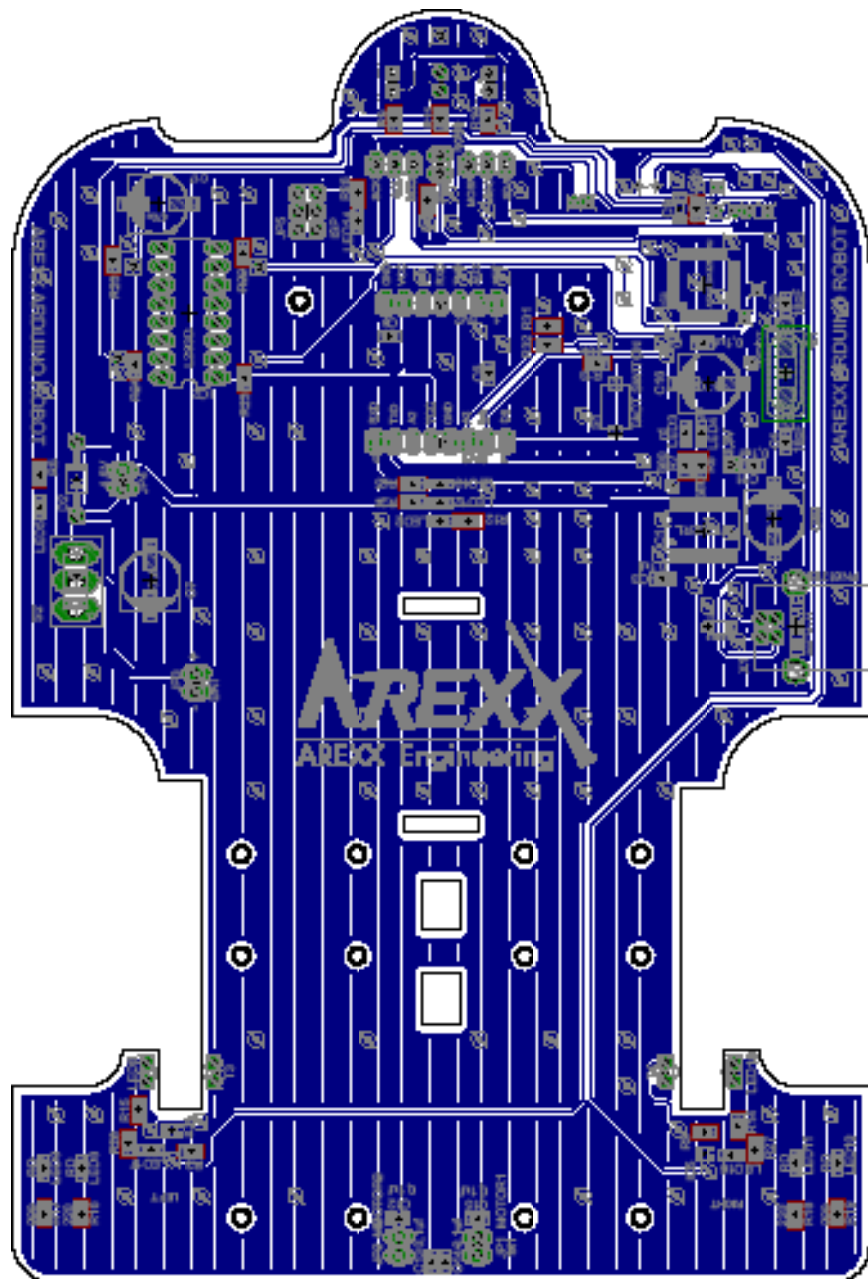
Element	Wartość	Pakiet
C1	18pF	0805
C2	18pF	0805
C3	0.1uF	C0805K
C4	0,1uF	0805
C6	0,1uF	0805
C7	470uF	CPOL-USF
C8	0,1uF	0805
C9	4,7uF	1206
C11	0,1uF	0805
C12	0,1uF	0805
C13	0,1uF	0805
C14	0,1uF	0805
C15	0,1uF	0805
C16	470uF	CPOL-USF
C17	470uF	CPOL-USF
C19	470uF	CPOL-USF
D1	MBR0520	SOD-123
D2	1N4001	DO41-10
IC1	FT232RL	SSOP28
IC2	L293D	DIL16
IC3	ATMEGA328AU	ATMEGA328P-AU
IC4	74AHC1G14DCK	74AHC1G14DCK
IC5	74AHC1G14DCK	74AHC1G14DCK
JP1	M1	1X02
JP2	M2	1X02
JP3	BAT	1X02
JP4	4,8V	1X02
JP5	ISP	2X03
SV2	złącze żeńskie	FE07-1
T1	SFH300	LED5MM
T2	SFH300	LED5MM
T3	LPT80A	LPT80A
T4	LPT80A	LPT80A
U\$1	3,3V	PIN-T
U\$2	FE03-1	FE03-1
U\$3	FE03-1	FE03-1
U\$4	FE02-1	FE02-1
X1	PN61729-S	PN61729-S
LED1	Rd	LED5MM
LED2	BI	LEDCHIP-LED0805
LED3	Rd	LEDCHIP-LED0805
LED4	Gn	LEDCHIP-LED0805
LED5	BI	LEDCHIP-LED0805
LED6	Rd	LEDIRL80A

Element	Wartość	Pakiet
LED8	Rd	LEDCHIP-LED0805
LED9	Rd	LEDCHIP-LED0805
LED10	Rd	LEDCHIP-LED0805
LED11	Rd	LEDCHIP-LED0805
LED12	Gn	LEDCHIP-LED0805
LED13	Rd	LEDCHIP-LED0805
LED14	BI	LEDCHIP-LED0805
LED16	Rd	LEDCHIP-LED0805
LED17	Rd	LEDCHIP-LED0805
LED18	Rd	LEDIRL80A
Q1	16MHz	CRYSTALHC49UP
R1	20k	R-US_R0805
R2	20k	R-US_R0805
R3	1k5	R-US_R0805
R4	220	R-US_R0805
R5	1k5	R-US_R0805
R6	1k	R-US_R0805
R7	680	R-US_R0805
R8	680	R-US_R0805
R9	20k	R-US_R0805
R10	20k	R-US_R0805
R11	220	R-US_R0805
R12	220	R-US_R0805
R13	10k	R-US_R0805
R14	220	R-US_R0805
R15	220	R-US_R0805
R16	220	R-US_R0805
R17	220	R-US_R0805
R18	220	R-US_R0805
R19	220	R-US_R0805
R20	10k	R-US_R0805
R21	10k	R-US_R0805
R22	10k	R-US_R0805
R23	10k	R-US_R0805
R24	220	R-US_R0805
R25	220	R-US_R0805
R26	220	R-US_R0805
R27	220	R-US_R0805
R28	220	R-US_R0805
R29/C3	0.1uF	C0805
R31	10k	R-US_R0805
R32	12k	R-US_R0805
S1	TACT SWITCH	TACT_SWITCH
S2	255SB	255SB
SV1	złącze żeńskie	FE08-1

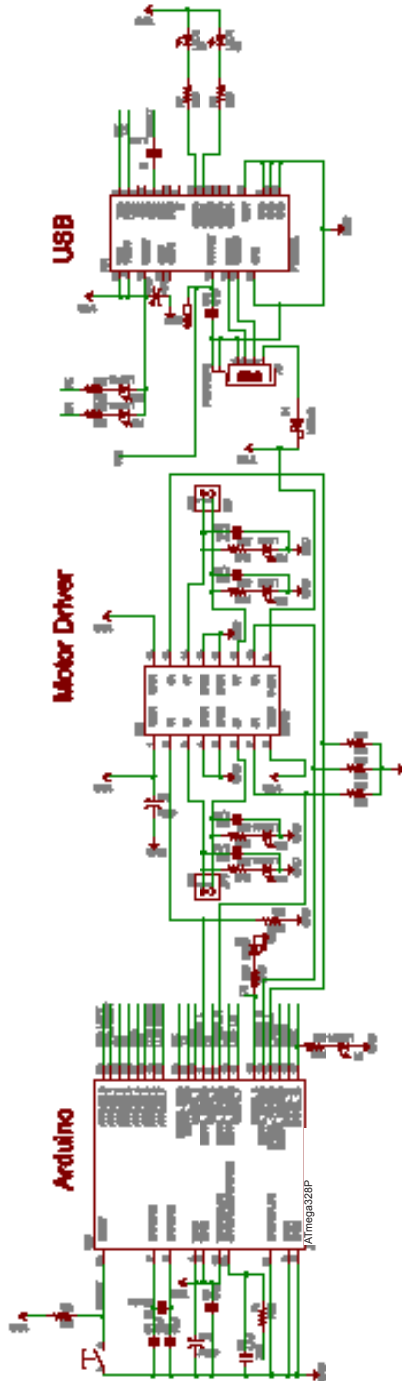
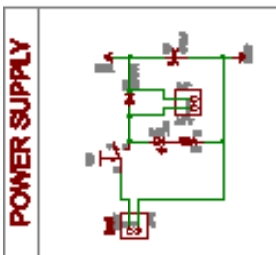
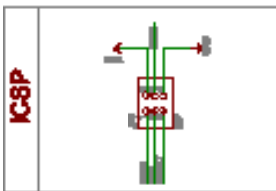
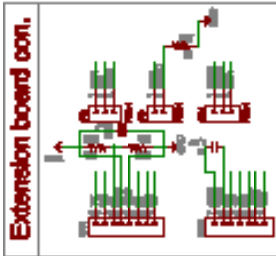
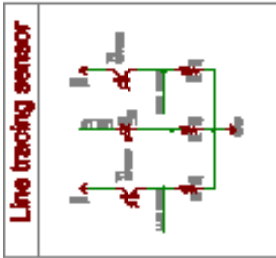
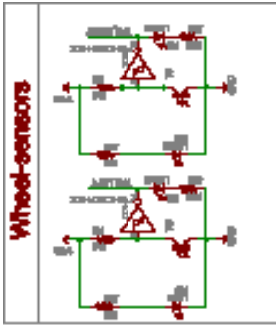
B. GŁÓWNA PŁYTKA PCB - RZUT Z GÓRY



C. GŁÓWNA PŁYTKA PCB - RZUT Z DOŁU



D. SCHEMATY AAR



E. 3D PCB AAR

