



KoNaR

KOŁO NAUKOWE ROBOTYKÓW

ROBOT KLASY FEMTOSUMO

DANIEL DUDZIK

KOŁO NAUKOWE ROBOTYKÓW KoNaR

WWW.KONAR.VOT.PL

7 KWIETNIA 2014

1. Wstęp

W dokumencie omówiona jest budowa skonstruowanego w ramach projektu inżynierskiego robota klasy femtoSUMO o nazwie Leo. Przedstawiono jego elementy funkcjonalne i ich właściwości. Pokróćce omówiono również oprogramowanie robota Leo.

Robot Leo pokazany na rysunku 1, składa się z:

- płyty głównej,
- panelu przedniego,
- podwozia,
- źródła zasilania.

Płyta główna zawiera jednostkę centralną robota, część sekcji zasilania, sterownik silników oraz dwie diody LED. Została ona wykonana z flexPCB o grubości 0.22mm. Panel przedni zawiera dwa czujniki odległości służące do detekcji przeciwnika, dwa czujniki białej linii oraz kilka elementów biernych. Projekt 3D panelu przedniego przedstawia rysunek 2. Panel przedni został wykonany podobnie jak płyta główna z flexPCB o grubości 0.22mm. Elementami podwozia są silniki i koła. Rolę źródła zasilania pełni jedno ogniwo Li-pol. Robot wraz z akumulatorem przedstawiony został na rysunku 3.

2. Elementy funkcjonalne robota

2.1. Układ detekcji przeciwnika

Układ detekcji przeciwnika robota Leo zrealizowano wykorzystując dwa optyczne czujniki odległości VCNL4020 [18]. Zamieszczono je w górnej części panelu przedniego jak pokazano na rysunku 4. Wybrano te układy, ponieważ charakteryzują się one odpowiednimi wymiarami oraz dogodnym ułożeniem wyprowadzeń.

Czujniki VCNL4020 posiadają interfejs I²C, którego szerszy opis można znaleźć na stronie [20]. Adres *slave* układu VCNL4020 26_h jest na stałe wpisany w strukturę układu bez możliwości jego zmiany. Z kolei, aby na magistrali I²C możliwa była komunikacja między układem *master* i dwoma układami podrzędnymi, układy te muszą mieć różne adresy. Mimo że zastosowany w robocie mikrokontroler posiada interfejs I²C, brak możliwości zmiany adresu układów VCNL4020 wymusił ich podłączenie do uniwersalnych pinów I/O i programowe zaimplementowanie dwóch niezależnych protokołów I²C. Takie rozwiązanie pozwoliło na komunikację mikrokontrolera z każdym z układów. Dodatkową zaletą było ułatwienie prowadzenia ścieżek na obwodzie drukowanym, ponieważ zaimplementowane interfejsy I²C mogły korzystać z dowolnych pinów I/O.

Aby układy VCNL4020 mogły pracować, należy je odpowiednio skonfigurować. W tym celu zgodnie z tabelą 1 zmieniono wartości kilku rejestrów, wartości pozostałych rejestrów pozostawiono bez zmian. Pomiar odległości wykonywany jest z zadaną w rejestrze 82_h częstotliwością, a jego wartość przechowywana jest w 16-bitowym rejestrze PROXIMITY RESULT o adresie 87_h. Deklarowany przez producenta zakres pracy 1–200mm okazał się nieosiągalny, ponieważ prąd płynący przez diodę IR należało ograniczyć, by światło odbite od ringu nie wpływało na wartości pomiarów.

Tabela 1. Wybrane rejestry konfiguracyjne układu VCNL4020

adres rejestru	nazwa rejestru	wybrana wartość	opis
80 _h	COMMAND	83 _h	odblokowanie wewnętrznego oscylatora aktywacja czujnika odległości
82 _h	PROXIMITY RATE	04 _h	31.25 pomiarów/sekundę
83 _h	IR LED CURRENT	14 _h	prąd diody emitera podczerwieni 100mA

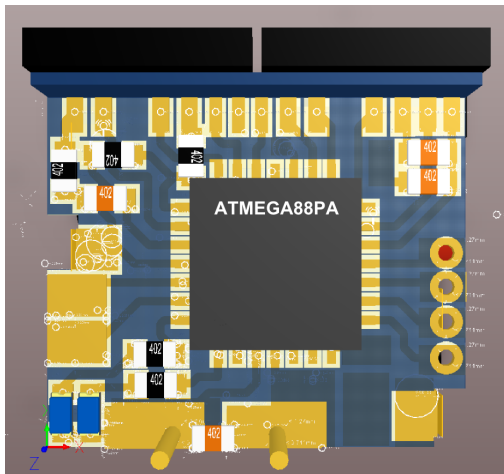
2.2. Układ detekcji białej linii

Do budowy układu detekcji białej linii w robocie Leo wykorzystano dwa czujniki GP2S60 [19]. Rysunek 5(a) przedstawia szczegółowe wymiary tego czujnika podane w milimetrach. Wymiary elementu pozwalają na umieszczenie dwóch takich czujników na spodzie panelu przedniego robota jak pokazano na rysunku 6.

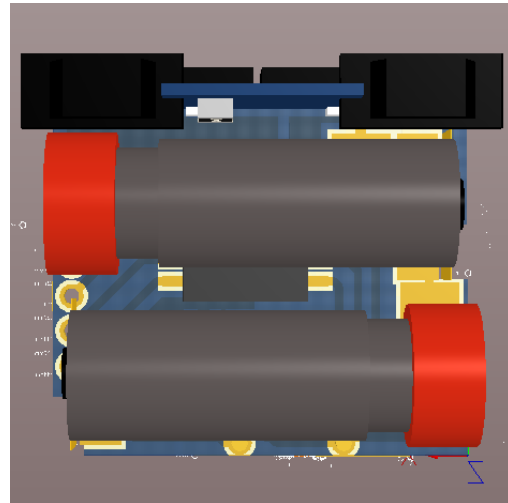
Schemat podłączenia czujnika ilustruje rysunek 5(b). Jak widać nie wymaga on wielu elementów zewnętrznych. Rezystor R_D ogranicza prąd płynący przez emiter podczerwieni. Dobrano jego wartość w taki sposób, by przy maksymalnym napięciu zasilania, które wynosi 4.2V, uzyskać prąd diody $I_F=20\text{mA}$. Opornik R_L służy ograniczeniu prądu płynącego przez detektor. Od jego wartości zależy zakres napięcia wyjściowego czujnika. Wartość opornika R_L należy tak dobrać, by zakres napięcia wyjściowego był możliwie jak największy. Wyjścia czujników zostały podłączone do wejść przetwornika ADC mikrokontrolera, dzięki czemu możliwy jest pomiar wartości napięcia, które jest odzwierciedleniem natężenia światła odbitego od podłoża.

Czujniki umieszczono na spodzie panelu przedniego w taki sposób, aby były od siebie jak najbardziej oddalone. Uzyskano w ten sposób maksymalne pole widzenia. Należy zwrócić uwagę, by zachować wymagany dystans czujnika GP2S60 od podłoża. Mieści się on w zakresie od 0.5mm do 1mm.

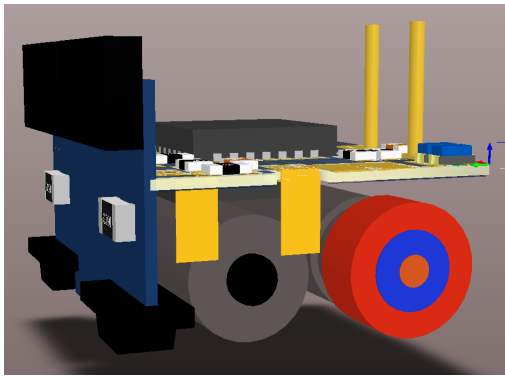
W robocie Leo czujniki linii spełniają również funkcję interfejsu komunikacyjnego UART. Interfejs UART umożliwia programowanie pamięci mikrokontrolera z wykorzystaniem bootloadera oraz debugowanie programu. Emiter lewego czujnika



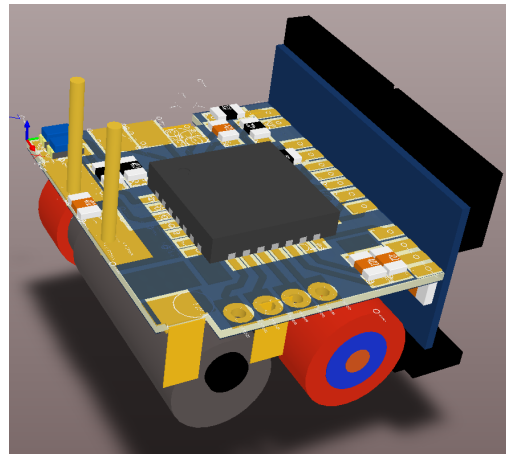
(a) Widok z góry



(b) Widok od spodu

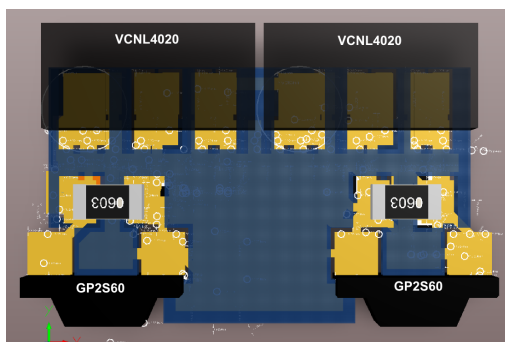


(c) Widok z lewej

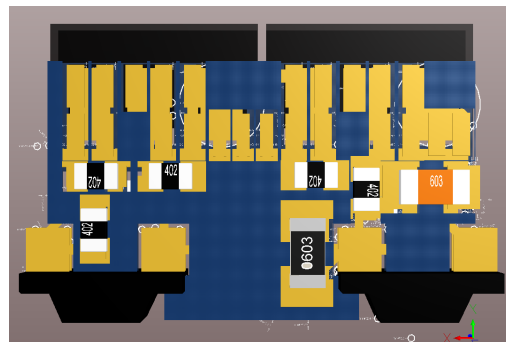


(d) Widok z prawej

Rysunek 1. Model 3D robota Leo

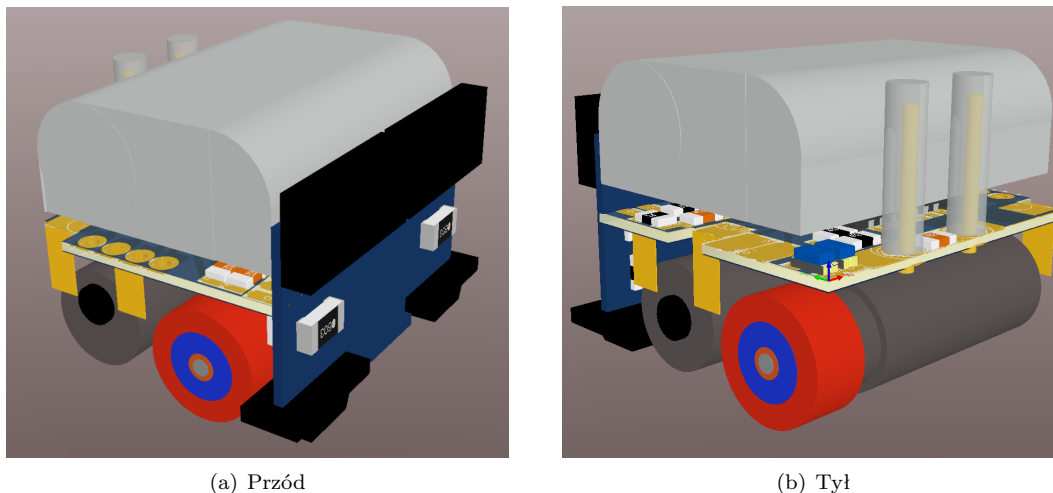


(a) Przód

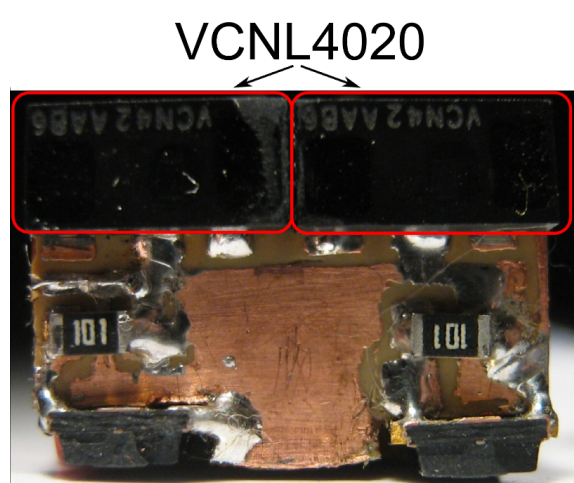


(b) Tył

Rysunek 2. Panel przedni robota Leo



Rysunek 3. Robot Leo wraz z akumulatorem



Rysunek 4. Dalmierze robota Leo

został podłączony do linii TXD, zaś detektor prawego czujnika do linii RXD. Takie rozwiązanie pozwala na programowanie mikrokontrolera z prędkością 2400bsp, a debugowanie z prędkością 9600bps.

2.3. Układ napędowy

Robot Leo posiada kołowy układ napędowy. Jego podstawowymi elementami są dwa silikonowe koła oraz dwa silniki pochodzące z wibracji telefonu komórkowego iPhone5. Rysunek 8 przedstawia podwozie robota Leo. Jak widać długość modułu podwozia nie przekracza 10mm.

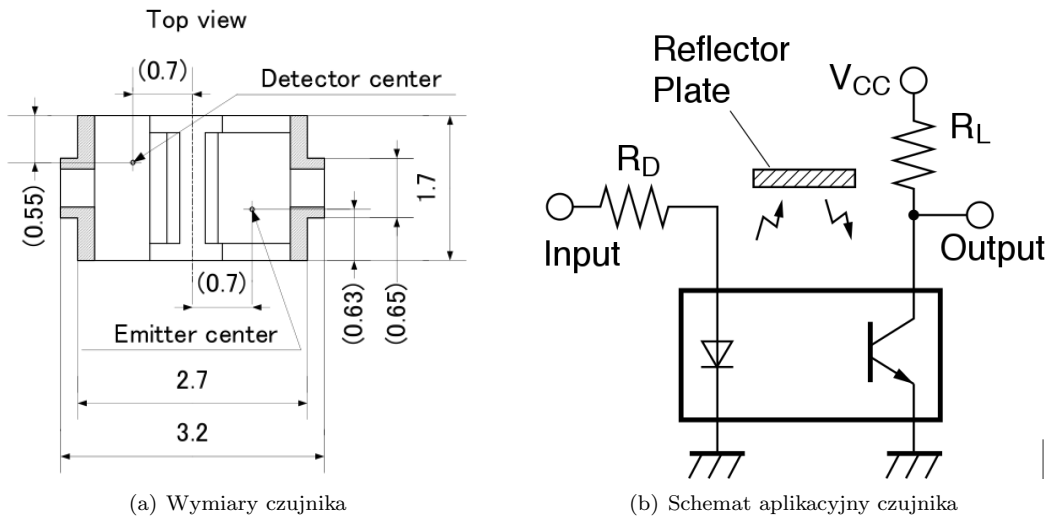
Każde koło robota składa się z trzech warstw izolacji pochodzących z przewodów elektrycznych. Rysunek 7 przedstawia model koła. Na oś silnika średnicy 0.4mm, nałożona została izolacja o średnicy 0.55mm, następnie izolacja o średnicy 2mm. Zewnętrzną warstwę tworzy miękka, silikonowa izolacja. Takie rozwiązanie powoduje, że koło bardzo dobrze trzyma się osi silnika, jest miękkie i zapewnia dobrą przyczepność na większości powierzchni.

Silniki pochodzące z wibracji telefonu iPhone5 charakteryzują się wysoką jakością wykonania oraz wystarczającą mocą. Do tego można je nabyć w wieku sklepach internetowych za przystępną cenę. Nowe silniki mają zgrzany metalowy element montażowy oraz ciężarek wibracyjny. Element montażowy można usunąć poprzez zeszlifowanie zgrzewów. Ciężarek wibracyjny również należy zeszlifować, uważając, by nie uszkodzić osi silnika.

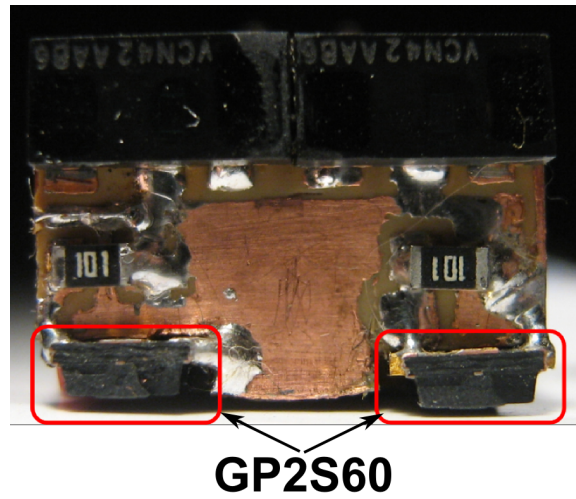
Silniki sterowane są za pomocą dwóch mostków H zawartych w scalonym układzie A3901 firmy Allegro MicroSystems [22]. Układ mieści się w obudowie DFN-10 o wymiarach 3 x 3 x 0.75mm. Maksymalny prąd wyjściowy każdego z mostków wynosi 400mA i jest wystarczający, ponieważ prąd zwarcia zastosowanych silników przy napięciu 4.2V osiąga 80mA.

2.4. Sekcja zasilania

Głównym elementem sekcji zasilania robota Leo jest ogniwo Li-pol GMB LP301009 [15], które bez trudu można nabyć w sklepach internetowych. Na sekcję zasilania składa się również układ zabezpieczenia akumulatora przed nadmiernym rozła-



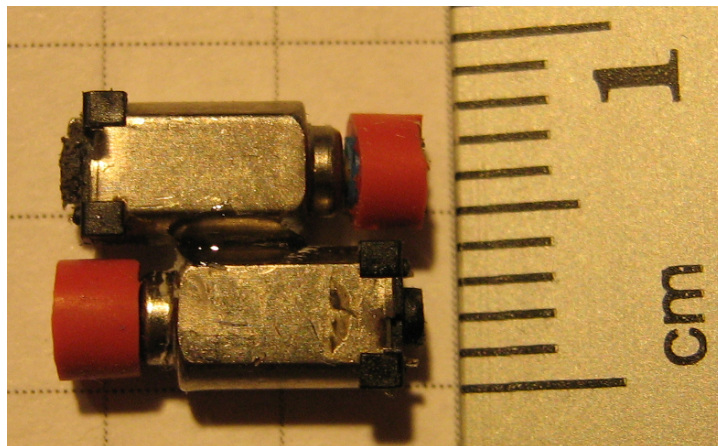
Rysunek 5. Czujnik GP2S60 [21]



Rysunek 6. Czujniki linii robota Leo



Rysunek 7. Koło robota Leo



Rysunek 8. Podwozie robota Leo

dowaniem. Pojedyncze ogniwo GMB LP301009 zapewnia wystarczającą do pracy robota wydajność prądową oraz pojemność pozwalającą na przeprowadzenie kilku starć z przeciwnikiem.

Akumulator jest jedynym rozłącznym elementem konstrukcji. Podłączenie akumulatora jest równoznaczne z włączeniem robota. Takie rozwiązanie eliminuje potrzebę montowania włącznika, jak również ułatwia serwis robota, ponieważ po wyjęciu akumulatora użytkownik ma dostęp do płyty głównej. Co więcej, dzięki takiemu rozwiązaniu, aby naładować akumulator można go wyjąć z robota i podłączyć do ładowarki ogniw Li-pol. Należy pamiętać, że maksymalny prąd ładowania ogniwa GMB LP301009 wynosi 80mA.

Układ zabezpieczający ogniwo Li-pol w robocie Leo jest maksymalnie uproszczony i składa się wyłącznie z dzielnika napięcia. Jednostka centralna kontroluje wartość napięcia akumulatora. Gdy napięcie zasilania spadnie poniżej 3.4V, mikrokontroler sygnalizuje rozładowanie akumulatora poprzez migotanie niebieskiej diody z częstotliwością około 3Hz.

2.5. Jednostka centralna

Jednostka centralna robota Leo w całości znajduje się na płycie głównej pokazanej na rysunku 9. Oparta jest o mikrokontroler ATMEGA88PA [11]. Układ ten oferuje wystarczającą liczbę wyprowadzeń. Posiada timery z funkcją PWM, które są wykorzystywane do sterowania silnikami. ATMEGA88PA zawiera również 10-bitowy, 6-kanalowy przetwornik ADC, interfejs UART, możliwość programowania z wykorzystaniem bootloadera oraz interfejs I²C, którego nie wykorzystano w robocie Leo. Mikrokontroler ATMEGA88PA jest dostępny w większości sklepów z elementami elektronicznymi.

Sterowanie silnikami odbywa się z wykorzystaniem funkcji PWM dwóch timerów: TIMER0 oraz TIMER1. Wybór tych właśnie timerów jest podyktowany ułożeniem pinów wyjściowych z funkcją PWM. Oba kanały PWM skonfigurowano do pracy 8-bitowej, w trybie fast-PWM, z częstotliwością 125kHz. W robocie Leo użyteczny zakres wypełnienia sygnału PWM to 5–30%. Dla większych wartości wypełnienia przyspieszenie robota jest zbyt duże, by w kontrolowany sposób poruszać się po dohyo.

Przetwornik ADC mikrokontrolera wykorzystywany jest do pomiaru napięcia zasilania oraz napięcia wyjściowego czujników linii. Wszystkie pomiary odbywają się w trybie pomiaru pojedynczego z 8-bitową rozdzielczością i częstotliwością zegara przetwornika 125kHz. Źródłem napięcia odniesienia jest napięcie zasilania robota.

Interfejs UART w robocie Leo wykorzystywany jest w dwóch celach. Służy on do przesyłania informacji o stanie czujników do komputera PC z prędkością 9600bps. Interfejs UART pełni również rolę interfejsu programowania pamięci FLASH z wykorzystaniem sekcji bootloader. Podczas programowania stosuje się niższą prędkość przesyłania danych 2400bps, by uniknąć błędów podczas komunikacji.

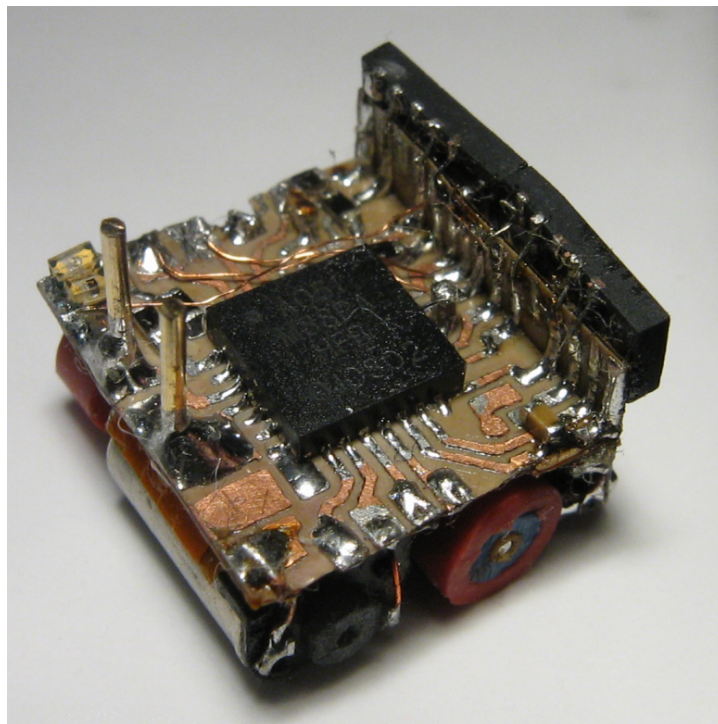
W celu sygnalizowania stanu robota na płycie głównej umieszczono dwie diody LED koloru niebieskiego oraz zielonego. Stan diod można dowolnie zmieniać podczas działania programu.

3. Oprogramowanie

Algorytm pracy robota Leo jest bardzo zbliżony do prostych algorytmów wykorzystywanych w robotach SUMO. Obsługa peryferiów takich jak timery, przetwornik ADC oraz diody LED również jest standardowa i nie wymaga omówienia. W niekonwencjonalny sposób mikrokontroler komunikuje się z czujnikami VCNL4020. Innowacyjną częścią oprogramowania jest sekcja bootloadera i sposób programowania pamięci Leo.

Czujniki VCNL4020 posiadają niezmienny adres *slave* 26_n. Interfejs I²C nie przewiduje możliwości podłączenia więcej niż jednego układu o tym samym adresie *slave*. Z tego powodu programowo zaimplementowano dwa interfejsy I²C. Implementację oparto o kod programu ze strony [23]. Na listingu 1 przedstawiono sposób generowania sygnału start na pierwszej magistrali I²C.

Do programowania pamięci FLASH robota Leo zastosowano bootloader wykorzystujący interfejs UART. Sekcja bootloadera zajmuje 1kB pamięci. Interfejs UART posiada dwie linie danych, w robocie obie zostały podłączone do czujników



Rysunek 9. Płyta główna robota Leo

linii. Dioda IR lewego czujnika sterowana jest linią TXD, zaś informacja odebrana prawym czujnikiem linii skierowana została na linię RXD interfejsu UART. Aby zaprogramować robota należy położyć go na specjalnie przygotowanej do tego celu podstawce, pokazanej na rysunku 10. Na podstawce, w miejscu gdzie należy ułożyć robota, umieszczone zostały dwa czujniki KTIR0711S [1]. Do czujników podłączono konwerter napięć umożliwiający podłączenie przejściówki USB-UART opartej o układ scalony CP2102 [24]. Dla ułatwienia programowania robota podstawka zawiera również złącze zasilania, które można podłączyć do robota w miejsce akumulatora. Gdy na komputerze zostanie uruchomione oprogramowanie do obsługi bootloadera, za pomocą umieszczonego na podstawce włącznika zasilania należy włączyć robota, a bootloader sam zadba o wgranie kodu do pamięci FLASH mikrokontrolera. W ten sposób otrzymano przejściówkę do programowania robota, która nie wymaga dodatkowego złącza do programowania. Czas potrzebny na zaprogramowanie 2kB programu to około 8 sekund.

Listing 1. Funkcja generująca sygnał start na magistrali I²C

```
void i2c_Start1(void)
{
    SDA_OUT1; // ustawienie linii SDA w tryb wyjściowy
    SCL_OUT1; // ustawienie linii SCL w tryb wyjściowy
    SET_SDA1; // ustawienie na linii SDA stanu wysokiego
    SET_SCL1; // ustawienie na linii SCL stanu wysokiego
    delay(); // opóźnienie
    CLR_SDA1;
    delay();
    CLR_SCL1;
}
```

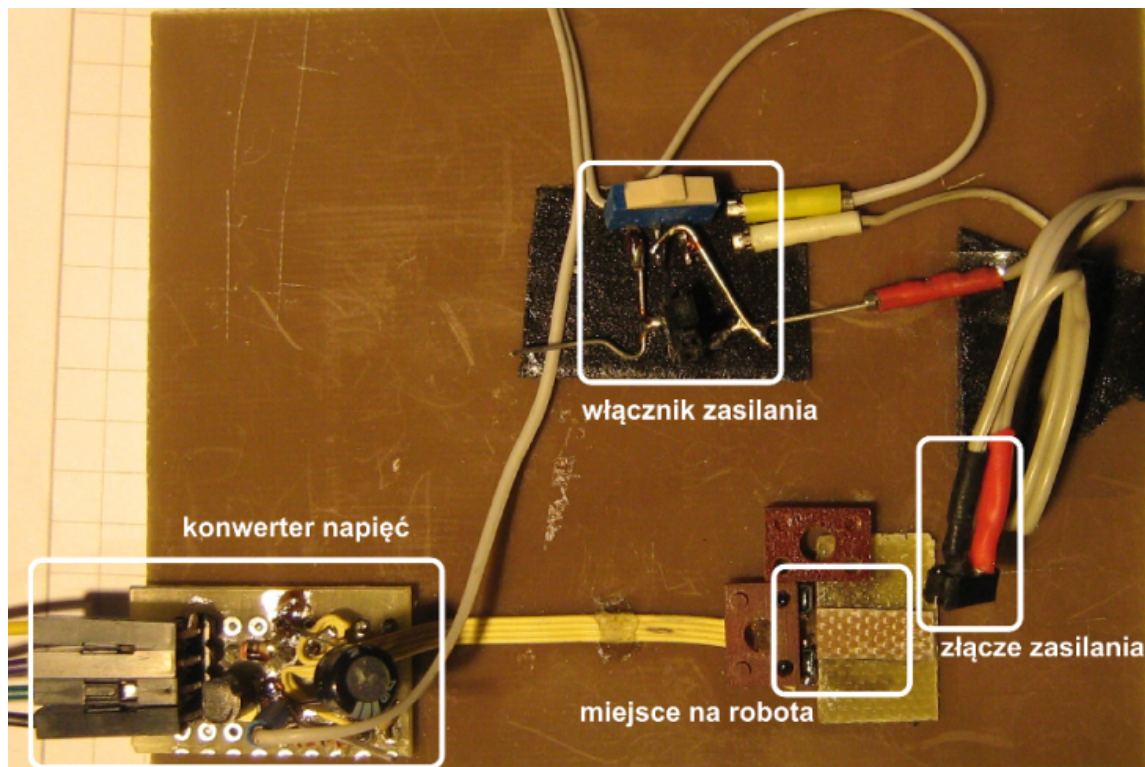
4. Wnioski

Zbudowanie robota klasy femtoSUMO jest trudne, aczkolwiek nie wymaga ono specjalistycznego zaplecza laboratoryjnego ani dużego nakładu finansowego. Niezbędne okazują się jednak cierpliwość i precyzja.

Dzięki uprzejmości Karola Kapera przeprowadzono kilka walk robota Leo z robotem Stuart. Robotowi Leo udawało się realizować cel starcia, jakim jest wypchnięcie przeciwnika z ringu, czasami jednak rywal okazywał się być bardziej skuteczny. Podczas starć żaden z robotów nie był zdecydowanym faworytem, a wynikiem końcowym był remis.

Skonstruowany robot Leo zgodnie z założeniami mieści się w sześcianie o krawędzi 10mm, a jego masa nie przekracza 3g. Robot w sposób autonomiczny walczy na ringu.

Mimo niewielkich wymiarów konstrukcji zdołano wyposażyć robota w dwa dalmierze oraz dwa czujniki białej linii. Zasięg dalmierzy jest uzależniony od faktury oraz koloru obudowy przeciwnika. Zdarza się, że Leo nie jest w stanie zauważyć rywala z odległości promienia ringu. To zagadnienie należy w przyszłości dopracować.



Rysunek 10. Podstawka do programowania

W pracy czujników linii nie stwierdzono żadnych problemów. Wykrywają one białą linię przy różnych, nawet niekorzystnych warunkach oświetlenia. Sensory udało się umieścić na panelu przednim, są więc blisko przedniej krawędzi robota.

Energia elektryczna, jaką gromadzi ogniwo GMB LP301009-10C, wystarcza na przeprowadzenie kilku pojedynków z przeciwnikiem. Czas pracy uzależniony jest od zachowania robota. Nie zauważono jednak, by Leo przegrał walkę z powodu kończącego się zapasu energii elektrycznej.

Układ jezdny robota Leo zapewnia mu moc do przepychania robota Stuart. Stuart waży około 2g, co stanowi $\frac{2}{3}$ maksymalnej masy robota femtoSUMO. Trudno określić, w jaki sposób robot Leo zachowałby się z przeciwnikiem o maksymalnej wadze.

Pamięć mikrokontrolera zgodnie z założeniami może być wielokrotnie programowana. Operacja ta, podobnie jak ładowanie akumulatora, nie wymaga podłączania żadnego przewodu do robota. Niektóre z elementów robota spełniają więcej niż jedną funkcję, na przykład czujniki GP2S60 wykorzystywane są przy programowaniu pamięci mikrokontrolera oraz do komunikacji z komputerem PC, a terminal akumulatora służy jako włącznik zasilania. Robot Leo został zbudowany z elementów, które można nabyć w Polsce w przystępnych cenach. Dobrym, źródłem elementów do budowy robota były urządzenia mobilne, na przykład telefony komórkowe.

W ocenie autora opracowana konstrukcja stwarza ciekawe możliwości w zakresie dalszych prac, nie tylko w kontekście walk SUMO. W ramach jej rozwoju należałoby w pierwszej kolejności zwiększyć zasięg widzialności dalmierzy. Kolejnym krokiem może być podłączenie do robota modułu radiowego do zdalnej komunikacji. Takie rozwiązanie umożliwi, na przykład zbudowanie roju małych robotów, które będą razem współpracować w celu zrealizowania jakiegoś zadania. Może to być przesunięcie pewnego elementu, którego masa przekracza wielokrotnie masę każdego z robotów.

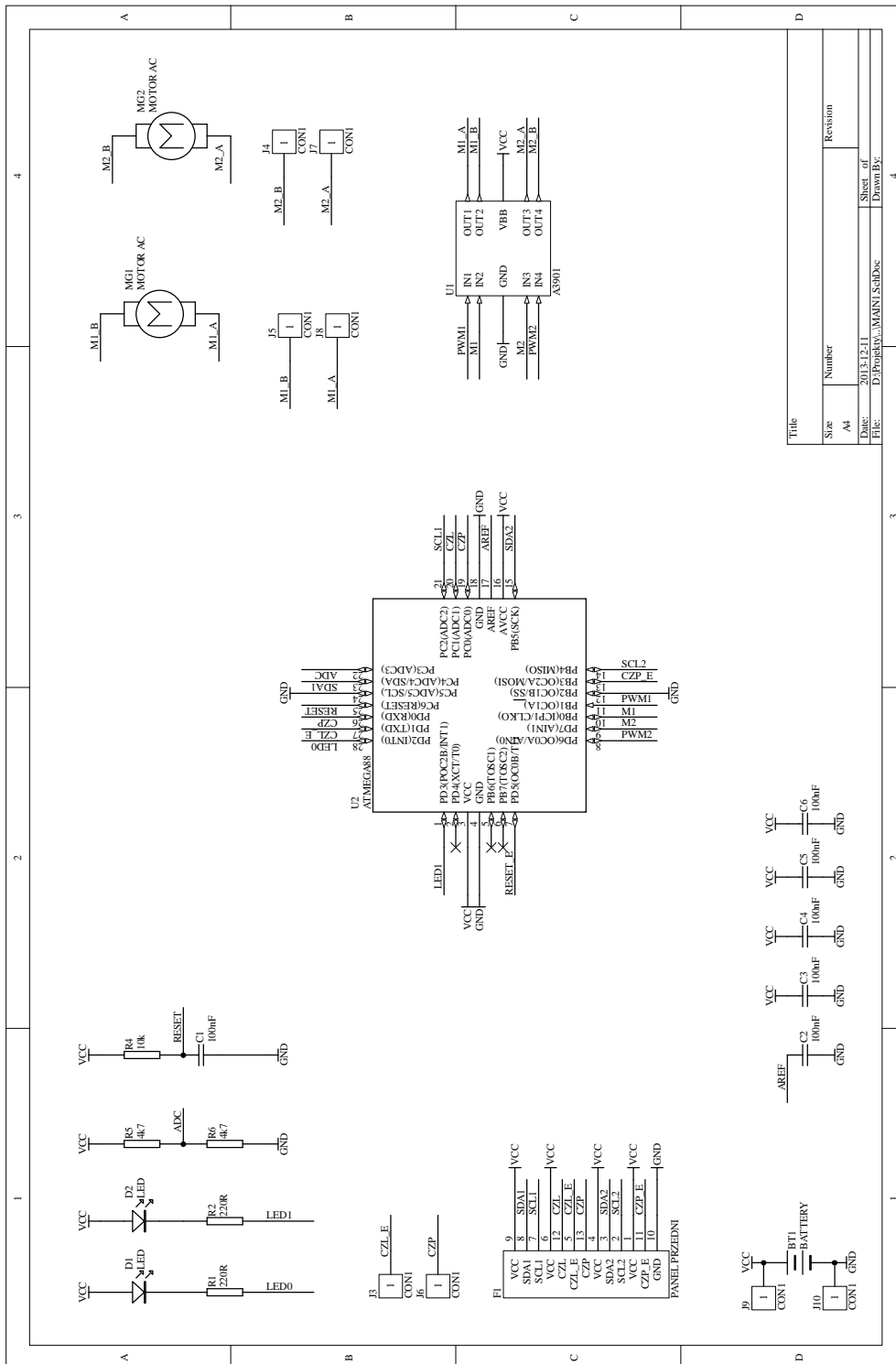
A. Projekty płyt PCB

Dodatek zawiera schematy ideowe oraz projekty obwodów drukowanych robota Leo.

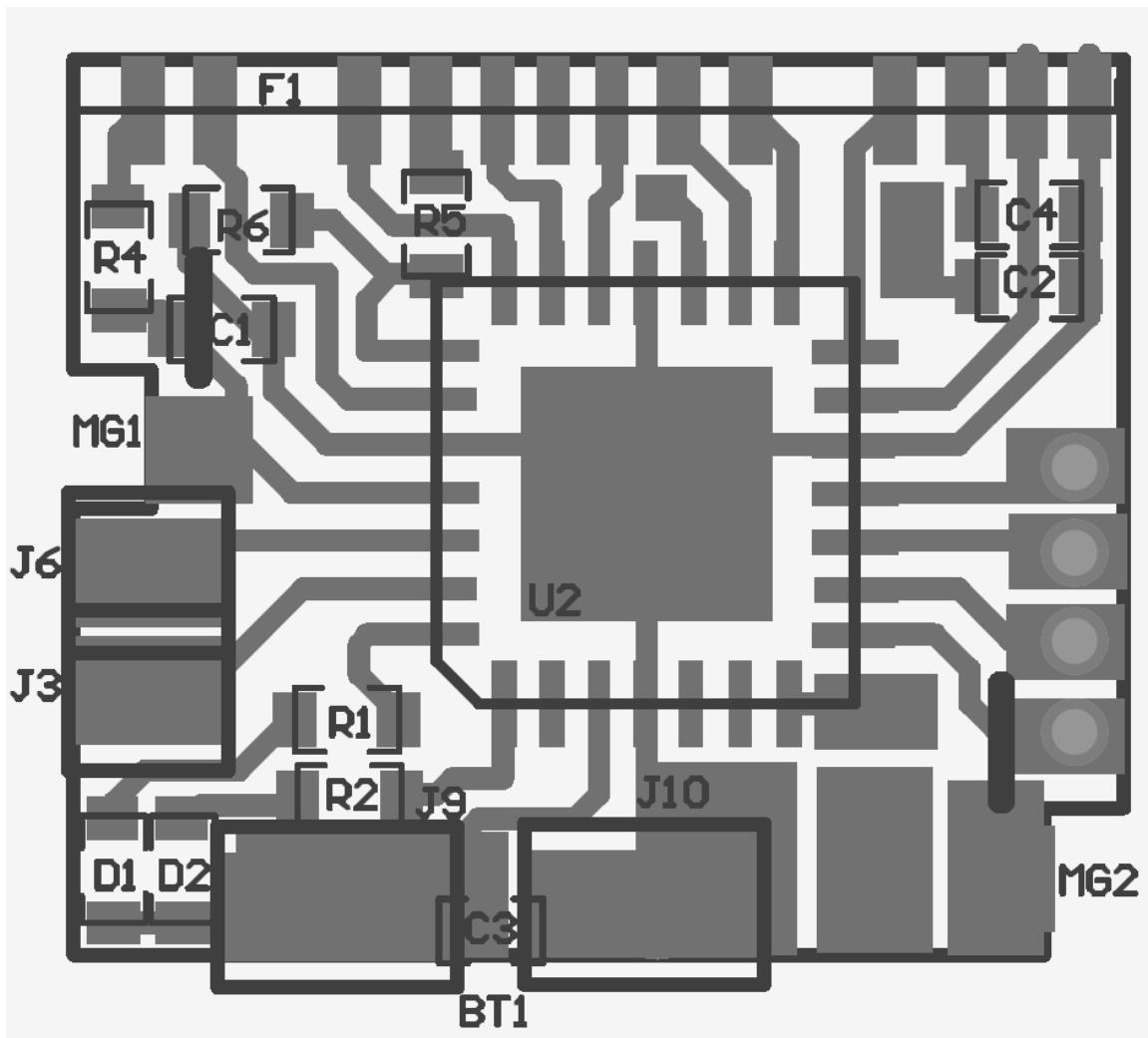
1. Schemat ideowy płyty głównej rysunek 11
2. Projekt obwodu drukowanego płyty głównej, widok z góry rysunek 12
3. Projekt obwodu drukowanego płyty głównej, widok od spodu rysunek 13
4. Schemat ideowy panelu przedniego rysunek 14
5. Projekt obwodu drukowanego panelu przedniego, widok z góry rysunek 15
6. Projekt obwodu drukowanego panelu przedniego, widok od spodu rysunek 16

B. Kosztorys

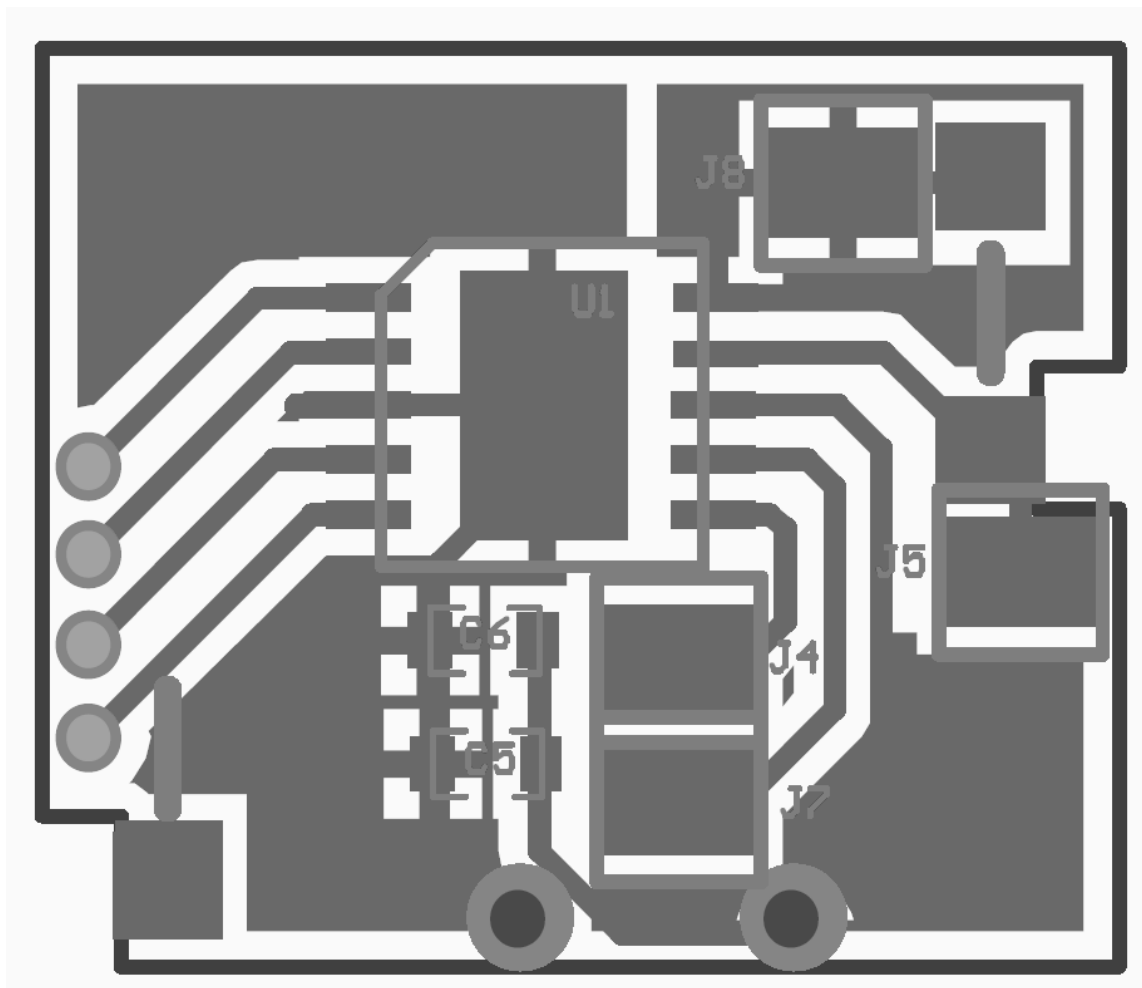
Dodatek zawiera orientacyjny kosztorys podzespołów wykorzystanych do budowy robota Leo.



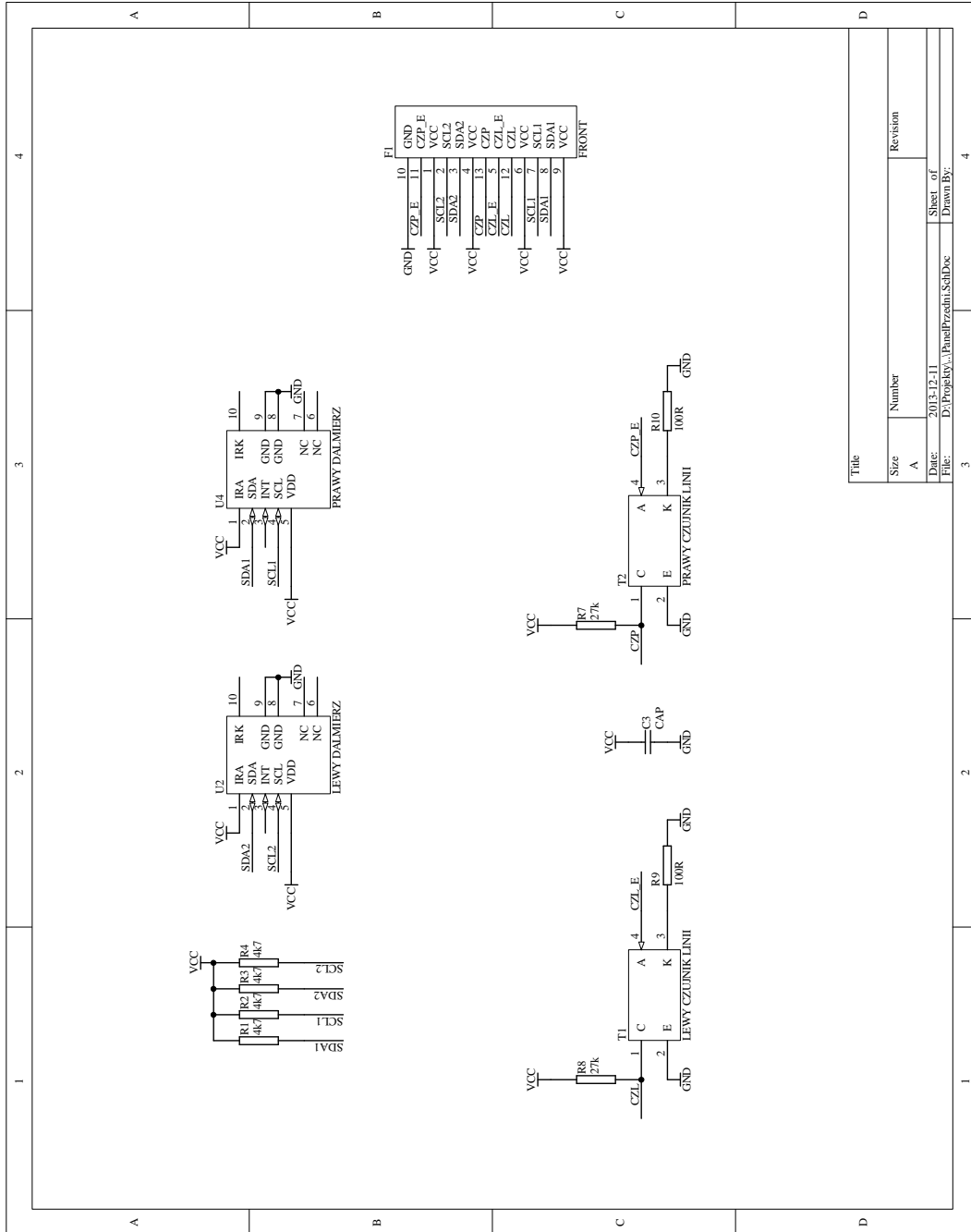
Rysunek 11. Schemat ideowy płyty głównej



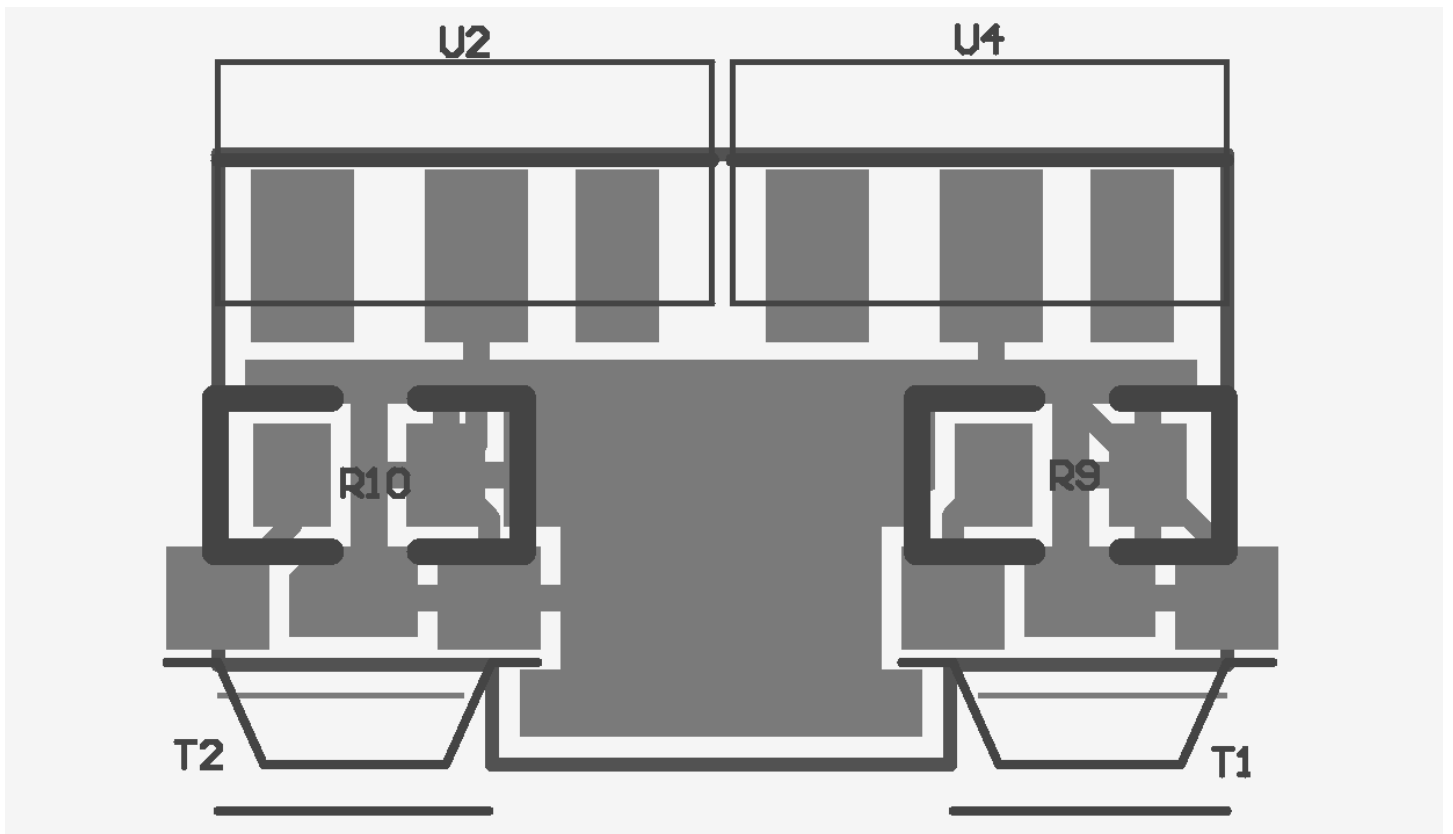
Rysunek 12. Projekt obwodu drukowanego płyty głównej, widok z gry



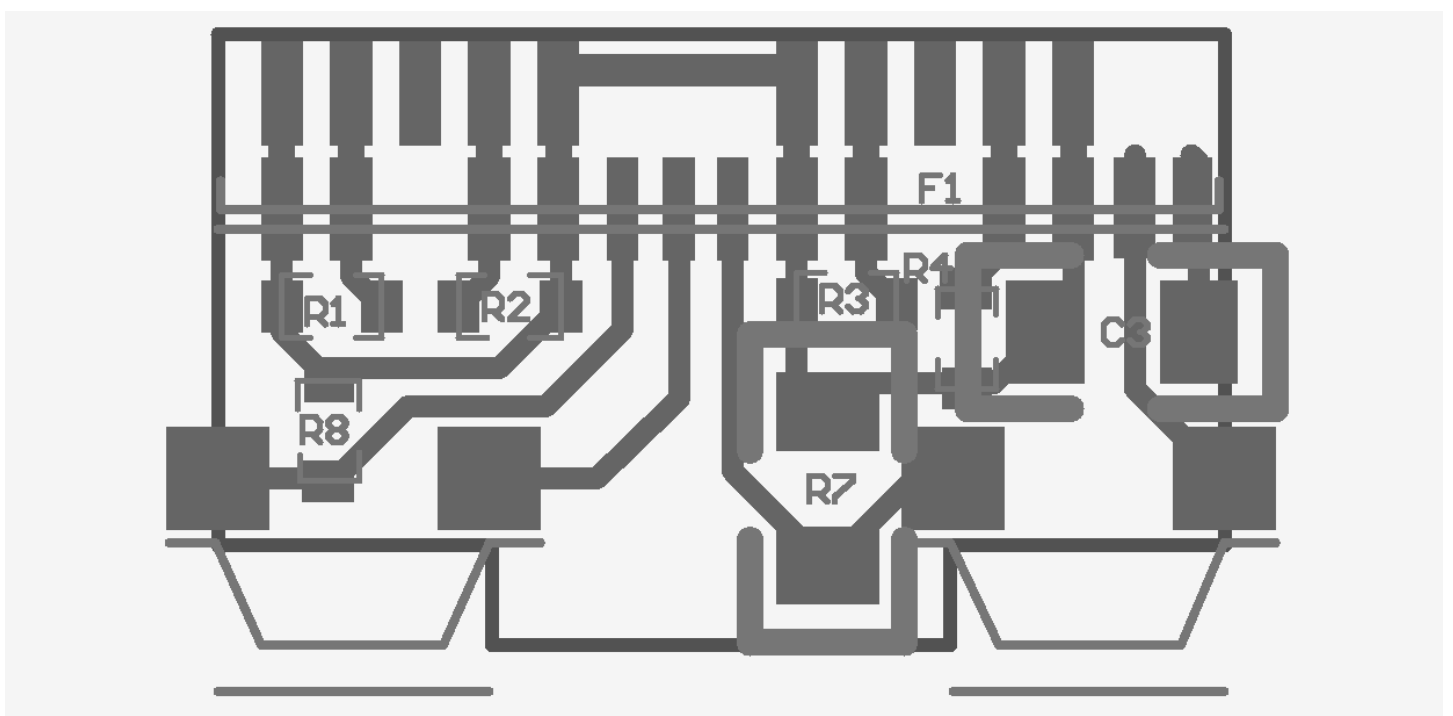
Rysunek 13. Projekt obwodu drukowanego płyty głównej, widok od spodu



Rysunek 14. Schemat ideowy panelu przedniego



Rysunek 15. Projekt obwodu drukowanego panelu przedniego, widok z góry



Rysunek 16. Projekt obwodu drukowanego panelu przedniego, widok od spodu

• czujnik VCNL4020	2szt.	–	20zł
• czujnik GP2S60	2szt.	–	8zł
• silnik iPhone5	2szt.	–	20zł
• akumulator GMB LP301009	1szt.	–	35zł
• mikrokontroler ATMEGA88PA-MMH	1szt.	–	10zł
• sterownik silników A3901	1szt.	–	5zł
• dioda LED	2szt.	–	2zł
• elementy bierne		–	10zł
	SUMA		110zł

Literatura

- [1] <http://www.aeontekasia.com/pdf/KTIR0711S%20%20KTIR0721DS.pdf>
- [2] <http://www.vishay.com/docs/83760/tcrt5000.pdf>
- [3] <http://electropark.pl/72-czujniki-optyczne>
- [4] <http://www.konar.pwr.wroc.pl/sites/default/files/uploaded/RaportBL.pdf>
- [5] Komputer Świat Ekspert - maj 2007r. strona 13.
- [6] http://pl.wikipedia.org/wiki/Akumulator_litowo-polimerowy
- [7] <http://www.konar.pwr.wroc.pl/pl/zrob-sam/zabezpieczenie-akumulatora-li-poly>
- [8] <http://kmfeniks.wordpress.com/category/porady/>
- [9] <http://www.stm32java.com/portal/>
- [10] http://jv.wikipedia.org/wiki/Gambar:Mikon_ATmega_128_SMD.jpg
- [11] <http://www.atmel.com/devices/atmega88pa.aspx>
- [12] http://www.beam-wiki.org/wiki/Robot_Sumo
- [13] <http://robotbox.net/project/cakemonster/femto-sumo>
- [14] http://www.lunacommerce.com/search.php?orderby=position&orderway=desc&search_query=vibrator
- [15] http://www.bto.pl/B2CProdukt.aspx?id_artykulu=39040
- [16] <http://www.digikey.com/product-detail/en/VCNL4000-GS08/751-1505-1-ND/2526871>
- [17] <http://botland.com.pl/czujniki-koloru/962-sfh7773-czujnik-odleglosci-i-natezenia-swiatla-otoczenia-.html>
- [18] <http://www.futureelectronics.com/en/technologies/semiconductors/optoelectronics/optical-sensors-switches/Pages/8025877-VCNL4020-GS08.aspx>
- [19] <http://il.rsdelivers.com/product/sharp/gp2s60/photointerrupter-reflec-sdd-05mm-10ctr/6675041.aspx>
- [20] <http://pl.wikipedia.org/wiki/I%C2%B2C>
- [21] http://sharp-world.com/products/device/lineup/data/pdf/datasheet/gp2s60_e.pdf
- [22] <http://www.allegromicro.com/en/Products/Motor-Driver-And-Interface-ICs/Bipolar-Stepper-Motor-Drivers/A3901.aspx>
- [23] <http://www.elektroda.pl/rtvforum/topic670366.html>
- [24] <http://www.silabs.com/Support%20Documents/TechnicalDocs/CP2102-9.pdf>

Spis rysunków

1	Model 3D robota Leo	2
2	Panel przedni robota Leo	2
3	Robot Leo wraz z akumulatorem	3
4	Dalmierze robota Leo	3
5	Czujnik GP2S60 [21]	4
6	Czujniki linii robota Leo	4
7	Koło robota Leo	4
8	Podwozie robota Leo	5
9	Płyta główna robota Leo	6
10	Podstawka do programowania	7
11	Schemat ideowy płyty głównej	8
12	Projekt obwodu drukowanego płyty głównej, widok z gry	9
13	Projekt obwodu drukowanego płyty głównej, widok od spodu	10
14	Schemat ideowy panelu przedniego	11
15	Projekt obwodu drukowanego panelu przedniego, widok z góry	12
16	Projekt obwodu drukowanego panelu przedniego, widok od spodu	12

Spis tabel

1	Wybrane rejestry konfiguracyjne układu VCNL4020	1
---	---	---