

Robot Conan



Głównym powodem budowy prezentowanego robota była konieczność realizacji projektu na studiach. Miał on polegać na stworzeniu układu elektronicznego, który „coś” robi. Stwierdziłem, że dobrze byłoby połączyć elektronikę z mechaniką, bo wtedy można otrzymać interesujący efekt końcowy. I tak po rozważeniu wielu za i przeciw, wybór padł na „ramię” – tak na początku nazywany był robot.

Zaprojektowanie i wykonanie części od zera byłoby dość kosztowne, więc jedynym rozsądnym rozwiązaniem było wykorzystanie tego, co zwykle zbywa nam w warsztacie. Postanowiłem użyć części pochodzących ze starych napędów FDD, CDROM-ów, jakiś zabawek wyposażonych w koła zębate, z których za pomocą szlifierki i wiertarki można zrobić ciekawe elementy składowe.

Ogólna zasada działania

Robot jest zasilany napięciem 5 V. Zasilają go serwomechanizmy umożliwiające ruch poszczególnych elementów (zapewniające tzw. stopnie swobody) oraz mikrokontroler.

Układ elektroniczny jest bardzo prosty. Oprócz mikrokontrolera (AT90S2313) jest wyposażony w moduł Bluetooth BTM112, zasilany napięciem 3,3 V uzyskiwanym za pomocą dodatkowego stabilizatora. Moduł BTM112 odbiera polecenia przesyłane przez komputer PC, przekazuje je mikrokontrolerowi, który steruje pracą serwomechanizmów. Polecenia sterujące mogą być też przesyłane przez inne urządzenie, jak np. telefon komórkowy wyposażony w interfejs Bluetooth oraz odpowiednie oprogramowanie realizujące funkcję panelu kontrolnego.

Jak wspomniano wcześniej, w tym projekcie użyto komputera PC. Pracuje on pod kontrolą systemu Linux Gentoo. Do komputera podłączono zewnętrzny moduł Bluetooth oraz joystick. W momencie naciśnięcia przycisku lub ruchu joysticka, aplikacja sterująca (napisana w języku Python) wysyła odpowiednie komendy przez Bluetooth do robota, gdzie zostają one odebrane i zamienione na ruch poszczególnych serwomechanizmów.

Budowa

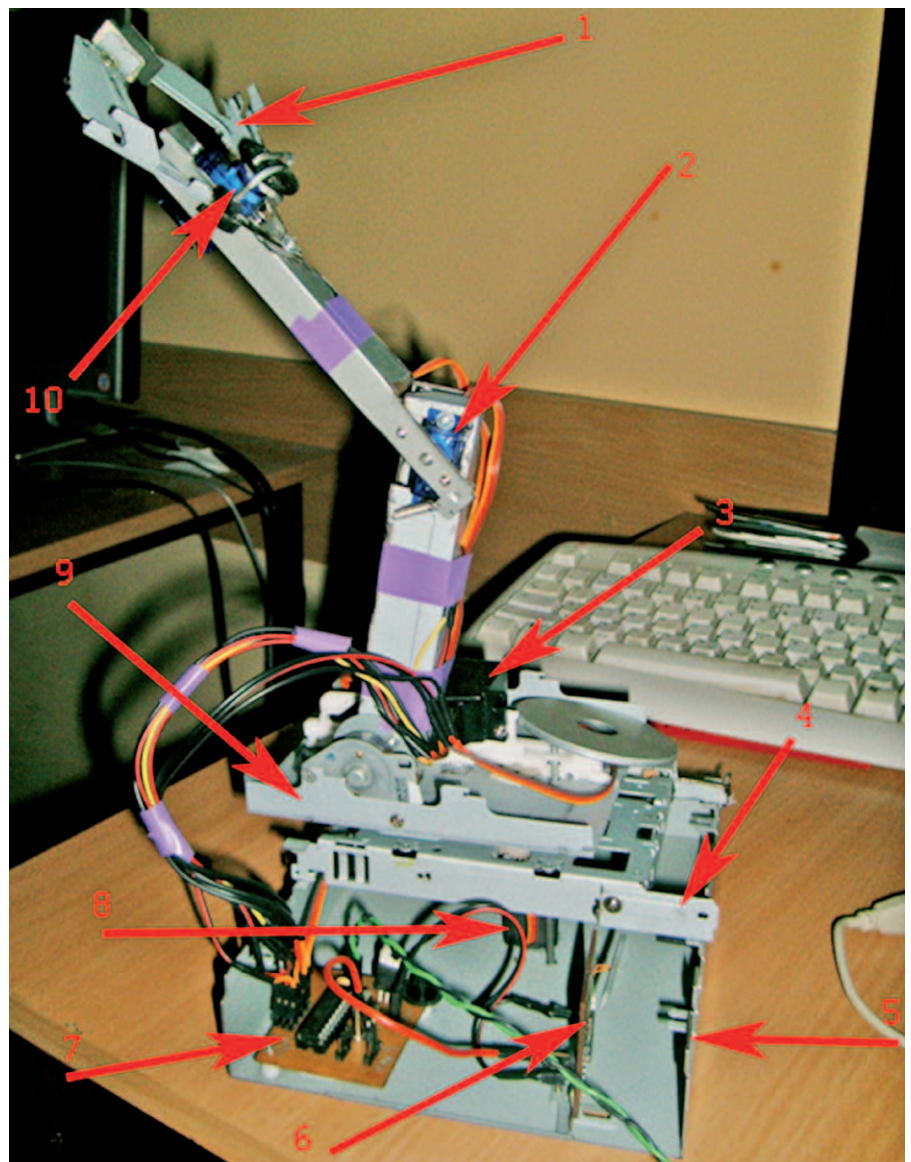
Wygląd robota pokazano na fotografii 1. Głównym elementem jest podstawa wykonana z części obudowy starego napędu CD ROM. Na niej są przymocowane pozostałe elementy. Do spodu podstawy są przyklejone gumowe nóżki, zapewniające stabilność konstrukcji. Na podstawie jest przymocowana płytki z mikrokontrolerem,

moduł Bluetooth oraz serwomechanizm odpowiedzialny za obrót ramienia w płaszczyźnie poziomej.

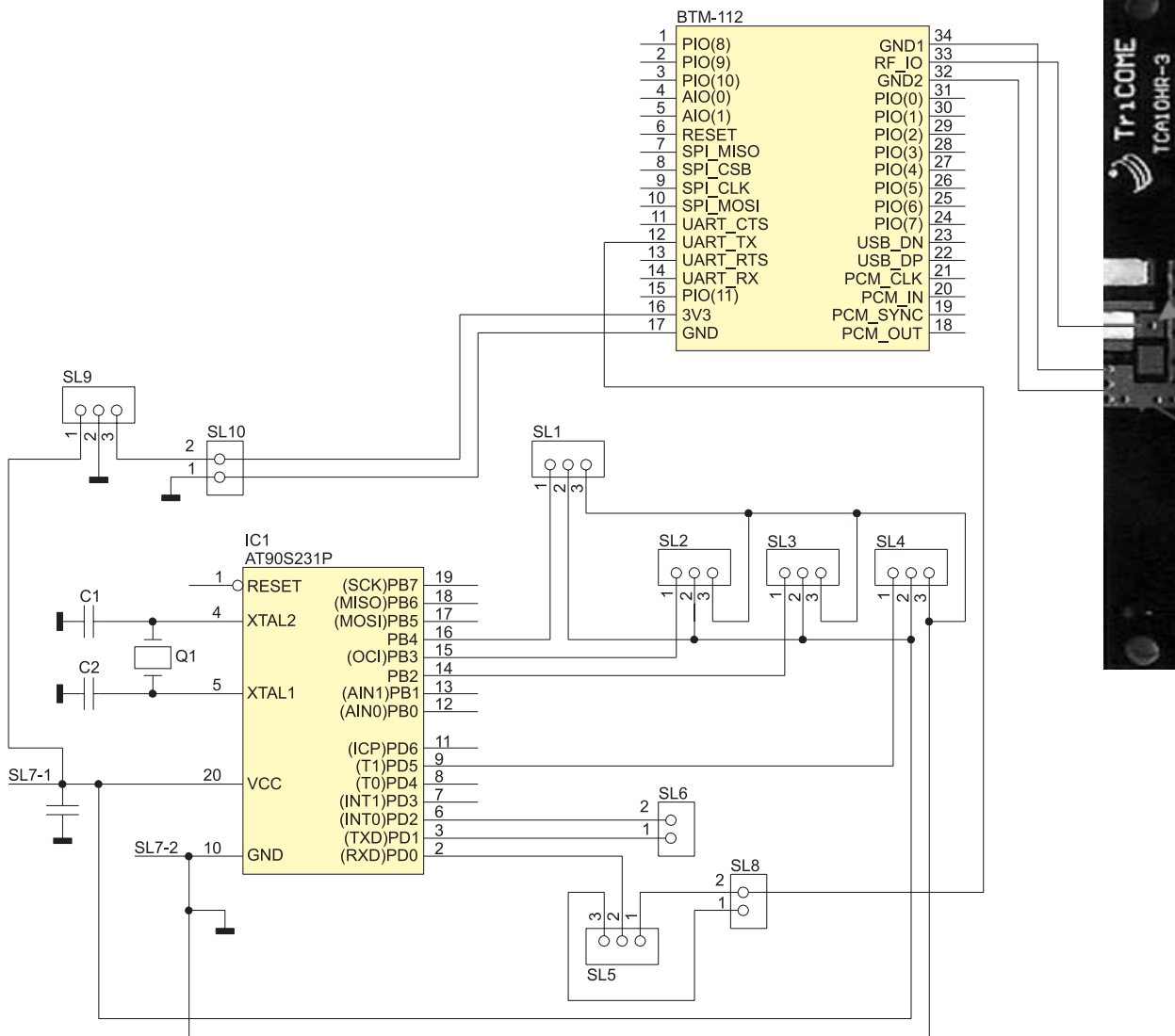
Serwomechanizm, za pomocą przekładni zębatej (o przełożeniu dającym obrót

o 300°) jest połączony z wałem obracającym ramię. Ów wał jest osadzony na platformie przytwierdzonej do podstawy. Drugi koniec wału jest przymocowany do obrotowej platformy.

Na obrotowej platformie jest serwomechanizm odpowiedzialny za kolejny stopień swobody, a mianowicie ruchy w przód i w tył. Do niego za pomocą orczyków jest przymocowany pierwszy człon ramienia, które wykonano z dwóch skrzyżowanych ze sobą kątowników. Na końcu tego członu jest serwomechanizm poruszający kolejnym



Fotografia 1. Robot CONAN. Legenda: 1 – chwytak, 2 – serwomechanizm TowerPro SG90 odpowiedzialny za ruch „w górę/w dół”, 3 – serwomechanizm TowerPro SG5010 odpowiedzialny za ruch „w przód/w tył”, 4 – platforma przymocowana do podstawy, 5 – podstawa, 6 – płytki z modułem Bluetooth i anteną, 7 – płytki z mikrokontrolerem ATtiny2313, 8 – serwomechanizm TowerPro SG5010 odpowiedzialny za obrót, 9 – platforma obrotowa, 10 – serwomechanizm TowerPro SG90 odpowiedzialny za ruch chwytaka



Rysunek 2. Schemat układu. SL1, SL2, SL3, SL4 – złącza serwomechanizmów; SL5 – złącze umożliwiające ustawianie źródła sterowania (Bluetooth lub port szeregowy); SL8, SL6 – złącza UART; SL7 – zasilanie układu; SL9 – stabilizator napięcia 3,3 V; SL10 – złącze zasilające moduł BTM112; C1 – kondensator 22 pF; C2 – kondensator 22 pF; C3 – kondensator 100 µF; Q1 – kwarc 8 MHz; IC1 – mikrokontroler ATtiny 2313; BTM 112 – moduł Bluetooth BTM112; ANTENA – antena Bluetooth

członem ramienia wykonanym z pojedynczego kątownika. Zwieńczeniem wszystkim jest serwomechanizm umożliwiający poruszanie chwytakiem. Poszczególne elementy są opisane na fotografii 1.

Elektronika

Sterowanie serwami umożliwia układ stworzony przy użyciu mikrokontrolera ATtiny2313. Jego schemat umieszczono na rysunku 2, a wygląd prostej płytki drukowanej na rysunku 3. Serwa są sterowane za pomocą sygnałów PWM, a komunikacja z Bluetooth odbywa się przy użyciu wyjścia UART pracującego z następującymi ustawieniami: 19200 bps, 8 bitów danych, parzystość – none, bit stopu – 1.

Wybrana została komunikacja przy użyciu Bluetooth ponieważ jest ona stosunkowo prosta w obsłudze, a w sklepach można kupić sporo różnych modułów, które wystarczy zamontować, zasilić i działają bez konieczności konfigurowania. Po-

nadto, dzięki powszechności interfejsów Bluetooth instalowanych w komputerach i telefonach oraz dostępności narzędzi softwara'owych, łatwo można napisać oprogramowanie sterujące.

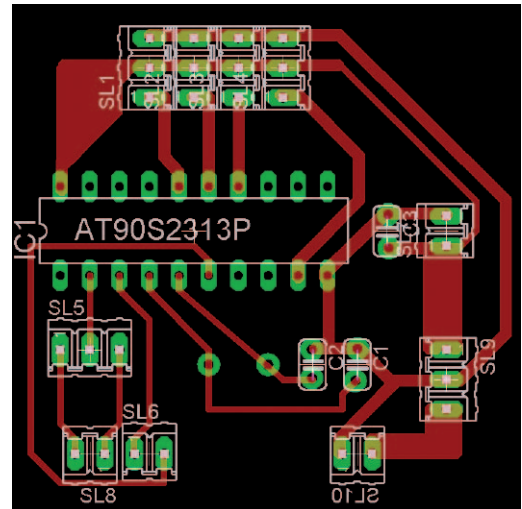
Aplikacja sterująca

Program sterujący robotem został napisany w języku Python wyposażonym w następujące biblioteki:

- pygame (odpowiedzialna za obsługę joysticka),
- serial (za komunikację przez port szeregowy),
- gtk (do stworzenia GUI- interfejsu graficznego).

Python jest językiem interpretowanym, obiektowym, wysokiego poziomu, umożliwiającym szybkie i proste tworzenie aplikacji. Dla potrzeb robota interesuje nas obsługa Joysticka i komunikacja przez port

szeregowy (z modułem Bluetooth). Poniższy zestaw instrukcji pokazuje jak prosto można wykonać pewne czynności:



Rys. 3. Rozmieszczenie elementów na płytce drukowanej

```

- self.myserial = serial.Serial("/dev/
rfcomm2", 19200, timeout=None,
parity=serial.PARITY_NONE,
stopbits=serial.STOPBITS_ONE,
bytesize=serial.EIGHTBITS)

```

Utworzenie obiektu odpowiedzialnego za komunikację przez port szeregowy, nadanie odpowiednich wartości parametrom połączenia, takim jak bit parzystości, prędkości transmisji itd.

```

- pygame.joystick.init()

```

inicjalizacja obsługi Joysticka, funkcja skanuje system w poszukiwaniu urządzenia

```

- self.stick = pygame.joystick.Joystick(0)

```

tworzy nowy obiekt typu Joystick skojarzony z fizycznym urządzeniem

```

- pygame.display.init()

```

inicjalizacja modułu pygame.display

```

- self.stick.init()

```

inicjalizacja obiektu typu Joystick w celu dostępu do jego funkcji

```

- event = pygame.event.wait()

```

zwraca zdarzenie w tym wypadku spowodowane przez joystick

```

- if event.button==7: self.myserial.
write('u')

```

jeżeli wciśnięty był przycisk numer 7 to wyślij znak 'u' przez port szeregowy

Bluetooth konfigurujemy jako port szeregowy. W systemie Linux Gentoo tę czynność należy wykonać w następujący sposób.

Znajdujemy plik (/etc/bluetooth/rfcomm.conf) i tworzymy w nim taki wpis:

```

rfcommX {
    # Automatically bind the
device at startup
    (wzorymy węzeł urządzenia,
/dev/rfcomm0 na początku)
    bind yes;
    # Bluetooth address of the
device
    (Należy wpisać adres
urządzenia do, którego chcemy się
łączyć)
    device adres;
}

```

Gdzie X to dowolna liczba nie wykorzystywana przez inne urządzenia rfcomm, natomiast adres to MAC urządzenia Bluetooth zamontowanego w robocie.

Kolejnym krokiem jest odłączenie kabla łączącego moduł Bluetooth z mikrokontrolerem i podłączenie zasilania do robota. Teraz, jako superuser, należy wykonać następujące polecenie: `rfcomm connect X` (gdzie X to ta sama liczba, którą wpisaliśmy w pliku rfcomm.conf). W tym momencie komputer powinien nawiązać połączenie z modułem i powinno pojawić się okienko z możliwością wpisania kodu PIN - wpisujemy „1234”, domyślny pin urządzeń typu BTM-112. Jeżeli w konsoli pojawi się komunikat „Connected”, to oznacza, że połączenie przebiegło pomyślnie, zostało utworzone urządzenie /dev/rfcommX, przez które będzie możliwa komunikacja. Pozostaje tylko uruchomić aplikację w Pythonie i... Dobrej zabawy!

Podsumowanie

Jak widać budowa robotów jest dziedziną interdyscyplinarną, która łączy w sobie elementy mechaniki, elektroniki i informatyki. Dodatkowo, zaprezentowane, proste urządzenie jest doskonałym przykładem jak można zrobić „coś” z „niczego”.

W przypadku budowy profesjonalnych robotów tego typu najpierw należałoby wykonać projekt, dokonać odpowiednich obliczeń. Pewnie i ja gdybym takie wykonał, to nie musiałbym używać sprężyny wspomagającej poruszanie ramieniem, ponieważ zastosowany przeze mnie serwo-mechanizm był zbyt słaby. To jest jednak dopuszczalne, jeśli buduje się coś, co bardziej służy zabawie i edukacji aniżeli zastosowaniom profesjonalnym.

Piotr Zagnański

Zestawy CNC

CNC-Z0.38

3x Silnik krokowy 42BYGH101B

Hybrydowe silniki serii 42BYGH charakteryzują się stosunkowo dużym momentem w porównaniu do ich rozmiarów. Znajdują zastosowanie tam, gdzie od dynamiki ważniejsza jest pewność zachowania dużej siły przy małej wykorzystywanej przestrzeni.

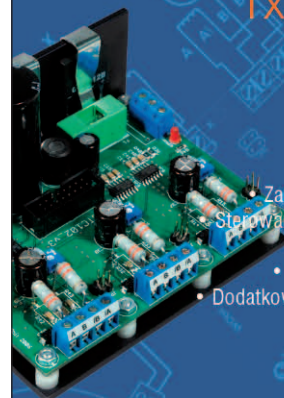
Podstawowe dane:

- Dwufazowy bipolarny
- Prąd 1.7 A
- Moment 0,38 Nm
- Podział kroku 1.8°
- Wym. 42x42x39 mm
- Waga 240 g



1x Sterownik STC102 WP

- Prąd 2.5A na fazę z możliwością płynnej regulacji
- Podział kroku do 1/8
- Zakres napięcia 10 – 20 VAC
- Sterowanie trzech niezależnych osi
- Zintegrowany zasilacz
- Interfejs KROK-KIERUNEK
- Dodatkowe optoizolowane sygnały 3xHOME + STOP



1x Transformator TS TZS150/20



już od 498 zł netto

ZADZWOŃ

61 2912 225

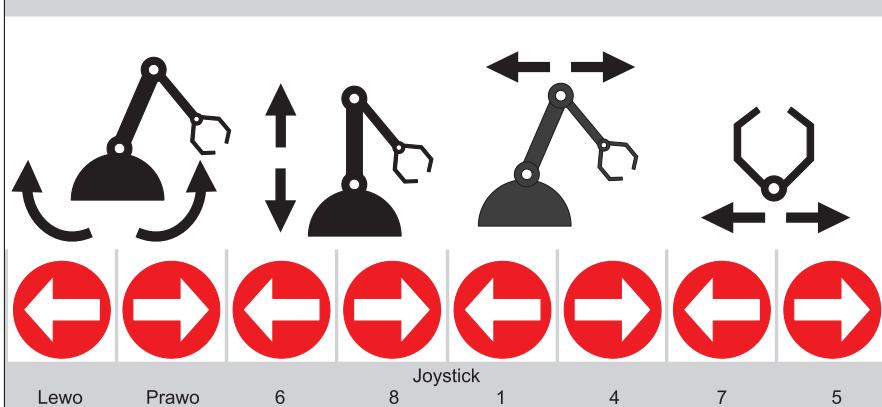
lub

napisz do nas

wobit@wobit.com.pl

www.wobit.com.pl

MAIN CONTROL PANEL



Rysunek 4. Okno aplikacji sterującej