

# Podwieszany ploter z Raspberry Pi

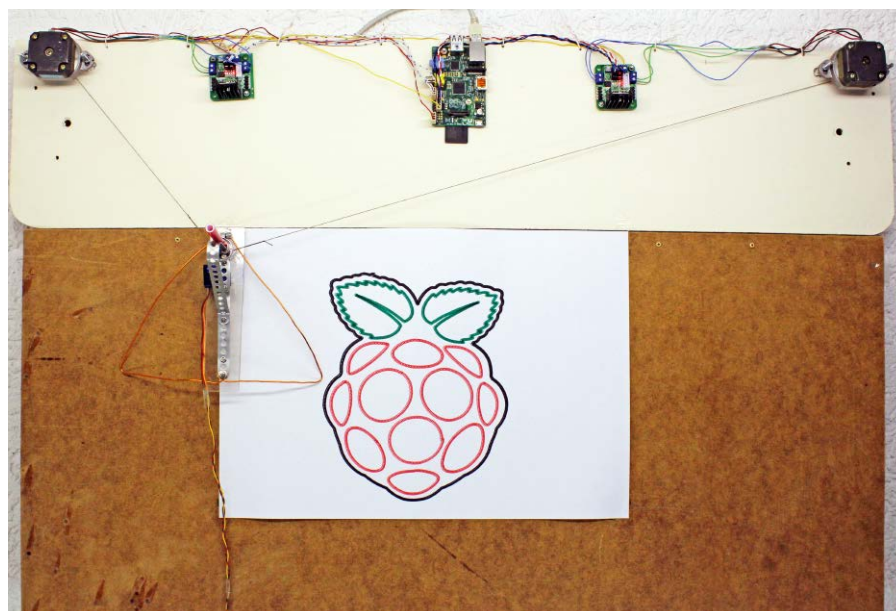
**Wydruk dużych schematów i grafik wektorowych nierzadko stanowi problem. Zazwyczaj stosuje się w tym celu plotery, których koszt zakupu wcale nie jest mały. Ploter można też zbudować samodzielnie, względnie niewielkim kosztem, np. w oparciu o Raspberry Pi. Co prawda nie będzie on miał precyzji urządzeń profesjonalnych, a tempo pracy będzie raczej powolne, ale po udoskonaleniu może okazać się przydatnym narzędziem. W artykule prezentujemy ploter tego typu, opracowany przez Norberta Heinza z Niemiec.**

Autor projektu od lat tworzy maszyny CNC i podobne urządzenia własnej konstrukcji. Jego projekty oparte są na długoletnim doświadczeniu oraz służą do opracowywania kolejnych urządzeń. V plotter jest jedną z nowszych maszyn i wykorzystuje komputer Raspberry Pi. Ponadto, całość, choć ma względnie niewielkie wymiary, pozwala na rysowanie na teoretycznie dowolnie dużych powierzchniach. Wynika to z faktu, że głowica plotera wisi na dwóch linkach i rysuje po kartkach umieszczonych pionowo, na tablicy ze sklejk lub drewna. Wygląd tego nietypowego urządzenia pokazano na **fotografii 1**.

## Elektryka i mechanika

Od strony elektrycznej, V plotter składa się z jednego Raspberry Pi, dwóch silników krokowych oraz sterujących nimi dwóch podwójnych modułów z mostkiem H, a także z dodatkowego, małego serwonapędu, sterowanego bezpośrednio z GPIO komputerka. Sposób wykonania połączeń elektrycznych został przedstawiony na **rysunku 2**.

Silniki krokowe są przymocowane do kół zębatach, które służą jako przekładnie. Koła przytwierdzone są do tablicy z użyciem łożysk kulkowych. Wewnętrzny pierścień łożyska nie obraca się i sztywno trzyma tablicy. Zewnętrzny jest przyklejony do koła zębatego i obraca się razem z nim. Na zewnętrzny pierścień łożyska jednego koła zębatego przy prawym silniku i odpowiadającego mu łożyska koła przy lewym silniku nawinięte są linki. Drugi koniec linek jest przymocowany do głowicy drukującej. Głowica zawiera pisak oraz lekki serwonapęd, podłączony luźno zwisającym przewodem do GPIO Raspberry Pi. Pisak jest zamontowany za pomocą uchwytu, do metalowej, perforowanej blaszki. Blaszka przyczepiona jest w jednym punkcie do akrylowej płytki. Miniaturowy serwonapęd jest zamontowany również na blaszce, ale w taki sposób, by jego ruch powodował wyginanie się metalu i odchylenie pisaka od akrylowej płytki. Otwór w płytce sprawia, że jeśli pisak jest do niej przysunięty, w praktyce dotyka końcówką kartki i może po niej rysować.



Fotografia 1. Gotowy ploter

## Dodatkowe informacje

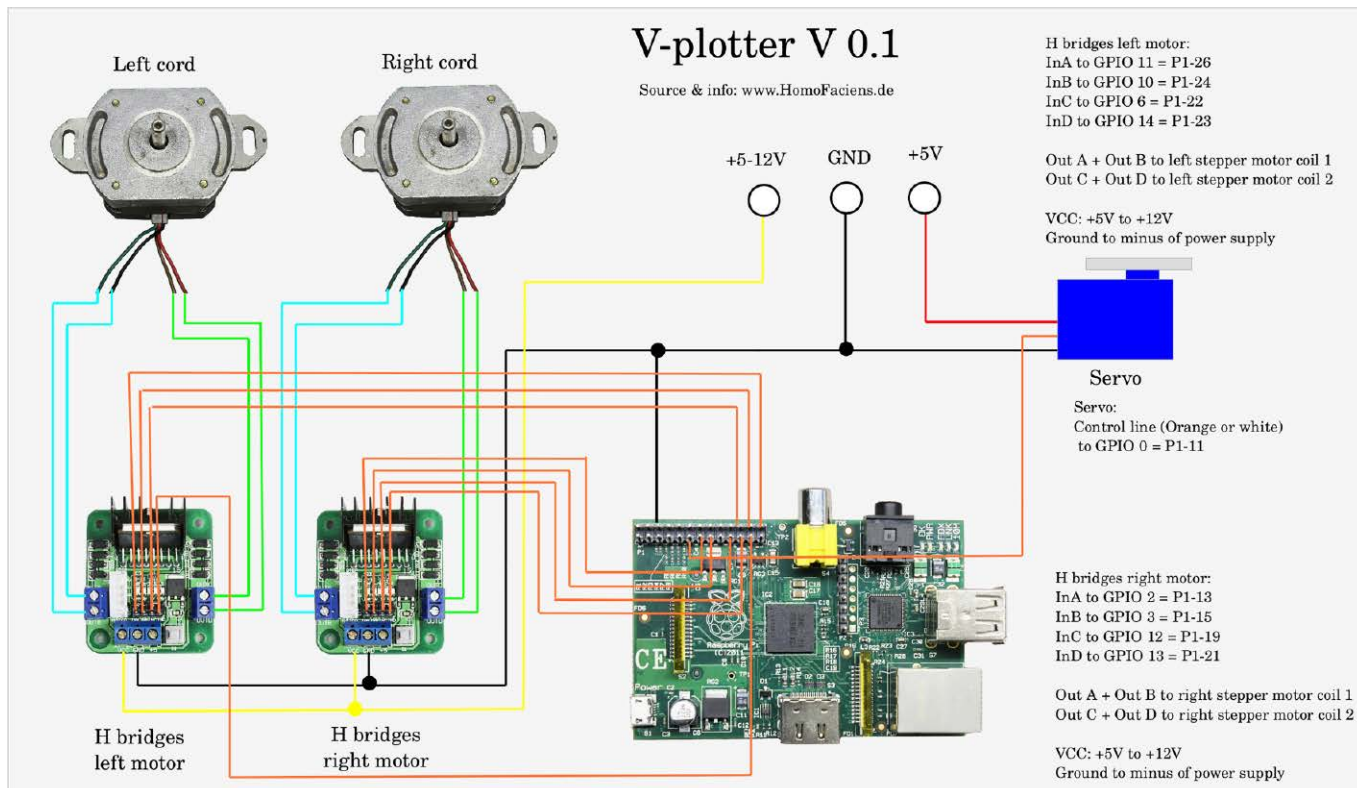
Opis całego projektu można znaleźć pod adresem <http://goo.gl/BmiFyT>, kod źródłowy programu dostępny jest na stronie <http://goo.gl/GAQgyl>, a dodatkowo autor opublikował film na YouTube, gdzie przedstawia działanie i budowę plotera – można go obejrzeć pod adresem: <https://goo.gl/iRHqom>.



Wygięcie serwonapędem blaszki odsuwa pisak od kartki, co pozwala na przesuwanie głowicy po papierze bez pozostawiania śladu.

Kluczowe było użycie odpowiednio małego serwonapędu, by nie uczynić głowicy zbyt ciężką. Ważne jest też by sensownie poprowadzić przewody, tak by nie płały się. Użycie płytki akrylowej pozwala też stabilniej trzymać pisak względem kartki. Dzięki niej pisak nie obraca się w żadnej płaszczyźnie, ani nie buja – tarcie kartki o płytkę szybko tłumi taki ruch.

Sam komputer jest umieszczony w górnej części plotera i przymocowany do tablicy, mniej więcej na wysokości silników krokowych. Moduły mostków umieszczone są pomiędzy Raspberry Pi a silnikami. Nawijanie i rozwijanie linek za pomocą silników powoduje ruch głowicy względem kartki. Wykorzystywana jest przy tym grawitacja – Równomierne i równoczesne rozwijanie linek powoduje opuszczanie głowicy pionowo w dół, a ich nawijanie – ruch do góry. Jeśli rozwijana jest tylko jedna linka, głowica opuszcza się po łuku okręgu o promieniu równym aktualnej długości drugiej linki i środki w punkcie drugiej osi. Oczywiście, osie kół na które nawijane są linki muszą



Rysunek 2. Schemat połączeń elektrycznych

znajdować się na skrajach kartki, gdyż ograniczają obszar wydruku od lewej i prawej.

### Matematyka

Obliczenie tego, jak przekłada się pozycja silników na pozycję pisaka wymaga zastosowania wielu obliczeń matematycznych. Po pierwsze, należy obliczyć, jak przekłada się długość rozwiniętych linek na pozycję względem kartki. Sumaryczna długość rozwiniętych linek nie może być mniejsza od odległości pomiędzy osiami kół, na których linki są nawijane. Jeśli długości rozwiniętych linek są sobie równe, głowica jest umieszczona po środku, pomiędzy osiami, a pozycja w pionie zależy od sumarycznej długości rozwiniętych linek.

Budując samodzielnie ploter tego typu trzeba dokładnie określić odległość pomiędzy serwonapędami (a dokładniej, pomiędzy punktami, z których wyprowadzone są linki – warto je precyzyjnie ustalić, np. poprzez

użycie dodatkowych, małych, metalowych oczek). Autor stworzył oprogramowanie w taki sposób, że po zamontowaniu plotera ustala się punkt startowy głowicy, będący lewym, górnym rogiem zadrukowywanej kartki. Współrzędne tego punktu, wyrażone w milimetrach, wraz z odległością pomiędzy punktami wyprowadzeń linek, należy podać jako parametry programu. Konieczne jest też sprawdzenie, jak przy zastosowanych kołach zębatych i silnikach krokowych, jeden krok przekłada się na ruch głowicy. W tym celu użytkownik musi najpierw zmierzyć odległość w linii prostej głowicy w pozycji startowej od punktu wyprowadzenia jednego z silników, a następnie wykonać nim 2000 kroków. Gdy głowica osiągnie nową pozycję, następuje kolejny pomiar jej odległości od punktu wyprowadzenia linki tego silnika. Różnica w pomiędzy odległością końcową a początkowo to długość, na jaką wysuwana jest linka po 2000 krokach. Program wymaga

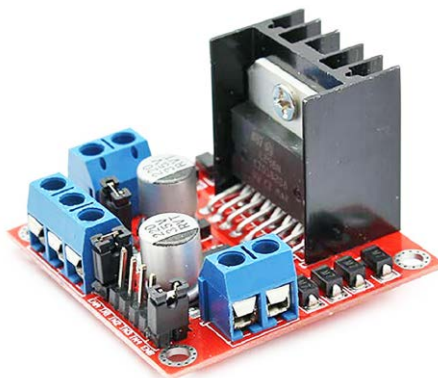
podania ile kroków przypada na wysunięcie linki o 1 mm. Można jeszcze zastanowić się nad zależnością szybkości rozwijania linki od jej aktualnego stopnia nawinięcia – w końcu im warstwa linki na łożysku jest grubsza, tym efektywna średnica szpuli większa, a więc każdy obrót koła powoduje szybsze rozwijanie linki. W praktyce jednak, przy zastosowanych łożyskach, których zewnętrzna średnica to 30 mm, 140-centymetrowa linka nie powoduje znaczącego zwiększenia grubości szpuli nawet przy pełnym nawinięciu i można ten problem pominąć.

Umiejąc przeliczać liczbę przebytych kroków jednego i drugiego silnika na długość rozwijanej linki oraz znając aktualną pozycję głowicy, można obliczać liczby kroków potrzebne do przesunięcia głowicy do nowej pozycji. Trzeba jednak pamiętać, że tor ruchu głowicy pomiędzy dwoma punktami znajdującymi się od siebie w tej samej odległości będzie zależał od ich pozycji względem powierzchni zadrukowywanej. Przykładowo, silniki krokowe, poruszając się jednocześnie, przy równo wysuniętych linkach, będą sprawiały że głowica będzie się poruszała pionowo w górę lub w dół. Natomiast jeśli początkowo linki nie będą w równym stopniu rozwinięte, każdy równocześnie wykonany krok będzie powodował przesunięcie głowicy po ukosie, pod kątem zależnym od aktualnych proporcji długości rozwiniętych linek.

Inaczej mówiąc, ploter powinien otrzymywać dokładnie obliczone pozycje punktów na rysowanych liniach, a nie tylko początek



Fotografia 3. Silnik krokowy Tandon KP4M2-217



Fotografia 4. Moduł mostka H

Listing 1. Fragmenty kodu źródłowego programu

```

// deklaracje stałych, w tym wyprowadzeń GPIO
#define PI 3.1415927
#define SERVOUP 10
#define SERVODOWN 20
#define LEFT_STEPPER01 11
#define LEFT_STEPPER02 10
#define LEFT_STEPPER03 6
#define LEFT_STEPPER04 14
#define RIGHT_STEPPER01 2
#define RIGHT_STEPPER02 3
#define RIGHT_STEPPER03 12
#define RIGHT_STEPPER04 13
#define STEP_PAUSE 1500
#define STEP_MAX 220.0
#define Z_SERVO 0
#define BUFFERSIZE 120
#define BASELENGTH 845
#define CORDLENGTH_LEFT 220
#define CORDLENGTH_RIGHT 667
#define CORDFLEXFACTOR 0.0f

double StepsPermm = 2000.0 / 98.0;
//funkcja wykonująca krok lewym silnikiem. Analogiczna funkcja została przygotowana dla silnika prawego
void MakeStepLeft(int direction){
    StepX += direction;
    if(StepX > 3){
        StepX = 0;
    }
    if(StepX < 0){
        StepX = 3;
    }
    if(StepX == 0){
        digitalWrite(LEFT_STEPPER01, 1);
        usleep(STEP_PAUSE);
        digitalWrite(LEFT_STEPPER02, 0);
        digitalWrite(LEFT_STEPPER03, 0);
        digitalWrite(LEFT_STEPPER04, 0);
    }
    if(StepX == 1){
        digitalWrite(LEFT_STEPPER03, 1);
        usleep(STEP_PAUSE);
        digitalWrite(LEFT_STEPPER01, 0);
        digitalWrite(LEFT_STEPPER02, 0);
        digitalWrite(LEFT_STEPPER04, 0);
    }
    if(StepX == 2){
        digitalWrite(LEFT_STEPPER02, 1);
        usleep(STEP_PAUSE);
        digitalWrite(LEFT_STEPPER01, 0);
        digitalWrite(LEFT_STEPPER03, 0);
        digitalWrite(LEFT_STEPPER04, 0);
    }
    if(StepX == 3){
        digitalWrite(LEFT_STEPPER04, 1);
        usleep(STEP_PAUSE);
        digitalWrite(LEFT_STEPPER01, 0);
        digitalWrite(LEFT_STEPPER02, 0);
        digitalWrite(LEFT_STEPPER03, 0);
    }
    usleep(STEP_PAUSE);
}

//funkcja poruszająca głowicę o odległość X w poziomie i Y w pionie
void moveXY(long X, long Y){
    long newCordLengthLeft;
    long newCordLengthRight;
    double forceLeft, forceRight;
    double deltaCordLeft, deltaCordRight;
    double deltaCordLeft0, deltaCordRight0;
    double alpha;
    char TextLine[1000];
    //uwzględnienie rozciągliwości linek
    alpha = atan((double)(X0) / (double)(Y0));
    forceLeft = 1.0 * cos(alpha);
    forceRight = 1.0 * sin(alpha);
    deltaCordLeft0 = (double)(CORDLENGTH_LEFT * StepsPermm) * forceLeft * CORDFLEXFACTOR;
    deltaCordRight0 = (double)(CORDLENGTH_RIGHT * StepsPermm) * forceRight * CORDFLEXFACTOR;
    //oszacowanie sił działających na linki w danym punkcie
    alpha = atan((double)(X + X0) / (double)(Y + Y0));
    forceLeft = 1.0 * cos(alpha);
    forceRight = 1.0 * sin(alpha);
    X += X0;
    Y += Y0;
    newCordLengthLeft = sqrt((double)(X * X) + (double)(Y * Y));
    newCordLengthRight = sqrt((double)((BaseLength-X) * (BaseLength-X)) + (double)(Y * Y));
    deltaCordLeft = (double)(newCordLengthLeft) * forceLeft * CORDFLEXFACTOR - deltaCordLeft0;
    deltaCordRight = (double)(newCordLengthRight) * forceRight * CORDFLEXFACTOR - deltaCordRight0;
    newCordLengthLeft -= deltaCordLeft;
    newCordLengthRight -= deltaCordRight;
    while(newCordLengthLeft > CordLengthLeft){
        MakeStepLeft(1);
        CordLengthLeft++;
    }
    while(newCordLengthLeft < CordLengthLeft){
        MakeStepLeft(-1);
        CordLengthLeft--;
    }
    while(newCordLengthRight > CordLengthRight){
        MakeStepRight(1);
        CordLengthRight++;
    }
    while(newCordLengthRight < CordLengthRight){
        MakeStepRight(-1);
        CordLengthRight--;
    }
}

```

i koniec linii. Dzięki temu, tor poruszania się głowicy będzie dokładnie taki, jak potrzeba, a jednocześnie sam ruch i tak będzie płynny, gdyż szybkość sterowania silnikami krokowymi z użyciem Raspberry PI jest wystarczająco duża.

### Zasilanie

Zastosowane silniki krokowe to japońskie modele KP4M2-217 firmy Tandon (**fotografia 3**). Pobierają prąd zależny od podłączonego napięcia. Przy 12 V przepływa przez nie 340 mA, co przekłada się na ok. 4 W mocy, a przy 5 V prąd spada do 150 mA, a więc pobierana moc spada do ok. 0,75 W. Użyte moduły z mostkami bazują na układach L298N (**fotografia 4**), a więc scalonych, dwukanałowych sterownikach silników. Ich maksymalny prąd ciągły to 2 A, ale autor twierdzi, że mocno się nagrzewają już przy 1 A.

W efekcie potrzebne jest jedno źródło napięcia do mostków (5...12 V) i jedno 5-voltowe do zasilania Raspberry PI oraz serwonapędu umieszczonego na głowicy. Masa jest wspólna dla wszystkich elementów.

### Oprogramowanie

Autor napisał program w języku C i udostępnił jego kod źródłowy. Program korzysta z biblioteki **wiringPi**, którą można także bezpłatnie pobrać z Internetu i zainstalować. Najlepiej zrobić to z użyciem polecenia git, które w razie potrzeby też trzeba doinstalować. W Raspbianie służą do tego komendy:

```
sudo apt-get update
sudo apt-get install git-core
Pobranie i instalowanie wiringPi poleceniem git odbywa się następująco:
git clone git://git.drogon.net/wiringPi
cd wiringPi
git pull origin
./build
Pobrany kod źródłowy można skompilować za pomocą komendy gcc:
gcc v-plotter.c -o v-plotter -I/usr/local/include -L/usr/local/lib -lwiringPi -lm
a później by go uruchomić, wystarczy wpisać:
./v-plotter
```

Fragmenty kodu źródłowego wraz z komentarzami pokazano na **listingu 1**.

Program pozwala na wczytywanie plików SVG, przy czym rysuje jedynie kontury, a nie wypełnienia. Ponadto, wszystkie obiekty muszą być przekonwertowane na wielokąt.

### Podsumowanie i ocena projektu

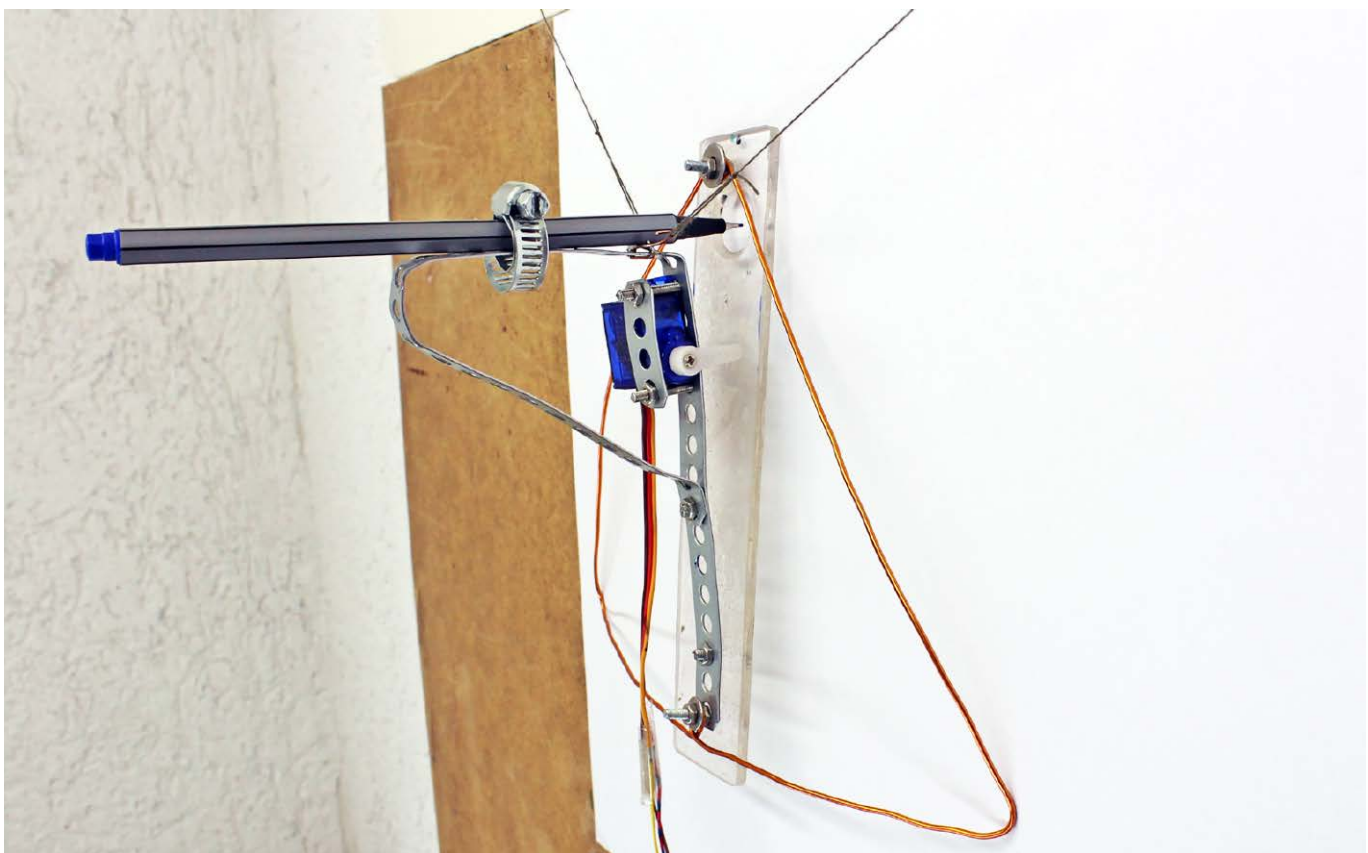
Przygotowany projekt pokazuje, jak łatwo jest wykonać przydatne urządzenie elektromechaniczne z użyciem niedrogo komputera jednopłytkowego. Fakt wykorzystania komputera z systemem operacyjnym, zamiast modułu typu Arduino, które autor wykorzystuje w wielu innych swoich projektach, zdecydowanie ułatwia wykonywanie

wydruków w oparciu o pliki graficzne. Co więcej, użycie Raspberry Pi z bezprzewodowym interfejsem sieciowym sprawia, że ploter pracuje bezprzewodowo – przekazywanie plików do druku może odbywać się przez sieć Wi-Fi, za pomocą SSH.

Warto też zwrócić uwagę na staranność wykonania projektu. Autor, dzięki dużemu doświadczeniu wiedział, co będzie istotne w tego typu konstrukcji. Zastosował łożyska, przekładnie, a linki utrzymujące głowicę wyprowadził z użyciem dodatkowych, miniaturowych przelotek. W programie uwzględniono także fakt rozciągania się linek. Zjawisko to ma różne nasilenie, w zależności od rodzaju użytych linek oraz od pozycji głowicy. Im jest ona wyżej, tym naprężenie linek silniejsze. Niestety, głowicę plotera wykonano „jako tako” – jej wygląd prezentuje **fotografia 5**. Być może autor chciał jak najszybciej zobaczyć efekt swojej pracy, bo tak naprawdę najważniejszym elementem plotera jest układ napędowy i jego oprogramowanie – głowica jedynie trzyma pisak.

Co ciekawe, rzadkością jest też użycie języka C, który w projektach dla Raspberry PI pojawia się rzadko. Autor jednak wykorzystał swoje doświadczenie w tym języku, wynikające z wykonania wielu projektów na Arduino. Sam projekt może też stanowić podstawę do realizacji innych urządzeń, w których pozycja jakiegoś elementu jest zależna od ustawienia silników krokowych.

**Marcin Karbowiczek, EP**



**Fotografia 5. Sposób wykonania głowicy jest najmniej profesjonalny w tym projekcie**