

# Kosmiczna myszka DIY

Wiele rodzajów specjalizowanego oprogramowania, zwłaszcza CAD (Computer Aided Design – projektowanie wspomagane komputerowo), CAM (Computer Aided Manufacturing – komputerowe wspomaganie wytwarzania) czy EDA (Electronic Design Automation – automatyzacja projektowania elektronicznego), może współpracować ze specjalnymi manipulatorami ułatwiającymi obsługę. Specjalne „myszki 3D” pozwalają np. manipulować wirtualnymi przedmiotami w przestrzeni projektu.

Duże pakiety narzędziowe do projektowania urządzeń mechanicznych i elektronicznych oraz inne programy wykorzystujące wirtualną przestrzeń 3D często mogą być obsługiwane za pomocą zwykłej myszki i klawiatury. Jednak znacznie wygodniejsze jest obsługiwanie tych programów za pomocą specjalnych urządzeń wejściowych, które są wyposażone w większą liczbę stopni swobody oraz przyciski do często używanych skrótów czy narzędzi. Na **fotografii 1** pokazano

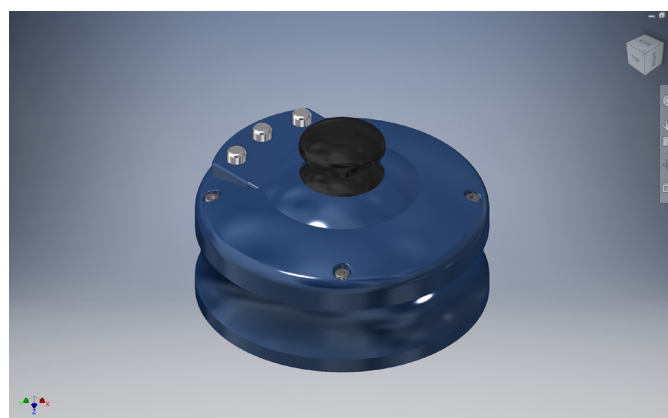
urządzenie firmy 3Dconnexion – SpaceNavigator, który jest wyposażony w aż sześć stopni swobody ruchu manipulatora, co pozwala z łatwością nawigować w trójwymiarowym środowisku pracy.

Urządzenia takie znacznie przyspieszają i ułatwiają prace nad projektami 3D. Niestety, mają także wady. Jedną z nich jest wysoka cena. W poniższym artykule opiszemy, jak w relatywnie prosty sposób można wykonać taki manipulator usprawniający nawigację w środowisku 3D. Dzięki zastosowaniu łatwo dostępnych modułów i druku 3D stworzenie takiego manipulatora nie jest kosztowne, a dzięki wykorzystaniu platformy Arduino można go dostosować do współpracy z niemalże każdym pakietem 3D.

Na **rysunku 1** pokazano projekt urządzenia Space Mouse, opisanego w dalszej części artykułu. Urządzenie jest wyposażone w mechaniczny joystick, zapewniający trzy osie swobody, trzy dodatkowe programowalne przyciski oraz dodatkowy przycisk wyzwalany naciśnięciem joysticka. Opisany projekt został pomyślnie przetestowany przez jego autora w programach takich jak Fusion 360 czy Inventor firmy Autodesk. Jednak z łatwością można go dostosować do potrzeb innych programów, a nawet rozbudować o dodatkowe przyciski i inne wejścia, zgodnie z naszymi potrzebami.



Fotografia 1. Urządzenie wejściowe SpaceNavigator z sześcioma stopniami swobody (commons.wikimedia.org)



Rysunek 1. Projekt manipulatora DIY Space Mouse

Listing 1. Fragmenty szkicu programu Arduino dla myszki Space Mouse

```

#include <Mouse.h>
#include <Keyboard.h>
...
void setup(){
  // Ustawia piny analogowe jako wejścia
  pinMode(horzPin, INPUT);
  pinMode(vertPin, INPUT);
  // Ustawia pin przycisku joysticka jako wejście
  pinMode(selPin, INPUT);
  // Ustawia pin przycisku 1 jako wejście
  pinMode(b1Pin, INPUT);
  // Ustawia pin przycisku 2 jako wejście
  pinMode(b2Pin, INPUT);
  // Ustawia pin przycisku 3 jako wejście
  pinMode(b3Pin, INPUT);
  // Podciąganie przycisków do góry
  digitalWrite(selPin, HIGH);
  digitalWrite(b1Pin, HIGH);
  digitalWrite(b2Pin, HIGH);
  digitalWrite(b3Pin, HIGH);
  // Krótkie opóźnienie na ustabilizowanie się układu
  delay(1000);
  // Zebranie wartości bazowych joystick
  vertZero = analogRead(vertPin);
  horzZero = analogRead(horzPin);
  // Inicjalizacja emulacji myszy
  Mouse.begin();
  // Inicjalizacja emulacji klawiatury
  Keyboard.begin();
}

void loop(){
  unsigned long currentMillis = millis();
  // Odczyt pozycji pionowej
  vertValue = analogRead(vertPin);
  // Odczyt pozycji poziomej
  horzValue = analogRead(horzPin);

  // Jeżeli naciśnięto przycisk 1
  if ((digitalRead(b1Pin) == 0) && (!b1flag)){
    b1flag = 1;
    used = 1;
    if (used != lastused){
      Keyboard.write(f8key);
    } else {
      Keyboard.write(f9key);
      used = 0;
    }
  } else if (digitalRead(b1Pin) && (b1flag)){
    b1flag = 0;
  }
  lastused = used;
  // Jeżeli naciśnięto przycisk 2
  ...
  // Jeżeli naciśnięto przycisk 3
  ...

  // Jeżeli naciśnięto przycisk joysticka
  if ((digitalRead(selPin) == 0) && (!mouseClickFlag)){
    presstime = currentMillis;
    registertime = 1;
    mouseClickFlag = 1;
    pressed = 1;
    mode = 1;
  } else if ((digitalRead(selPin)
  && (mouseClickFlag)
  && (registertime))){
    releasetime = currentMillis;
    registertime = 0;
    mouseClickFlag = 0;
    pressed = 0;
    mode = 0;

    // Sprawdza długość czasu naciśnięcia
    diff = (releasetime - presstime);
    // Wybór trybu w zależności od czasu naciśnięcia
    if(diff < interval){
      Keyboard.write(f6key);
    }
  }
}

switch(mode){ // Wybór trybu
  case 0:
    // Powrót do bazowej orientacji myszy
    invertMouse = -1;
    // Pozycja 2, zamiast 0, redukuje fluktuacje pomiędzy odczytami
    if ( (vertValue > 2) || (vertValue < (-2)) ){
      Keyboard.press(f4key);
      Mouse.press(MOUSE_LEFT);
      // Ruch myszą w osi Y z wciśniętym F4
      Mouse.move(0, (invertMouse * (vertValue / sensitivity)), 0);
      moved=1;
    }
    if ( (horzValue > 2) || (horzValue < (-2)) ){
      Keyboard.press(f4key);
      Mouse.press(MOUSE_LEFT);
      // Ruch myszą w osi X z wciśniętym F4
      Mouse.move((invertMouse * (horzValue / sensitivity)), 0, 0);
      moved=1;
    }
    if ( (vertValue <= 2)
    &&(vertValue >= (-2))
    &&(horzValue <= 2)
    &&(horzValue >= (-2))){
      // Jeżeli joystick jest w bazowej pozycji,
      // puszczamy klawisze i naciskamy ESC, by powrócić
      Keyboard.releaseAll();
      Mouse.release(MOUSE_LEFT);
    }
  }
}

```

## Budowa manipulatora

Do budowy urządzenia potrzebne nam będą:

- Arduino Leonardo/Pro Micro (nic nie stoi na przeszkodzie, by przenieść kod na inną platformę z ekosystemu Arduino, jeśli potrzebujemy np. większej liczby wejść czy dodatkowych funkcji),
- moduł joysticka z dwoma osiami ruchu i przyciskiem,
- trzy przyciski np. typu microswitch,
- śruby M2,5×6 z łbem z gniazdem imbusowym,
- inserty gwintowane M2,5 do tworzyw sztucznych,
- śruby M1,5 i odpowiednie podkładki,
- kynar lub inny drobny przewód do połączenia ze sobą wszystkich elementów.

Potrzebny będzie również dostęp do lutownicy oraz do drukarki 3D.

## Układ mechaniczny i obudowa

Konstrukcja mechaniczna manipulatora jest prosta. Obudowa składa się z dwóch części – podstawy, która mieści wszystkie elementy, i pokrywy, która zamyka konstrukcję. Dodatkowo potrzebne są drobne elementy wewnętrzne, ich kształt będzie zależał od wybranych do projektu przycisków.

Na **rysunku 2** pokazano wizualizację 3D drukowanych elementów. W procesie drukowania ustawiono rozdzielczość warstwy na 0,12 mm dla wszystkich części z wyjątkiem górnej pokrywy, której rozdzielczość druku wynosi 0,06 mm, aby uzyskać gładkie wykończenie bezpośrednio z drukarki. By proces był dokładniejszy, kosztem czasu trwania druku, nie należy przekraczać prędkości ruchu ekstrudera powyżej 40 mm/s. Pliki STL, potrzebne do druku elementów układu, dostępne są na stronie projektu (link znajduje się na końcu artykułu).

## Montaż

Pierwszym krokiem w montażu jest instalacja modułu Arduino w podstawie. Podstawka ma miejsce przygotowane do umieszczenia goldpinów, co ułatwia zamontowanie modułu. Należy po prostu wepchnąć 4 lub więcej pinów do wydruku i umieścić płytkę na górze. Aby zamocować ją na stałe, można piny przylutować, jednak należy wykonać to szybko, aby nie roztopić wydruku 3D w tym miejscu.

Następnie należy podłączyć przyciski i moduł joysticka. Za pomocą cienkiego przewodu należy podłączyć przyciski szeregowo do wspólnej linii masy (GND) i doprowadzić trzy pojedyncze przewody do wejść cyfrowych Arduino o numerach 7, 8 i 9. Aby uniknąć zewnętrznych rezystorów, podciągnięcie tych wejść do linii zasilania zrealizowano wewnętrznie w mikrokontrolerze.

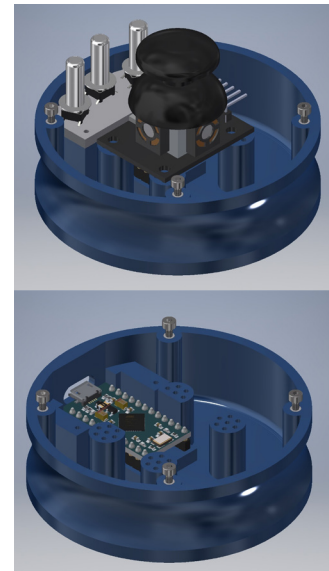
Płytkę przycisków wyposażona jest w niewielkie otworki przeznaczone do ich montażu. Aby umieścić w nich microswitche,

Listing 1. cd.

```

        if(moved==1){
            Keyboard.write(esc);
            moved=0;
        }
    }
    break;
    case 1:
        // Mysz odwrócona do przybliżania
        invertMouse = 1;
        if ( (vertValue > 2)|| (vertValue < (-2)) ){
            Keyboard.press(f3key);
            Mouse.press(MOUSE_LEFT);
            // Ruch myszą w osi Y z wciśniętym F3
            Mouse.move(0, (invertMouse * (vertValue / sensitivity)), 0);
            moved=1;
        }
        if ( (vertValue <= 2)
            &&(vertValue >= (-2))
            &&(horzValue <= 2)
            &&(horzValue >= (-2))) {
            Keyboard.releaseAll();
            Mouse.release(MOUSE_LEFT);
            if(moved==1){
                Keyboard.write(esc);
                moved=0;
            }
        }
    }
    break;
}
// Opóźnienie, które reguluje czas pomiędzy komunikatami do komputera, by system
// działał wolniej
delay(10);
}

```



Rysunek 2. Wizualizacja 3D zmontowanych elementów Space Mouse

wystarczy delikatnie rozgrzać ich piny i na ciepło wepchnąć w podstawkę. Dzięki temu tworzywo zmieknie i przyciski zostaną na trwałe zatopione w płytce. Gdy przyciski będą gotowe, należy umieścić cały moduł w podstawie i zamocować go za pomocą dwóch śrub 1,5 mm.

Teraz możemy podłączyć moduł joysticka do Arduino (VCC do VCC, GND do GND, X do A0, Y do A1 i przycisk do wejścia cyfrowego 10). Następnie joystick można umocować w podstawie za pomocą śrub 1,5 mm i małych podkładek. Podstawka została zaprojektowana z wieloma otworami do montażu modułu joysticka, aby możliwe było zastosowanie różnych modułów, dostępnych na rynku. Nie różnią się one od siebie elektrycznie, ale często mają różne zastawy otworów montażowych.

Ostatnim krokiem jest osadzenie w podstawie gwintowanych insertów, które posłużą do przykręcenia pokrywy. Elementy te należy wepchnąć na swoje miejsce za pomocą gorącej lutownicy – ciepłe inserty roztopią lokalnie tworzywo podstawy i gładko wsuną się na miejsce. Po zastąpieniu plastiku całość będzie sztywna i wytrzymała. Po osadzeniu insertów można założyć pokrywę i przykręcić ją czterema śrubami M2,5×6. Obudowę można pomalować farbą, jeśli nie podoba nam się wygląd surowego wydruku. Na **rysunku 3** pokazano zdjęcia z poszczególnych etapów montażu.

## Oprogramowanie

Space Mouse działa jako emulator myszy i klawiatury, podłączonych do komputera przez USB. W ten sposób wysyła do komputera komendy, takie jak skróty klawiszowe. Oprogramowanie układowe myszki może wymagać dostosowania do wymagań używanych przez nas narzędzi.

Pokazany na **listingu 1** fragment programu definiuje w bardzo rozbudowany sposób poszczególne parametry i funkcje przycisków. Dzięki temu z łatwością można go dostosować do swoich potrzeb, np. zmieniając funkcje (i wysyłane skróty klawiszowe) poszczególnych elementów urządzenia.

W pakietach mechanicznych firmy Autodesk program z **listingu 1** realizuje następujące funkcje

- ruch joysticka = obracanie kamery,
- szybkie naciśnięcie joysticka = powrót do bazowej pozycji kamery,
- naciśnięcie i ruch joysticka = przybliżanie kamery,
- przycisk 1 = włączenie przekroju,
- przycisk 2 = pokazanie/schowanie więzów,
- przycisk 3 = kasowanie.

Nikodem Czechowski, EP

Źródło: <https://bit.ly/2A0YqcV>



Rysunek 3. Poszczególne etapy montażu myszki