



# Koci robot sterowany modułem ESP32



*Konstrukcje robotów od dawna są wzorowane na naturze. Na rynku pojawiło się już wiele robotów, nawiązujących swoją konstrukcją do różnych zwierząt. Zaprezentowany w artykule robot wzorowany jest na kocie.*

Autor konstrukcji od dawna zafascynowany był robotami – niekoniecznie nawiązującymi swoją formą do zwierząt. Na swoim koncie ma już kopię znanego Strandbeesta, jak i innych konstrukcji Theo Jansena. Konstrukcje te jednak mają na ogół niewiele funkcji, poza umiejętnością przemieszczania się, nie są także zdalnie sterowane.

Najnowszą konstrukcją autora jest robot kot. Jest on sterowany za pomocą mikrokontrolera ESP32, co umożliwia kontrolowanie go poprzez sieć bezprzewodową Wi-Fi. Dzięki wyposażeniu w serwomechanizmy ze sterownikami robot ma wiele stopni swobody i może poruszać się niemalże jak prawdziwy kot.

## Bioinspiracje

Aby robot wyglądał jak kot, najpierw trzeba przyjrzeć się, jak wygląda to zwierzę. W szczególności jego szkielet. Na **rysunku 1** pokazano fragment wpisu dotyczącego kota

na angielskiej Wikipedii, wraz z rysunkiem samego zwierzęcia i jego szkieletu. Kot jest palcochodem, co oznacza, że stoi i chodzi na palcach. W związku z tym kot ma zauważalne trzy stawy w łapach, podczas gdy np. człowiek ma tylko dwa stawy. W przypadku przednich nóg kota śródręcza są stosunkowo krótkie, w związku z czym trzeci staw nie jest zbyt widoczny.

W konstrukcji robota autor zastosował dwa przeguby w przednich łapach i trzy przeguby w łapach tylnych. Daje to łącznie 10 serwomotorów w samych tylko łapkach. To odróżnia też konstrukcję tego robota od innych – większość robotycznych zwierzątek ma tylko po dwa przeguby w przednich i tylnych łapach.

Głowa i szyja powinny mieć wiele ruchomych, obracających się części, aby umożliwić elastyczne zmiany położenia głowy. Jeszcze trudniejszy jest ruch ogona – w rzeczywistości ogon jest połączony z dziesiątkami stawów i mięśni. Autor jednak chce

się skupić na ruchu łap, w związku z czym pozostałe elementy robota są mocno uproszczone. Wykorzystał dwa serwa do obracania i kiwania głową oraz jedno serwo do prostego obracania ogonem. Ogon kota normalnie skierowany jest ku ziemi, ale autor konstrukcji zdecydował się na skierowanie go ku górze, dzięki czemu robot lepiej się prezentuje.

## Potrzebne elementy

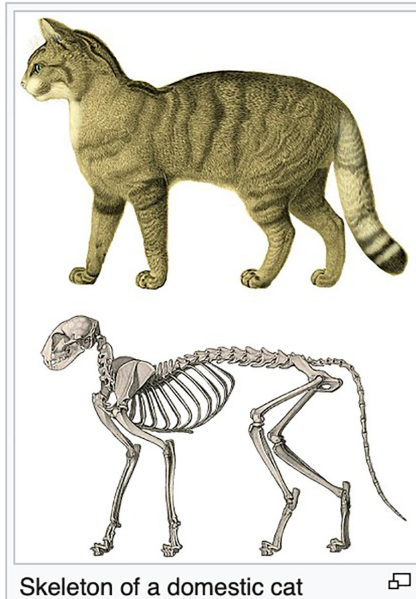
Do budowy robota potrzebne będą następujące elementy elektroniczne i elektromechaniczne:

- Moduł kamery PSRAM M5Stack, który wyposażony jest w mikrokontroler ESP32 i kamerę o rozdzielczości dwóch megapikseli;
- Moduł sterowania PWM/serwomotorami firmy Adafruit. Układ ten ma 16 kanałów i oferuje 12-bitową rozdzielczość sterowania;
- Serwomotory EMAX ES08MA II – 13 sztuk;
- Ogniwo litowo-polimerowe o napięciu wyjściowym 8,4 V. Pamiętajmy, aby zakupić także odpowiednią ładowarkę do tego akumulatora;

The anatomy of the domestic cat is similar to that of other members of the genus *Felis*.

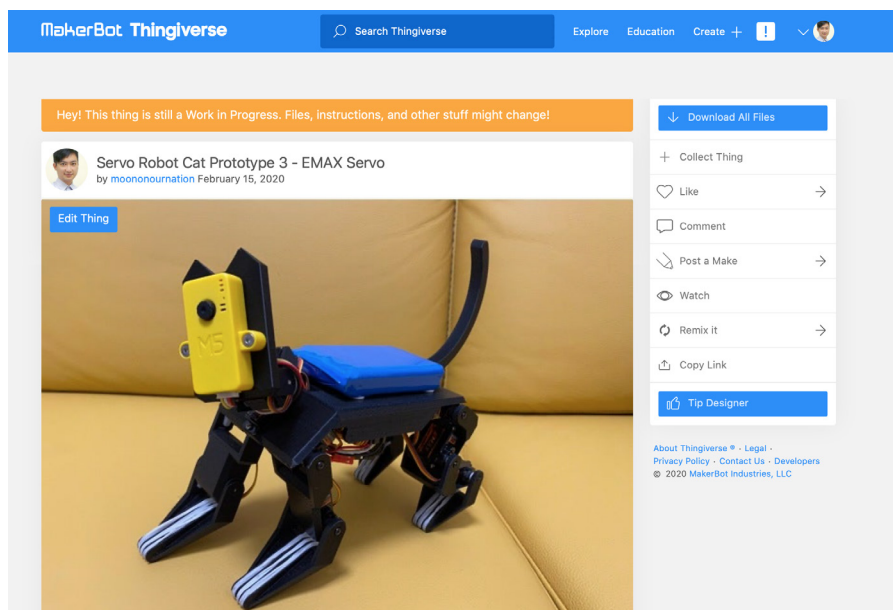
### Contents [hide]

- 1 Mouth
- 2 Ears
- 3 Nose
- 4 Legs
- 5 Claws
- 6 Temperature and heart rate
- 7 Skin
- 8 Skeleton
- 9 Muscles



Skeleton of a domestic cat

Rysunek 1. Informacje o anatomii kota domowego (<https://bit.ly/3dEDMiy>)



Rysunek 2. Projekt kota do druku 3D na portalu Thingiverse

- Stabilizator impulsowy do zasilania mikrokontrolera, kamery itp. (moduł oparty na układzie MP1584EN) oraz przetwornica do zasilania serwomechanizmów (moduł oparty na LM2596);
- Przejściówki ze złączy Grove na zwykłe goldpiny.

## Druk 3D

Prezentowany robot jest trzecią iteracją projektu mechanicznego i najpewniej nie ostatnim. Robot jest w pełni wykonany w technologii druku 3D. Wszystkie elementy dostępne są do pobrania na portalu Thingiverse (**rysunek 2**) (<https://bit.ly/3kdLVgj>). Dzięki temu można wydrukować robota we własnym domu.

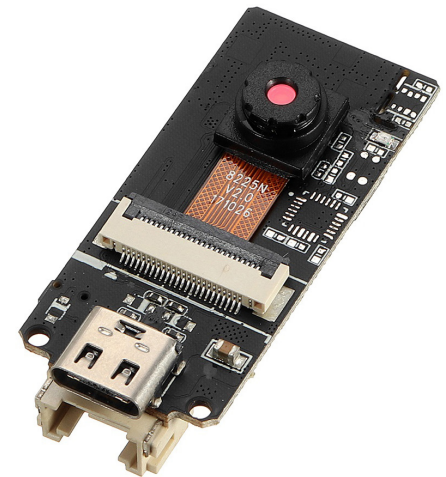
## Elektronika

Aby robot kot mógł się poruszać jak prawdziwe zwierzę, potrzebne jest elektroniczne

sterowanie dla wszystkich jego mechanizmów. Elektronika tego robota jest podzielona na kilka osobnych modułów.

### 1. Sterownik serwomechanizmów

Optymalizacja konstrukcji mechanicznej robota pozwoliła zmniejszyć liczbę ruchomych części, ale nadal wymaga on 13 serwomechanizmów do poruszania się. Sterowanie taką liczbą serwomechanizmów bezpośrednio z mikrokontrolera nie jest najlepszym rozwiązaniem. Sygnał PWM może zmieniać swoją częstotliwość, gdy mikrokontroler wykonuje inną pracę w tym samym czasie. W takiej sytuacji serwomechanizm może działać niestabilnie lub powodować nieoczekiwane zachowanie. Wykorzystanie dodatkowego sterownika, generującego sygnały do sterowania serwomechanizmów, jest znacznie lepszym rozwiązaniem. Autor zastosował moduł z układem PCA9685. Jest to 16-kanalowy sterownik PWM. Każdy z szesnastu



Fotografia 1. Wygląd modułu M5Stack

kanałów może być niezależnie konfigurowany z rozdzielczością 12 bitów poprzez interfejs szeregowy I<sup>2</sup>C. Wprawdzie układ jest przeznaczony do zasilania diod LED (w tym sterowania diodami LED RGBA w układach podświetlenia), ale równie dobrze sprawdza się w sterowaniu serwomechanizmami.

Zanim zainstalujemy wszystkie serwa w robocie, dobrze jest je przetestować. Przykładowy tester do serwomechanizmów można znaleźć na: <https://bit.ly/2IENVAh>.

### 2. Moduł z mikrokontrolerem

Do sterowania robotem można użyć wielu różnych mikrokontrolerów. Aby wybrać odpowiedni, autor sporządził listę wymagań:

- wbudowany sensor wizyjny (do wykorzystania na dalszym etapie rozwoju systemu),
- dostatecznie mały, aby zmieścić się w głowie robota,
- wyposażony w interfejs bezprzewodowy,
- wyposażony w interfejs I<sup>2</sup>C do sterowania modułem kontrolującym serwomechanizmów.

Wszystkie te wymagania spełniał moduł M5Stack pokazany na **fotografii 1**. Jest on wyposażony w SoM (System-on-Module) ESP32 firmy Espressif Systems, który zapewnia nie tylko kontrolę systemu, ale ma wbudowany interfejs bezprzewodowy, obsługujący komunikację poprzez Wi-Fi oraz

REKLAMA

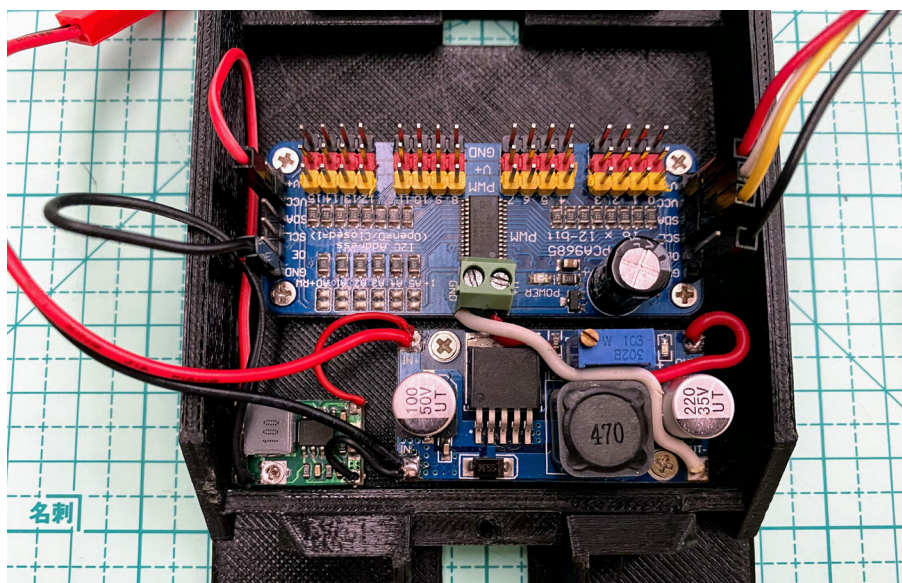
**SoMLabs**  
www.somlabs.com

**nowość!**

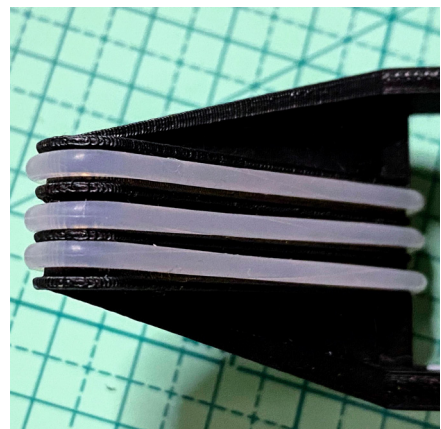
**VisionSOM-STM32MP1**  
Moduły serii VisionSOM z procesorem STM32MP1, z rdzeniami Cortex-A7 oraz Cortex-M4

Tabela 1. Zestawienie połączeń elementów elektronicznych w robocie	
<b>Bateria litowo-polimerowa</b>	
Pin dodatni	Wejście +V stabilizatora napięcia 5 V Wejście +V stabilizatora napięcia 6 V
Pin ujemny	Masa wejściowa stabilizatora napięcia 5 V Masa wejściowa stabilizatora napięcia 6 V
<b>Płytki sterownika serwomechanizmów PCA9685</b>	
Pin GND	Masa wyjściowa stabilizatora 5 V Masa wyjściowa stabilizatora 6 V Masa w złączu Grove mikrokontrolera (kabel czarny)
Pin SCL	Linia zegarowa I <sup>2</sup> C w złączu Grove (kabel żółty)
Pin SDA	Linia danych I <sup>2</sup> C w złączu Grove (kabel biały)
Pin VCC	Wyjście 5 V ze stabilizatora Linia zasilania w złączu Grove (kabel czerwony)
Pin V+	Wyjście 6 V ze stabilizatora (zasilanie serwomechanizmów)

Tabela 2. Lista kanałów sterowania i odpowiadające im funkcje serwomechanizmów	
Kanał	Lokalizacja serwomechanizmu
0	Przegub lewej przedniej łapy 1
1	Przegub lewej przedniej łapy 2
2	Przegub lewej tylnej łapy 1
3	Przegub lewej tylnej łapy 2
4	Przegub lewej tylnej łapy 3
5	Skręt głowy
6	Pochylenie głowy
7	Niepodłączony
8	Niepodłączony
9	Niepodłączony
10	Ogon
11	Przegub prawej tylnej łapy 3
12	Przegub prawej tylnej łapy 2
13	Przegub prawej tylnej łapy 1
14	Przegub prawej przedniej łapy 2
15	Przegub prawej przedniej łapy 1



Fotografia 2. Centralny element robota z zamontowanymi modułami elektronicznymi



Fotografia 3. Montaż łap robota kota

Bluetooth. Dodatkowo ma kamerę z sensorem OV2640, która umożliwia generowanie od 15 klatek na sekundę przy rozdzielczości 1600×1200 pikseli do 60 klatek przy rozdzielczości 352×288 pikseli. Na płytce znajduje się w pojedyncze złącze w standardzie Grove, na którym wyprowadzono interfejsy I<sup>2</sup>C oraz UART, a także pin GPIO. Moduł można uzupełnić na PCB o sensory takie, jak 9-osiowy moduł inercyjny (MPU6050), czujnik ciśnienia (BME280) i mikrofon analogowy (SPQ2410). Mikrokontroler programowany jest poprzez wbudowany port USB, z wykorzystaniem środowiska Arduino IDE. Więcej o programowaniu układu oraz o użytym firmware w dalszej części artykułu.

### 3. System zasilania

Podobnie jak większość płytek deweloperskich, moduł kamery M5Stack działa przy zasilaniu napięciem 5 V. Z kolei większość małych serwomechanizmów działa przy napięciu od 5 do 6 V. Moduł z mikrokontrolerem w czasie komunikacji poprzez Wi-Fi będzie pobierał nie więcej niż 150 mA. Z kolei nawet niewielki serwomechanizm

pobiera 200 mA i więcej. Przy 13 serwomechanizmach w układzie można się spodziewać szczytowego prądu dochodzącego do 2,6 A. Aby uniknąć wpływu niedoboru mocy na mikrokontroler sterujący robotem, co mogłoby powodować np. zapady napięcia i resety układu, należy zastosować osobne stabilizatory dla serwomechanizmów i dla mikrokontrolera. Napięcie wyjściowe obu stabilizatorów impulsowych, zastosowanych w układzie, jest regulowane. Należy o tym pamiętać i przed podłączeniem docelowych układów ustawić odpowiednie napięcia. Jest to 5 V dla mikrokontrolera oraz 6 V dla sterownika i serwomechanizmów. Jeśli zakupione przez nas serwomechanizmy zasilane są innym napięciem, należy dostosować ustawienie stabilizatora do wymagań naszego sprzętu.

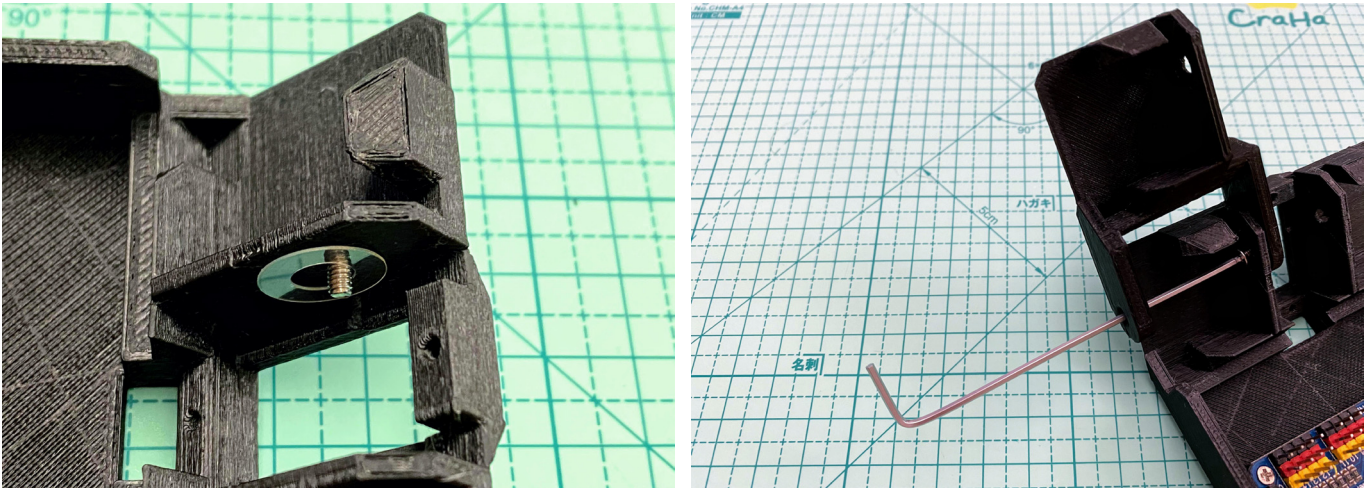
### Montaż

Montaż robota należy zacząć od instalacji komponentów elektronicznych w centralnym elemencie robota. Zmontowany element pokazano na **fotografii 2**. Instalujemy

tam moduł do sterowania serwomechanizmami oraz moduły zasilające. W **tabeli 1** zestawiono połączenia w systemie.

Po podłączeniu poszczególnych modułów możemy podłączyć serwomechanizmy. Dobrze jest podpiąć je do sterownika przed instalacją w robocie – oprogramowanie FSBrowserPlus, zainstalowane na mikrokontrolerze, ma wbudowaną inicjalizację początkową serwomechanizmów, która pozwala ustawić wszystkie serwa w środkowej pozycji zakresu obrotu. W **tabeli 2** zebrano informacje o tym, do którego kanału podłączone są poszczególne serwomechanizmy.

Po podłączeniu poszczególnych serwomechanizmów i zresetowaniu ich pozycji można rozpocząć montaż mechaniczny robota. Należy zacząć od złożenia łap robota. Tak jak pokazano na **fotografii 3**, dobrym rozwiązaniem jest nałożenie na nie okrągłego sznurka gumowego. Pozwala to zwiększyć przyczepność robota do terenu. Następnie, tak jak pokazano na **fotografii 4**, rozpoczynamy montaż poszczególnych przegubów w łapach. Później instalujemy przeguby głowy



Fotografia 4. Montaż poszczególnych przegubów w łapach robota

oraz wszystkie serwomechanizmy. Więcej dokładniejszych zdjęć, pokazujących proces montażu robota, znaleźć można na stronie projektu na portalu <https://bit.ly/2T6Lj00> (link na końcu artykułu).

## Oprogramowanie

Kluczowym elementem systemu jest oczywiście oprogramowanie. Autor, do jego budowy, wykorzystał gotową bibliotekę, dostarczaną dla ESP32 – FSBrowser. Jest to prosty serwer webowy, przeznaczony do udostępniania stron, przekopiowanych do pamięci mikrokontrolera poprzez SPIFFS. Biblioteka wyposażona jest w prosty webowy edytor tekstu, umożliwiający wprowadzanie przez Wi-Fi prostych poprawek w źródle strony oraz proste w obsłudze API, wyposażone w możliwość odczytywania wartości GPIO. Autor uzupełnił tę bibliotekę do wersji, którą nazwał FSBrowserPlus, dodając następujące komponenty:

- API do obsługi płytki z kontrolerem PWM PCA9685 do obsługi serw;
- Serwer WebSocket;
- Stronę z narzędziem do projektowania pozycji robota.

Skrypt wykorzystuje dwie biblioteki:

- ESPAsyncWebServer – do pobrania z: <https://bit.ly/2FIXn1c>;

- AsyncTCP – do pobrania z: <https://bit.ly/353obVG>.

Aby wgrać na urządzenie cały program, należy:

1. Zainstalować (jeśli jeszcze go nie posiadamy) Arduino IDE.
2. Zainstalować pakiet wsparcia dla ESP32 dla Arduino IDE. Pozwoli to kompilować i wgrywać programy na ten układ.
3. Pobrać źródła FSBrowserPlus. Dostępne są na repozytorium autora projektu (<https://bit.ly/3112lfw>), które można w całości sklonować.
4. Pobrać wspomniane wyżej biblioteki – ESPAsyncWebServer oraz AsyncTCP. Musimy zaimportować je w Arduino IDE, aby środowisko wiedziało, gdzie szukać tych plików źródłowych.
5. Podłączyć moduł M5Stack poprzez USB do komputera.
6. Po uruchomieniu szkicu FSBrowserPlus uzupełnić w kodzie SSID oraz hasło naszej sieci Wi-Fi, z którą łączyć będzie się robot.
7. Teraz możemy skompilować i załadować szkic do mikrokontrolera, naciskając przycisk ładowania w Arduino IDE.

Po wgraniu szkicu do mikrokontrolera możemy wejść na załadowaną domyślnie stronę www, która służy do projektowania

pozycji robota – <https://bit.ly/3lZXLLC> w naszej sieci lokalnej.

## Dalsze plany

Autor pisze, że jakkolwiek sam projekt rozpoczął się ponad 2 lata temu, to wciąż jest w fazie początkowej. Oczywiście, ma on dalsze plany co do rozwoju projektu. Może zainspirują one także naszych czytelników do rozwinięcia tej konstrukcji. Oto część z nich:

- Rozbudowa edytora pozycji o możliwość zapisywania sekwencji pozycji;
- Zapis pozycji i sekwencji w pamięci mikrokontrolera;
- Obsługa kamery, w tym system widzenia maszynowego do rozpoznawania obrazu;
- Redukcja wagi i lepsze wyważenie konstrukcji, zwłaszcza w sytuacji, gdy tylko 2 łapy dotykają podłoża;
- Dodanie czujników, w tym sensorów nacisku w łapach (aby zagwarantować kontakt z nierównym podłożem), inercyjny czujnik pozycji, mikrofon itp.;
- Więcej stopni swobody, zwłaszcza w szyi i ogonie;
- Dodanie głośnika, aby robot mógł miauczeć.

**Nikodem Czechowski, EP**

Źródło: <https://bit.ly/3lYqi3Z>

*O projektach, mini, soft i wielu innych  
dyskutuj  
na <https://forum.ep.com.pl>*