|  |
| --- |
| [Wpisz nazwę firmy] |
| Jeep-Bot |
| ramię robota na łaziku |
|  |
| **Łukasz Wajda** |
| **08.06.2018** |

|  |
| --- |
| [Wpisz tutaj streszczenie dokumentu. Streszczenie jest zazwyczaj krótkim podsumowaniem treści dokumentu. Wpisz tutaj streszczenie dokumentu. Streszczenie jest zazwyczaj krótkim podsumowaniem treści dokumentu.] |

**Rozdział 1**

# Wstęp

Projekt „Jeep-Bot” jest to manipulator dydaktyczny umieszczony na łaziku. Do napędu jego poszczególnych osi zastosowano silniki krokowe. Projekt można podzielić na trzy części: konstrukcję mechaniczną i elektroniczną, oprogramowanie i aplikację. Konstrukcja mechaniczna i układ sterowania robota, pozwala jedynie na wykonywanie na nim podstawowych ćwiczeń związanych z identyfikacją parametrów geometrycznych manipulatora.

## Cel i zakres pracy

Głównym celem stworzenia projektu jest wykonanie jak najtańszej, w miarę funkcjonalnej konstrukcji sterowanej prostą aplikacją. Stworzenie stanowiska z poruszającym się manipulatorem realizowane jest przez:

* wykonanie części mechanicznej Jeep-bota,
* zaprojektowanie i wykonanie elektroniki robota, pozwalającej współpracować z zewnętrznym układem sterowania,
* napisanie oprogramowania,
* stworzeni aplikacji do sterowania,
* przeprowadzenie badań.

W dokumentacji technicznej robota zamieszczono opis konstrukcji części mechanicznej i elektronicznej robota. Opisano też oprogramowanie służące do obsługi

układów wykonawczych wraz z aplikacją.

## Czas przygotowania

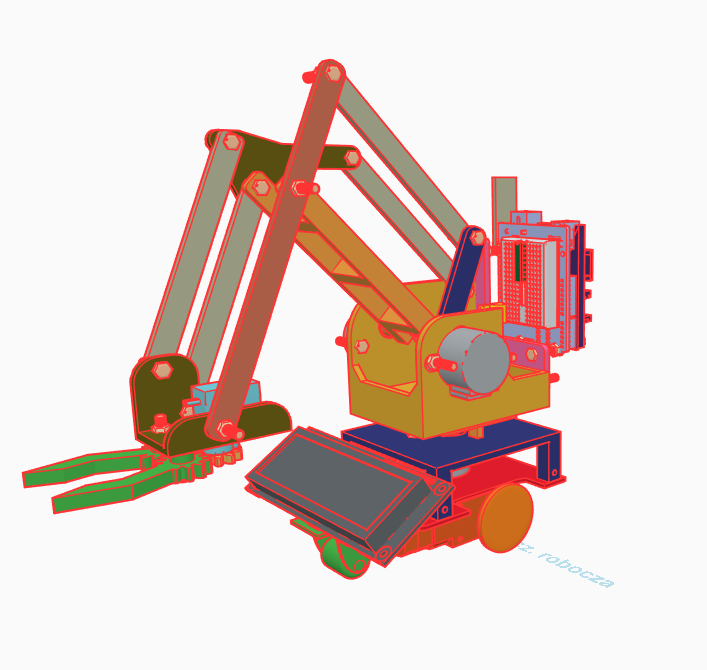
Przygotowanie projektu trwało około 2 tygodni. Ten aktywnie spędzony czas można podzielić na kilka etapów. Pierwszy to pozyskanie niezbędnych części i złożenie projektu według własnego pomysłu. Drugi to pisanie jednocześnie programu pod Arduino UNO (mikrokontroler sterujący) i pod aplikację utworzoną w MIT App Inventor, wraz z testowaniem kilku linijek kodu. Trzeci etap to z kolei jego opisanie oraz przygotowanie niezbędnych schematów.

**Rozdział 2**

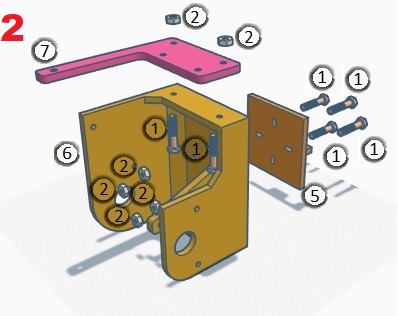
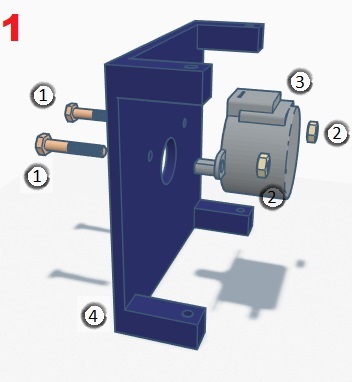
# Konstrukcja robota „Jeep-bot”

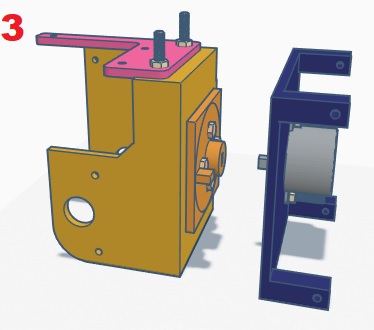
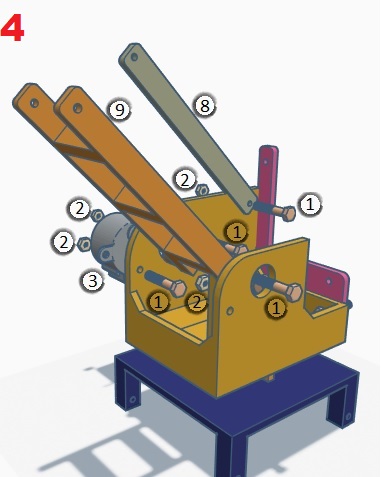
## Konstrukcja mechaniczna

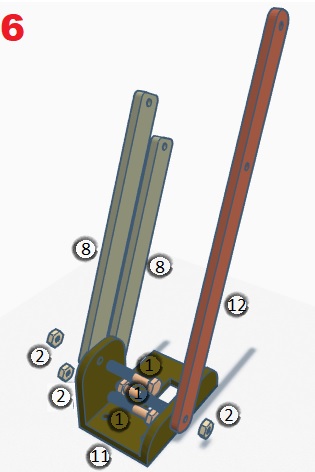
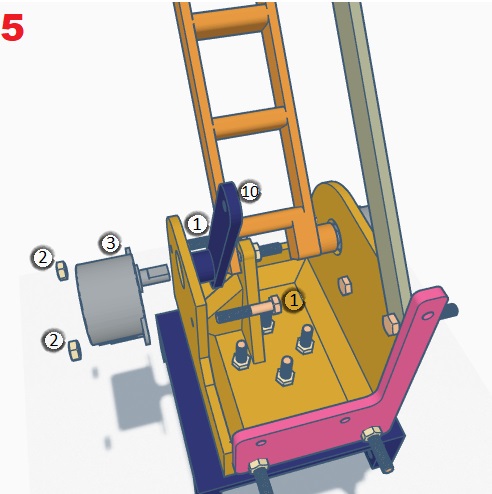
Projekt „Jeep-Bot” można podzielić na dwie podstawowe części: Ramię i Łazik. Rysunek ramienia powstał na podstawie projektu manipulatora MeArm Classic (<https://shop.mime.co.uk/products/mearm-pocket-sized-robot-arm>). Pozwoliłem sobie na kilka modyfikacji i udoskonaleń, ponieważ jego pierwowzór był przeznaczony na serwa modelarskie. Ja natomiast zaopatrzyłem się w tanie i dość wydajne silniki krokowe.

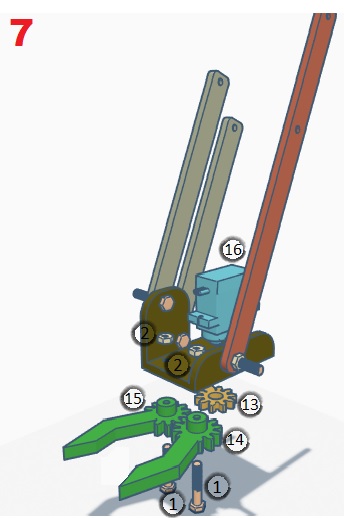
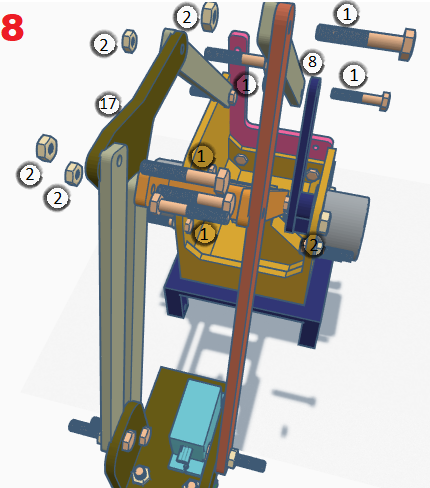


Do modyfikacji manipulatora posłużyłem się prostym i darmowym narzędziem online do tworzenia i edycji modeli 3D o nazwie TinkerCAD (jest własnością firmy Autodesk). Umieszczenie silników w skrzynce ma swoje pozytywne konsekwencje, a mianowicie stanowią on swego rodzaju przeciwwagę, co ma znaczenie ze względu na utrzymanie równowagi potrzebnej podczas ruchu głównego ramienia. Ramię zostało w całości wydrukowane na drukarce 3D. Jako materiał wybrałem ABS -[tworzywo](https://pl.wikipedia.org/wiki/Tworzywa_sztuczne) otrzymywane w procesie polimeryzacji, które cechuje się dużą [udarność](https://pl.wikipedia.org/wiki/Udarno%C5%9B%C4%87), twardością oraz odpornością na zarysowania. Jednak największym mankamentem tego materiału jest odklejanie się cienkich przedmiotów od stołu drukarki w czasie drukowania. Poniżej przedstawiono poszczególne kroki łączenia elementów.

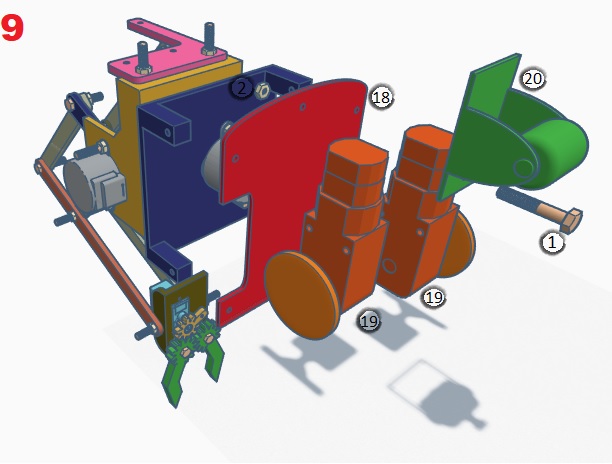


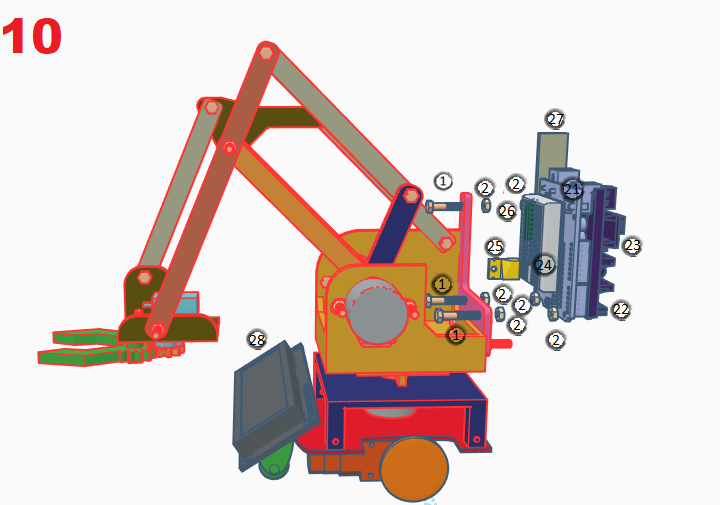


Płytki laminatu pod łazik zostały wycięta szlifierką i sklejone ciepłym klejem na których wspiera się manipulator. Poniżej zostało przedstawione umiejscowione dwóch silników i jednej rolki.



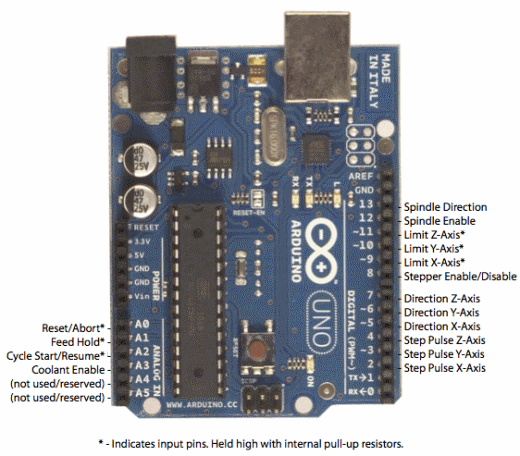
|  |  |  |
| --- | --- | --- |
| LISTA CZĘŚCI | | |
| Numer | Nazwa Części | Ilość |
| 1 | M3 x 20mm Śruba | 29 |
| 2 | Nakrętka M3 | 32 |
| 3 | Silnik krokowy 28BYJ-48 | 3 |
| 4 | Podstawa manipulatora | 1 |
| 5 | Łącznik z silnikiem osi X | 1 |
| 6 | Skrzynia | 1 |
| 7 | Uchwyt na Arduino | 1 |
| 8 | Łącznik 100mm | 4 |
| 9 | Ramię silnika osi Z | 1 |
| 10 | Ramię silnika osi Y | 1 |
| 11 | Podstawa chwytaka | 1 |
| 12 | Łącznik 160mm | 1 |
| 13 | Koło zębate serwa | 1 |
| 14 | Chwytak lewy | 1 |
| 15 | Chwytak prawy | 1 |
| 16 | Micro Serwo 9g Tower Pro | 1 |
| 17 | Prawy górny łącznik | 1 |
| 18 | Płytka laminatu | 1 |
| 19 | Silnik DC z redukcją | 2 |
| 20 | Płytka obrotowa z kółkiem 22mm | 1 |
| 21 | Arduino Uno | 1 |
| 22 | CNC Shield | 1 |
| 23 | Sterownik [A4988](http://botland.com.pl/sterowniki-silnikow-krokowych/148-sterownik-silnika-krokowego-a4988-reprap-35v-2a-modul-pololu.html) | 3 |
| 24 | Płytka stykowa | 1 |
| 25 | Stabilizator LM7805 | 1 |
| 26 | Mostek H L293d | 1 |
| 27 | Moduł HC-06 | 1 |
| 28 | Wyświetlacz LCD 2x16 z konwerterem I2C HD 44780 | 1 |



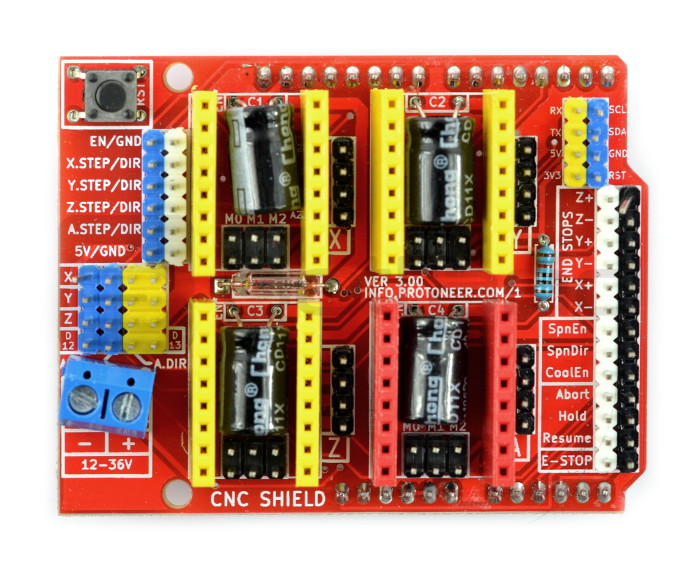
Pliki do druku zostały umieszczone w załączniku.

## Konstrukcja elektroniczna

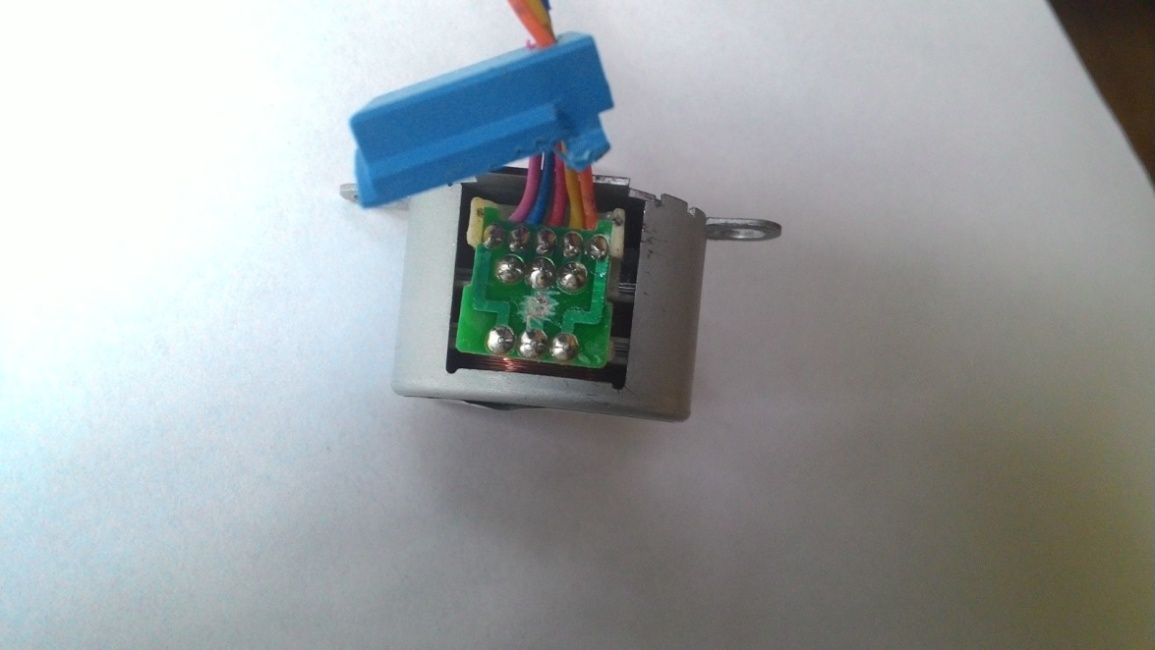
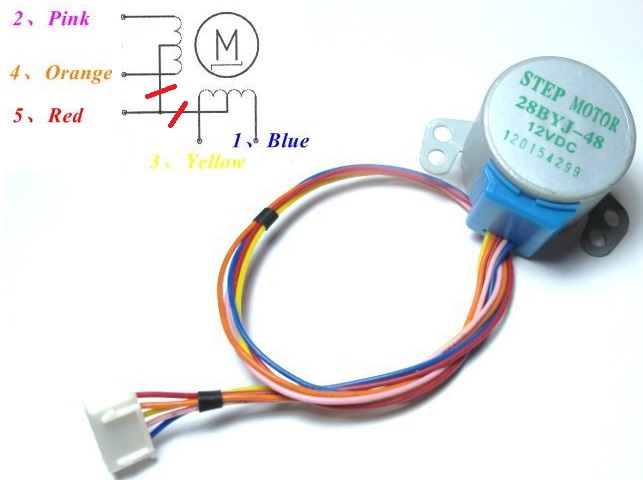
Sercem odpowiedzialnym za dokonywanie obliczeń, odbieranie danych i ich wysyłanie jest Arduino UNO, którego opis wyprowadzeń dla CNC shield i przykładowe umieszczenie przedstawiono poniżej.



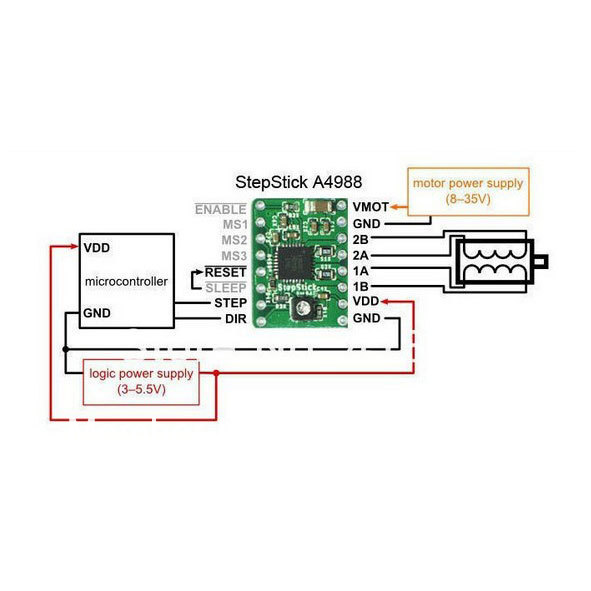
Specjalna nakładka CNC shield pozwala na wygodne umiejscowienie czterech sterowników silników  [A4988](http://botland.com.pl/sterowniki-silnikow-krokowych/148-sterownik-silnika-krokowego-a4988-reprap-35v-2a-modul-pololu.html). W znaczący sposób ułatwia to korzystanie z silników krokowych odpowiedzialnych za poruszanie ramieniem w różnych płaszczyznach x, y i z .



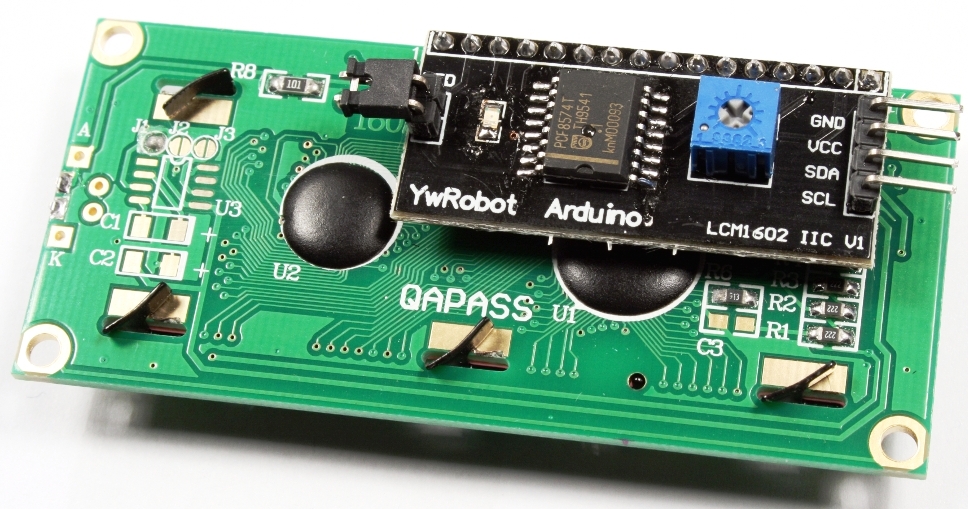
Silniki krokowe 28BYJ-48 są silnikami unipolarnymi, natomiast płytka jak i sterowniki są przeznaczone na silniki bipolarne. Ich przemiana była bardzo prosta i obejmowała ściągnięcie obudowy i zarysowanie, usunięcie jednej ścieżki łączącej środki uzwojeń silników.



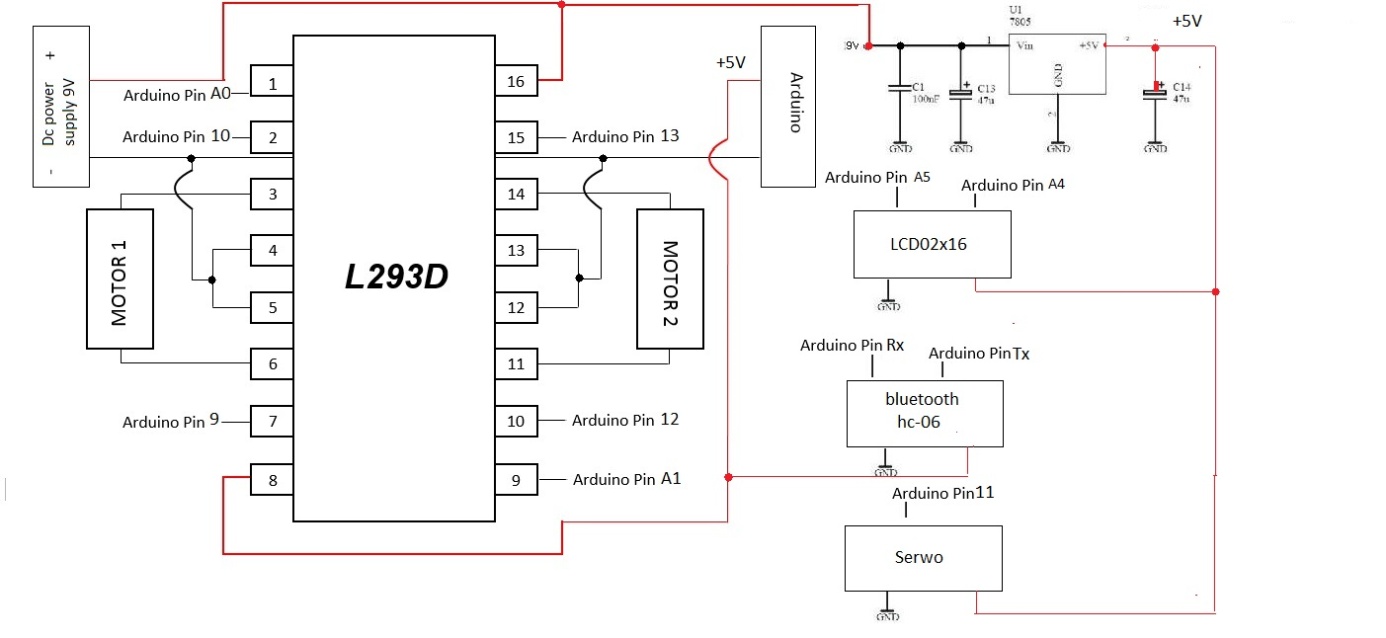
Do sterowania silnikami krokowymi za pomocą [A4988](http://botland.com.pl/sterowniki-silnikow-krokowych/148-sterownik-silnika-krokowego-a4988-reprap-35v-2a-modul-pololu.html) wymagane są dwa piny sterujące STEP i DIR. Jeden impuls podany na pin STEP powoduje jeden krok silnika w kierunku wybranym poprzez podanie odpowiedniego stanu logicznego na pin DIR.



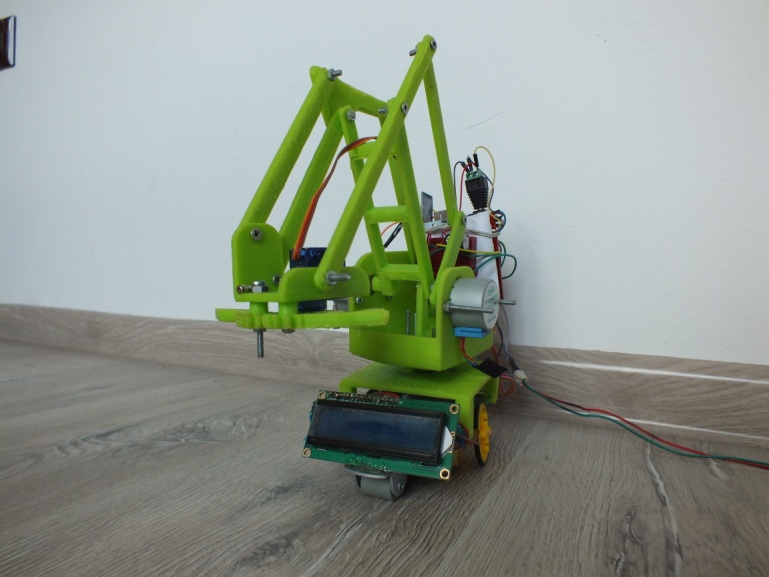
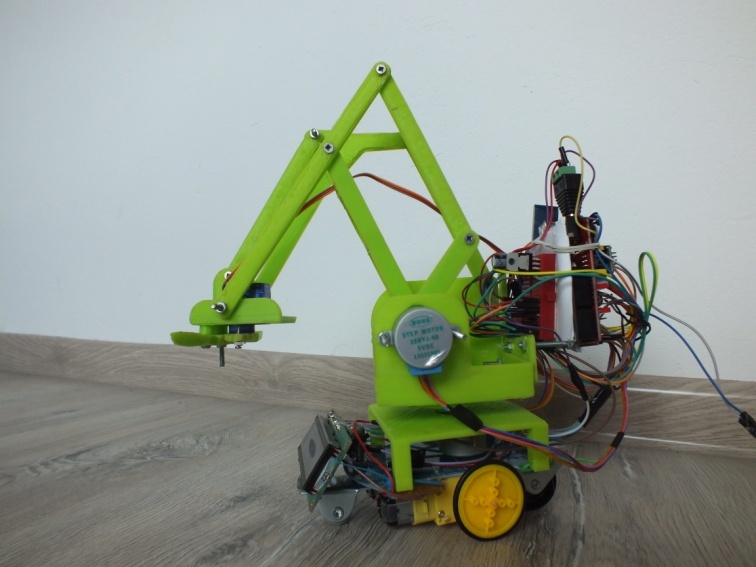
Mechanizmem który porusza chwytakiem w ramieniu jest serwo, wymaga tylko jednego pinu sterującego. Wyświetlacz LCD 2x16 został połączony z konwerterem I2C HD 44780. Dzięki temu połączenie między Arduino UNO, a wyświetlaczem zajmuje 4 piny (SCL, SDA, Vcc i GND).

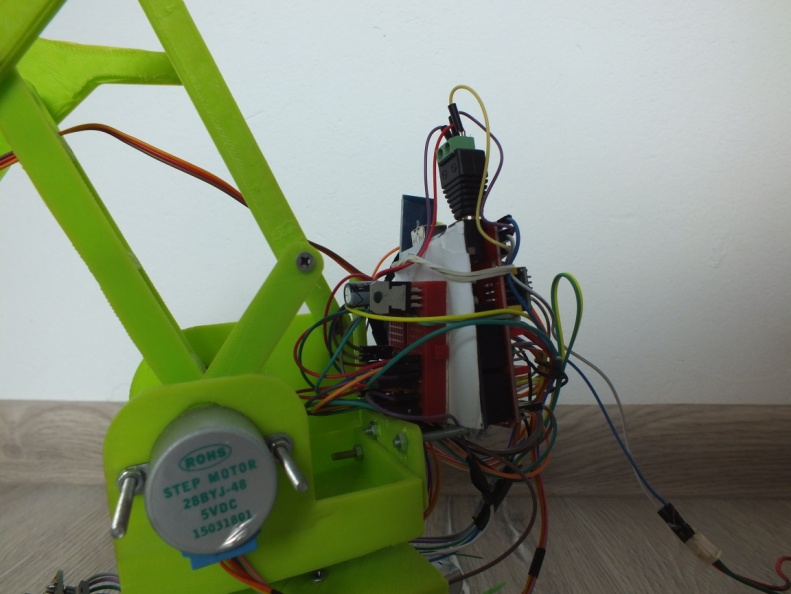


Silniki DC łazika są odpowiednio połączone z L293d, mostkiem H (zgodnie z poniższym schematem). Całość jest zasilana napięciem 9V i 5V po stabilizatorze LM7805.



Najważniejszym elementem zapewniającym komunikację z aplikacją jest moduł Bluetooth HC-06. Na koniec wszystko powinno wyglądać tak

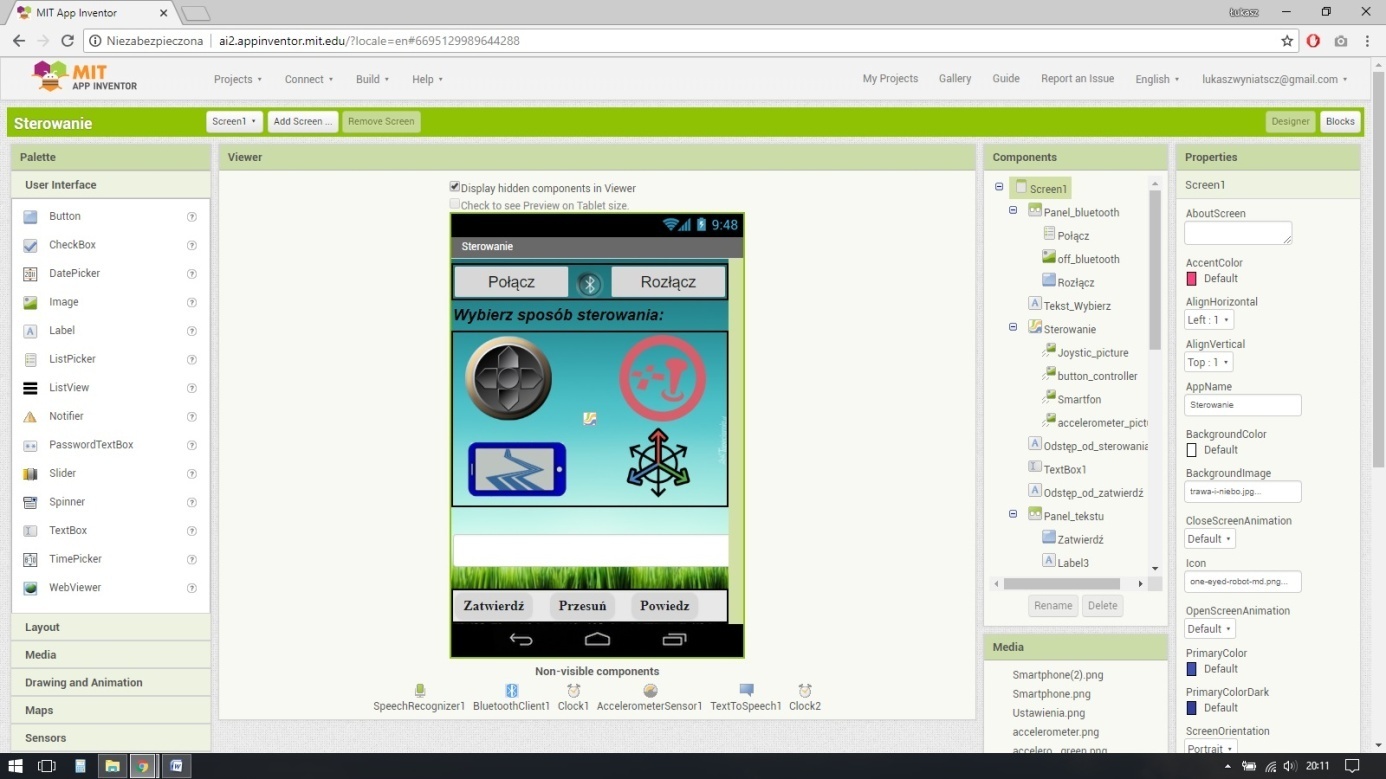




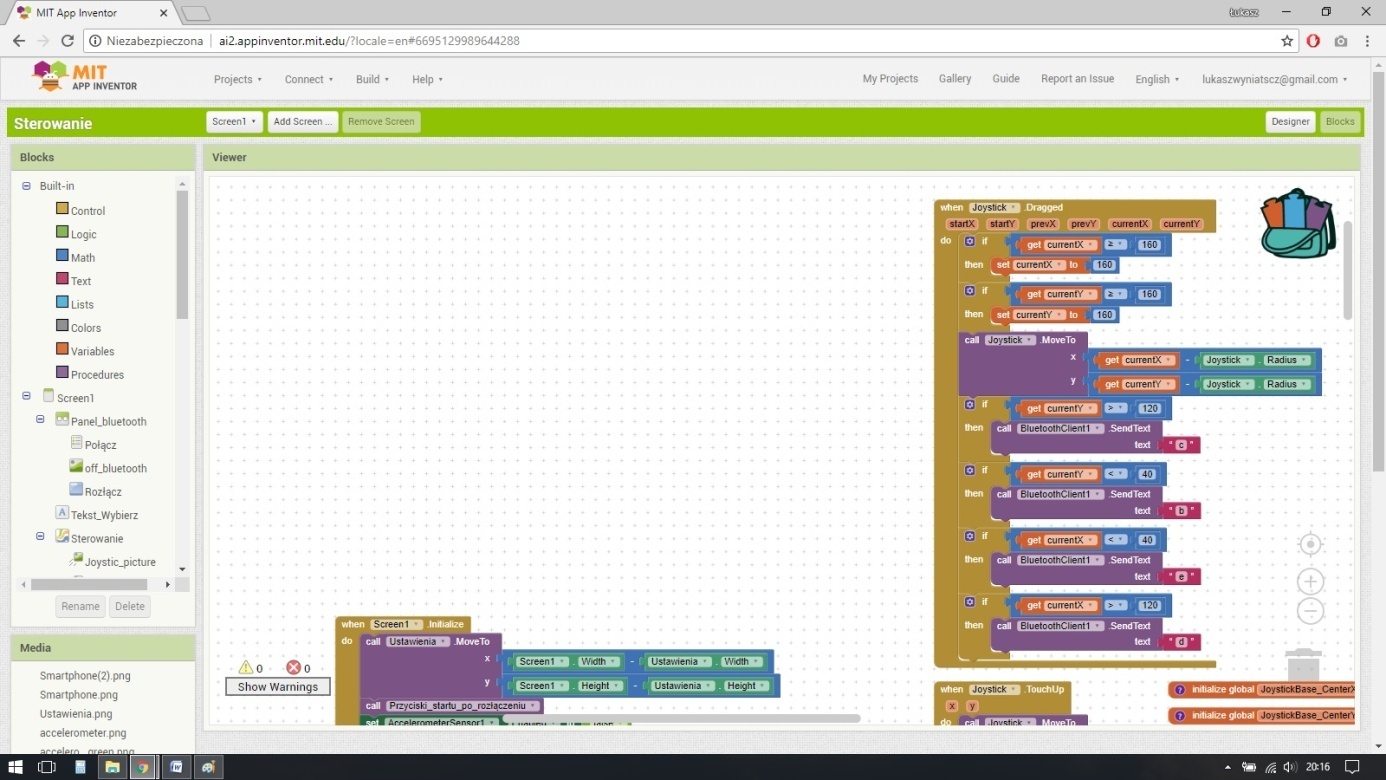
## Aplikacja

Do stworzenia aplikacji użyłem programu wizualnego MIT App Inventor. App Inventor umożliwia tworzenie aplikacji dla telefonów z systemem Android za pomocą przeglądarki internetowej i podłączonego telefonu lub emulatora. Budujemy aplikacje poprzez współpracę z:

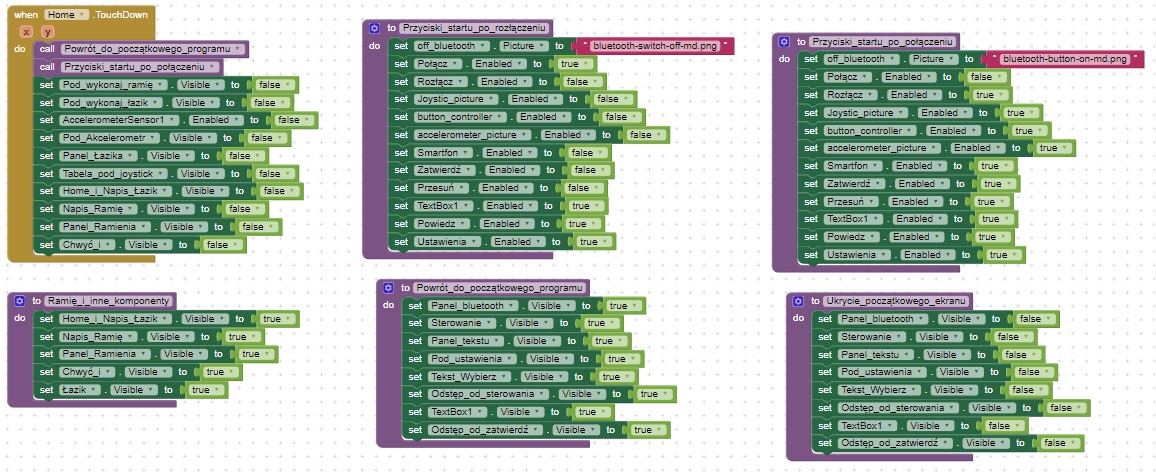
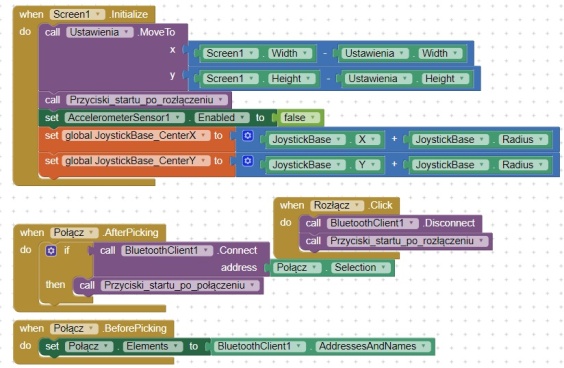
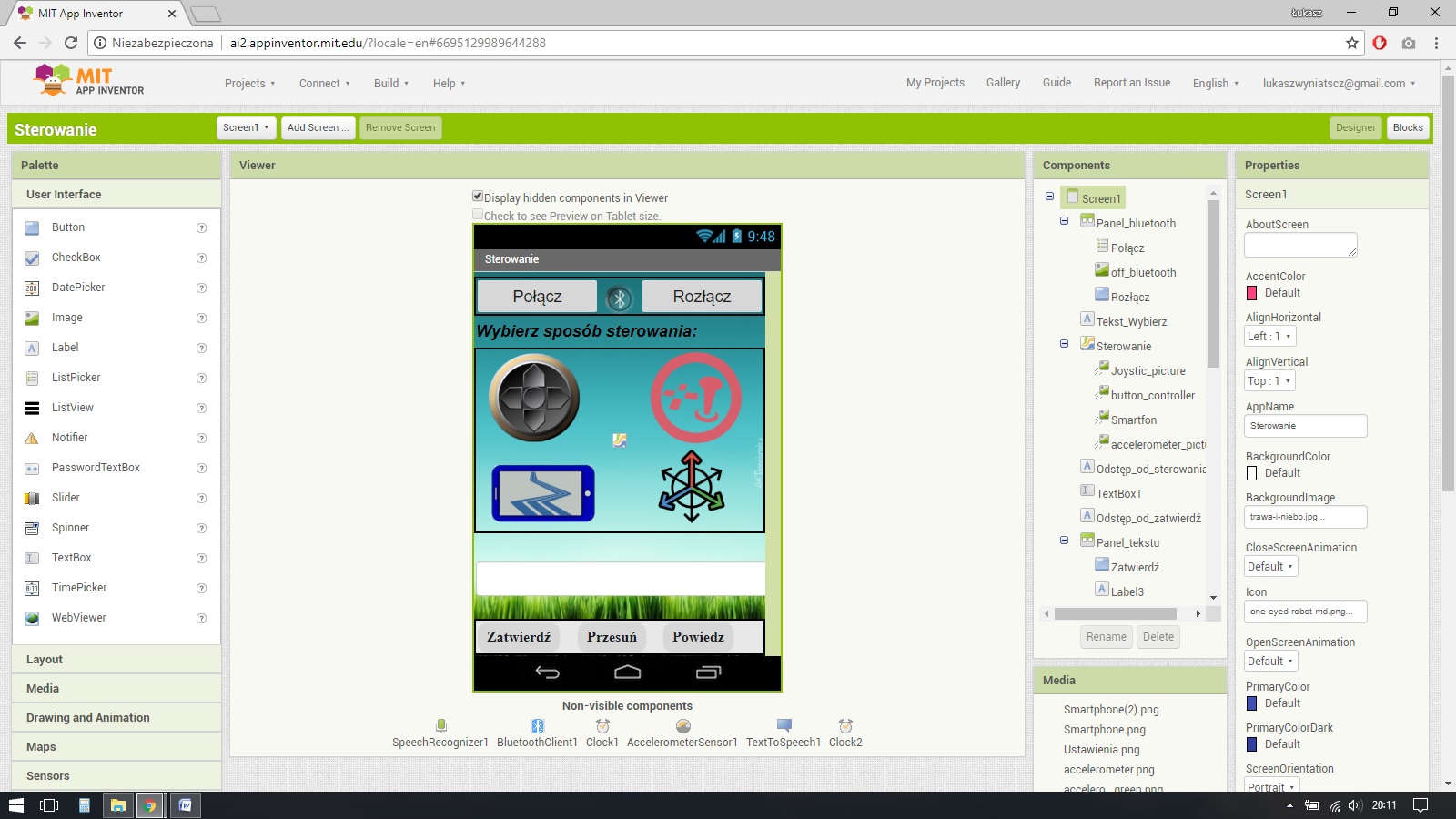
* App Inventor Designer, gdzie można wybrać składniki.



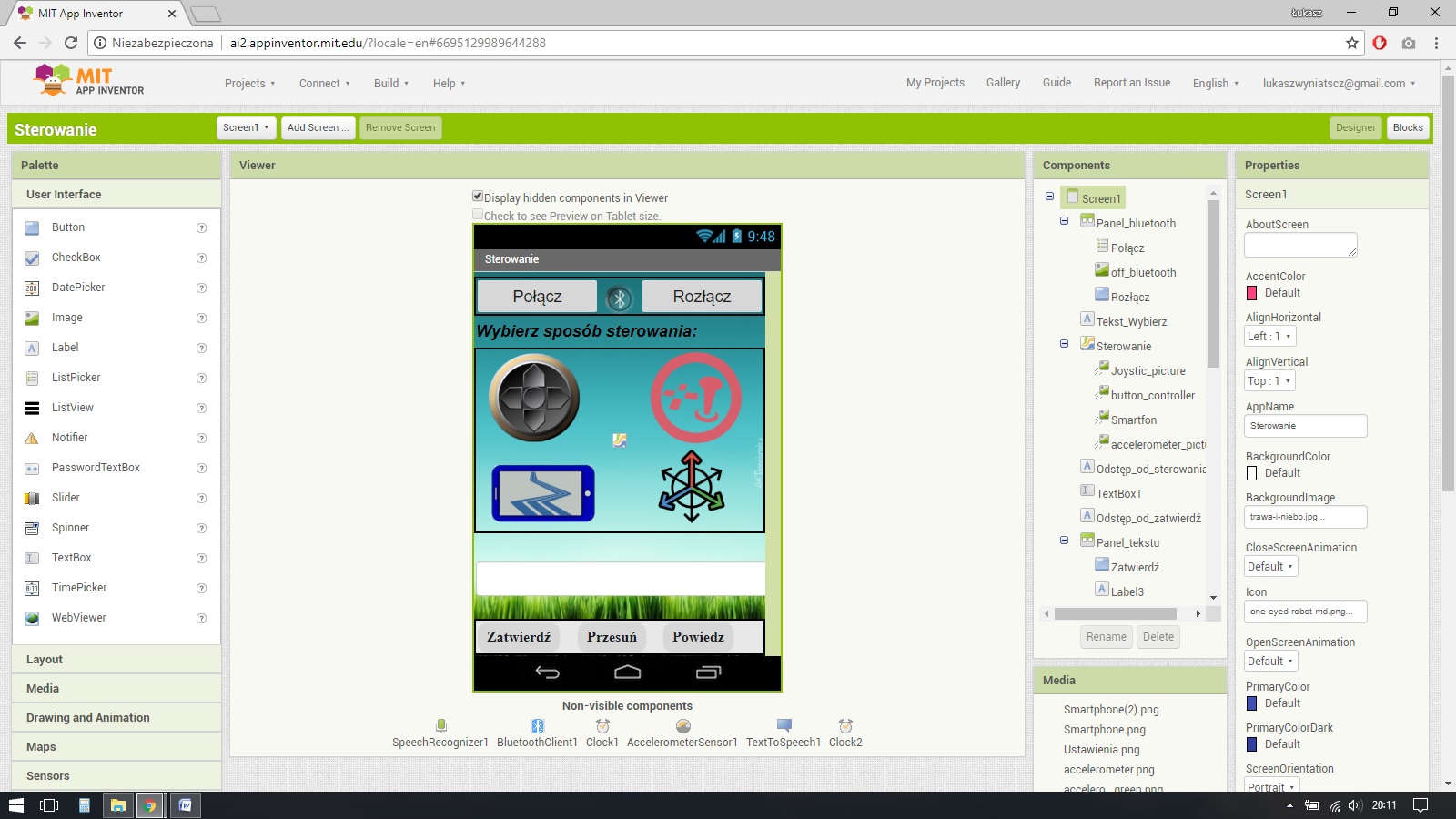
* App Inventor Blocks Editor, gdzie można zbudować bloki programu, które określają, w jaki sposób elementy powinny się zachowywać.



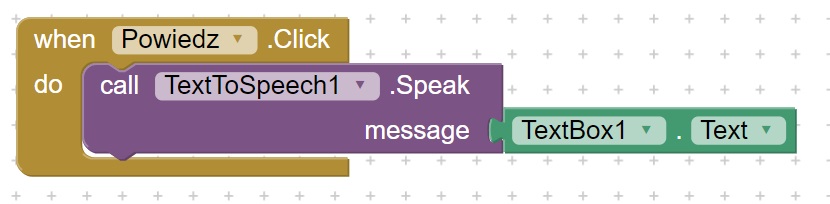
Każdemu składnikowi na ekranie tworzonej aplikacji odpowiadają konkretne bloki. Na samym początku okno startowe aplikacji wymaga od nas połączenia za pomocą Bluetooth. Wówczas aktywny jest tylko przycisk Połącz, dzięki procedurze przyciski\_startu\_po\_rozłączeniu. Do inicjalizacji ekranu wchodzą również komendy takie jak : akcelerometr jest wyłączony i ustawiany joysticka w odpowiedniej pozycji, które szerzej zostaną opisane później.



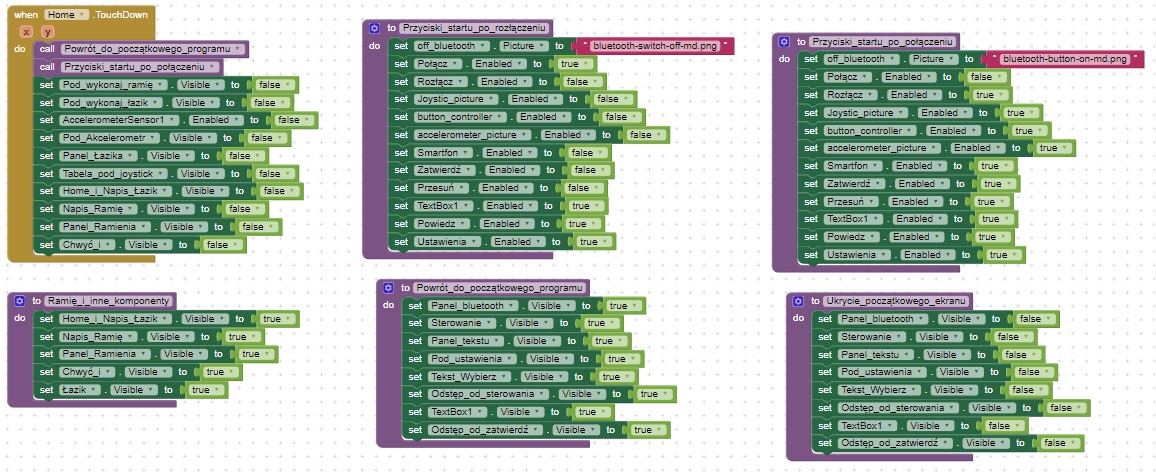
Po wybraniu znanego nam połączenia bluetooth mamy do wyboru kilka metod sterowania łazikiem, jak np.: przyciski, joystick, akcelerometr oraz przykładowy ruch i jedną metodę sterowania ramieniem.



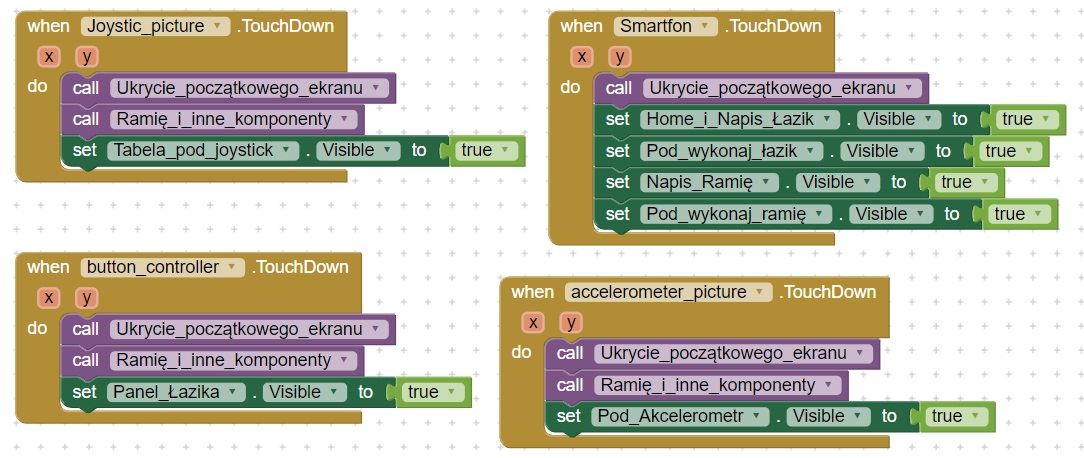
Dodatkowo możemy wysłać przykładowy tekst na wyświetlacz LCD, przesunąć go i sprawić, że nasz smartfon wypowie napisane słowo.



Po wybraniu dowolnego sposobu sterowania np. za pomocą przycisków, wykonywane są odpowiednie procedury odpowiedzialne za zakrywanie bądź odsłanianie konkretnych komponentów, tak aby uzyskać żądany ekran. Poniżej przedstawiono używane procedury.



Co nastąpi kiedy naciśniemy wybrany przycisk.



Uzyskane ekrany pod sterowanie:

* Przyciskami,



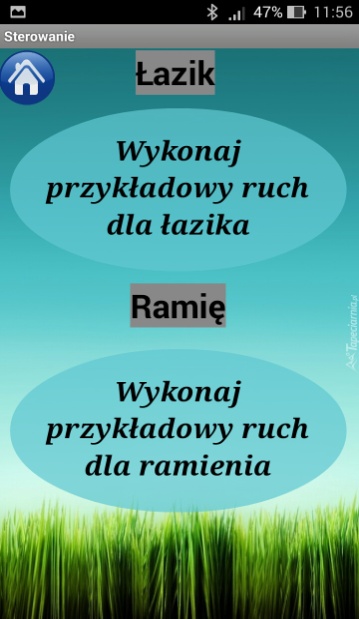
* Joystick,



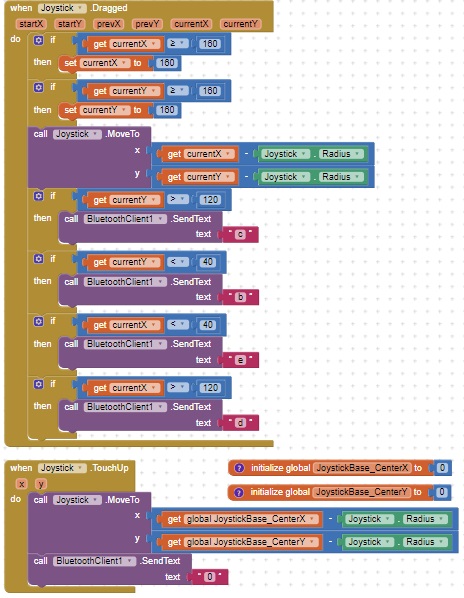
* Akcelerometr,



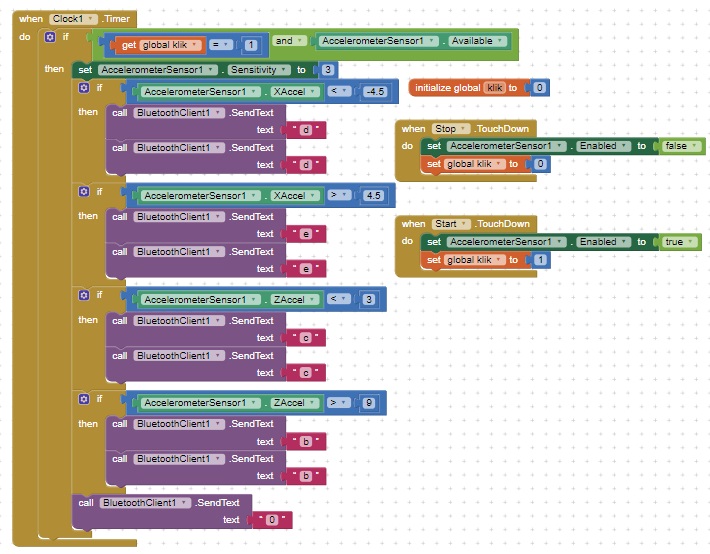
* Przykładowy ruch.

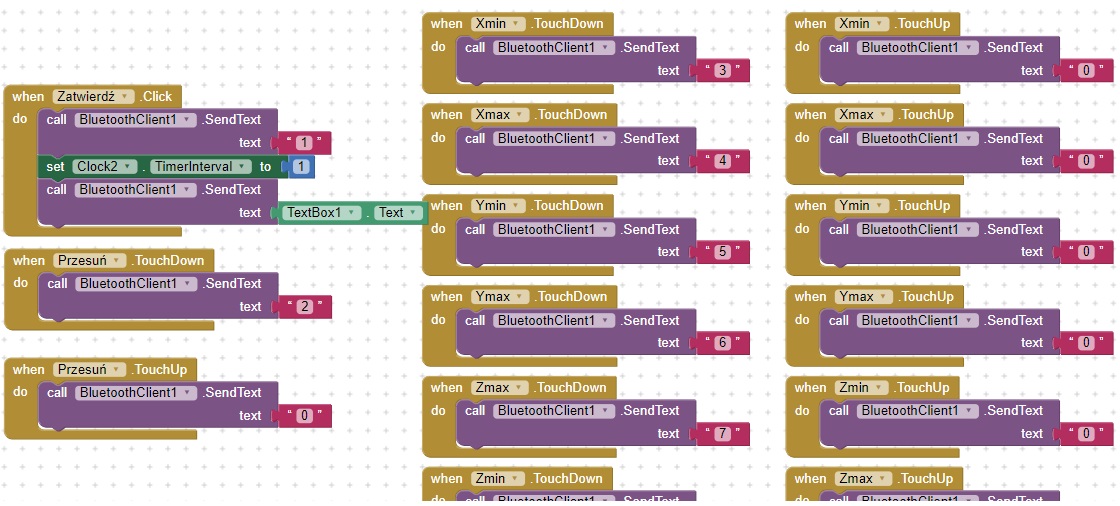


W przypadku joysticka i akcelerometru musiałem utworzyć martwe pole, w którym nie jest wysyłana żadna komenda do łazika. Z przedstawionych bloków można się domyślić, że dla joysticka pole to jest kwadratem o boku 80 pikseli i po jego przekroczeniu wysyłana jest odpowiednia widomość. Ogólnie czerwony joystick może się poruszać po polu czarnego koła o średnicy 160 pikseli. Kiedy joystick jest nieużywany wraca do położenia środkowego martwego pola.



Dla akcelerometru sprawa ma się podobnie. W tym przypadku działamy tylko w dwóch płaszczyznach, osi X i Z. Do każdej osi jest przyporządkowany odpowiedni przedział, w którym jest wysyłana komenda „0” odpowiedzialna za bezruch. Dopiero jej przekroczenie skutkuje ruchem łazika.



Aplikacja jest tak zaprojektowana, aby w czasie naciśnięcia przycisku, bądź zadziałaniu akcelerometru lub joystick smartfon wysyłał poszczególne znaki np.: ‘1’, ‘2’, ‘A’, ’B’, ‘C’ itp. 

Kolejno moduł Bluetooth odbiera ciąg znaków, które za pomocą programu są przekształcane na poszczególne czynności. Program z opisem pod mikrokontroler wraz z aplikacją w załączniku.