



**Akademia Górniczo-Hutnicza
im. Stanisława Staszica w Krakowie**

Wydział Inżynierii Mechanicznej i Robotyki

**Sprawozdanie z projektu z przedmiotu
Mechatronic Design.**

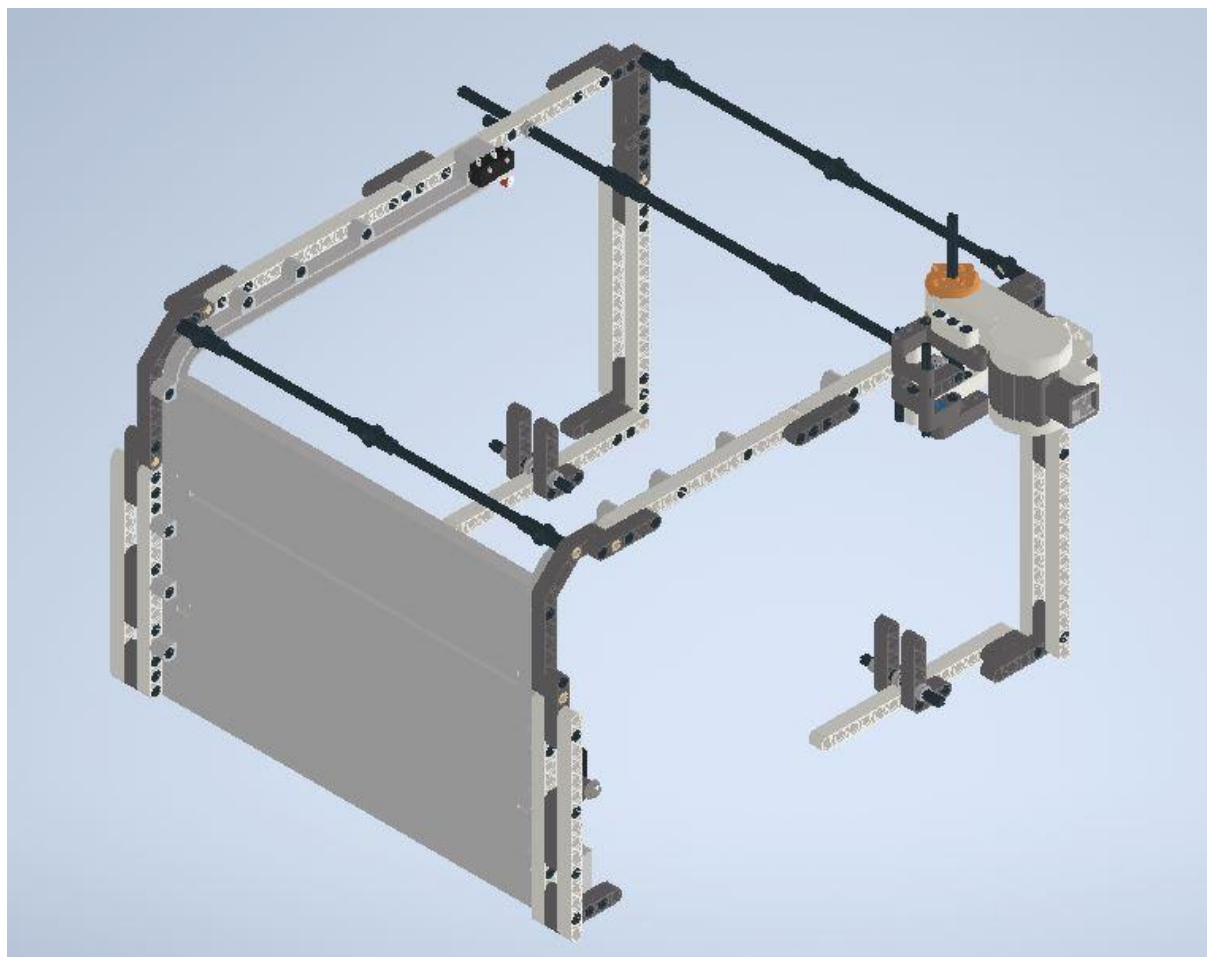
Temat: Brama garażowa.

Autorzy: Bałazy Patryk, Haręźlak Krzysztof, Kmak Dawid, Marczyński Bartosz,
Marciniec Piotr, Rodak Wojciech, Buda Krzysztof, Jordanek Jakub, Jurek
Krzysztof, Gut Paweł, Jakielaszek Miłosz, Knap Paweł.

Sprawozdanie z realizacji części mechanicznej projektu bramy garażowej.

Zespół mechaniki: Bałazy Patryk, Harężlak Krzysztof, Kmak Dawid,
Marczyński Bartosz, Marciniec Piotr, Rodak Wojciech

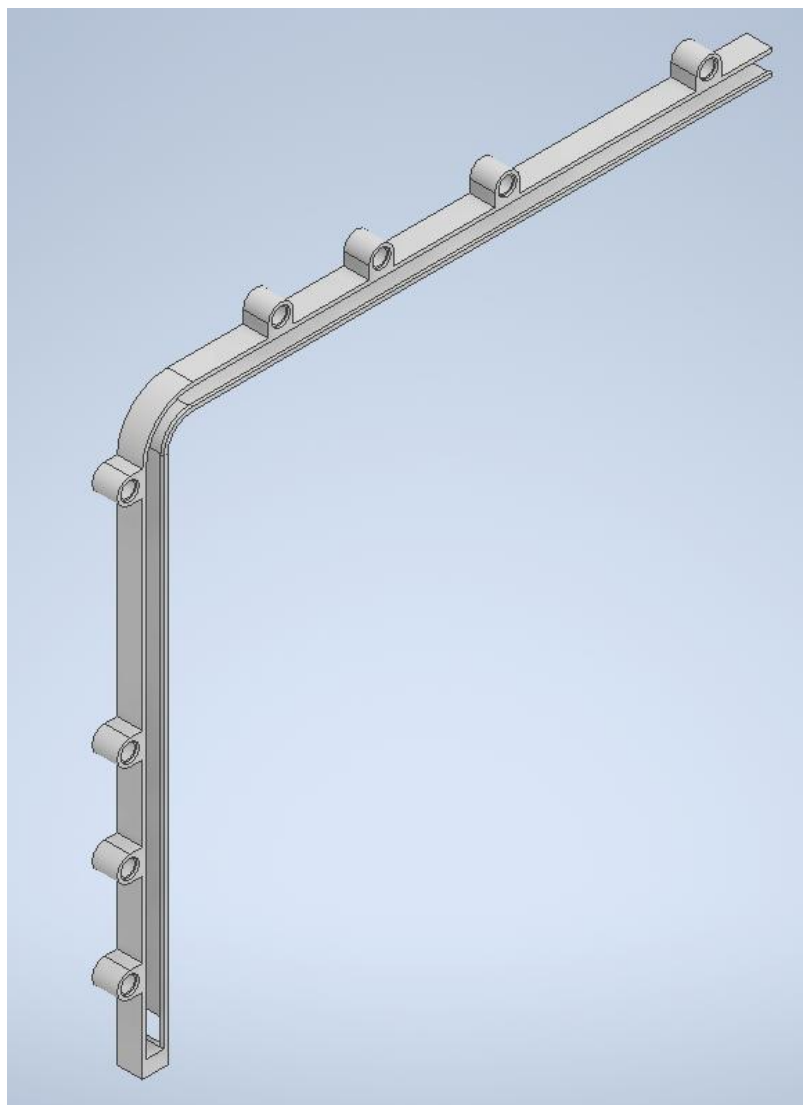
Zadaniem naszego zespołu było zaprojektowanie i wykonanie konstrukcji bramy garażowej. Po omówieniu różnych rozwiązań zdecydowaliśmy się na stworzenie bramy segmentowej. Głównym problemem, z którym przyszło nam się zmierzyć było przymocowanie prowadnic, ponieważ w praktyce prowadnice bramy przymocowywane są do ścian budynku. Z tego względu wykonaliśmy szkielet umożliwiający poprawne działanie mechanizmu. Niestety nie mieliśmy możliwości wykonania modelu fizycznie ze względu na brak dostępu do wystarczającej ilości klocków LEGO. W związku z tym użyliśmy programu Autodesk Inventor Professional. Finałowy efekt naszej pracy można zaobserwować na poniższym zdjęciu:



Zdj. 1 Widok bramy garażowej

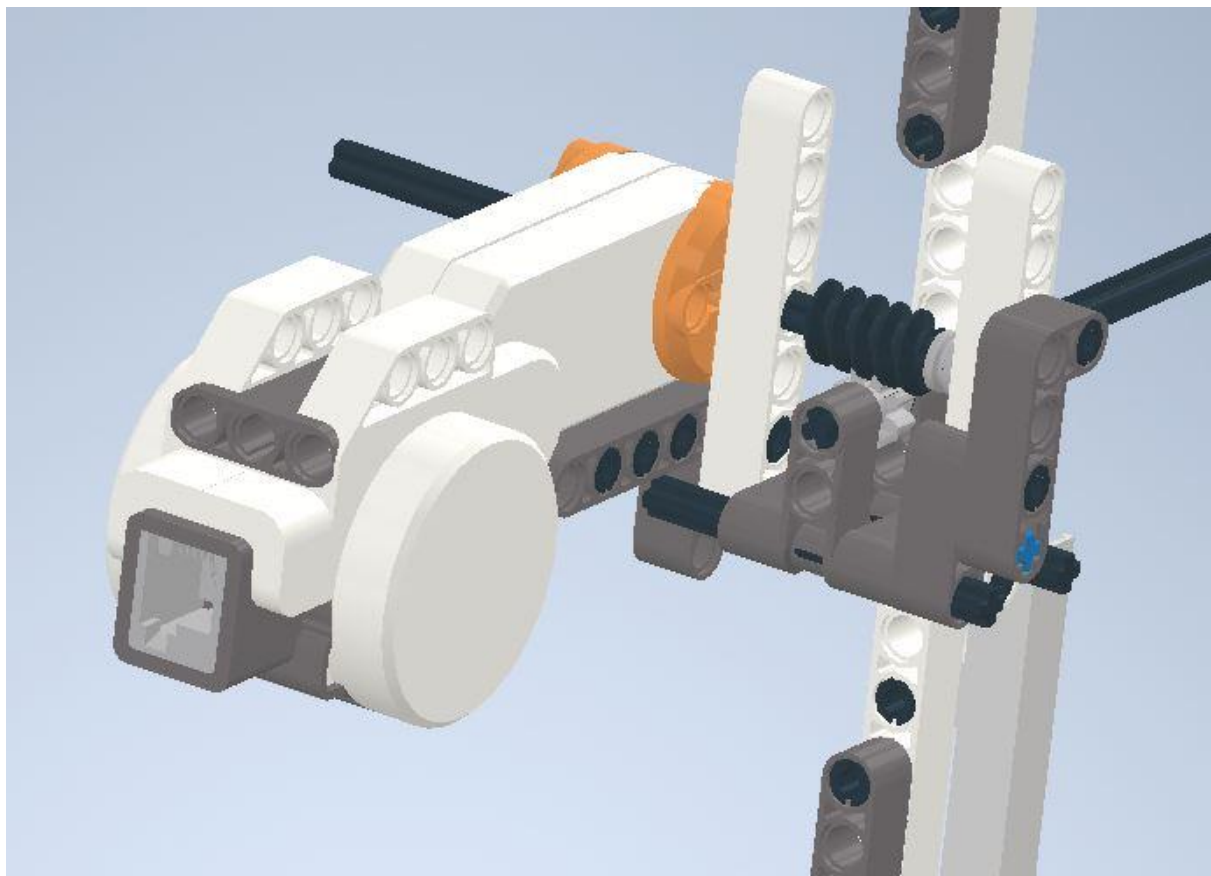
Sam szkielet służący do stabilizacji i utrzymania bramy jest wykonany z samych elementów LEGO. Musieliśmy wykonać prowadnice, ponieważ nie było możliwości stworzenia ich z klocków.

Otwory w prowadnicach służące do złączenia ze szkieletem bramy mają taką średnicę co otwory w elementach LEGO. Umożliwia to bezproblemowe połączenie.



Zdj. 2 Prowadnica

W naszym modelu moment z silnika na wał przekazuje przekładnia ślimakowa. Brama jest podnoszona lub opuszczana w zależności od kierunku obrotów silnika. Jeśli jest podnoszona, linki, które znajdują się na rolkach na obydwóch końcach wału, są zwijane i wciągają kolejne segmenty w górę po prowadnicach. Opuszczanie bramy następuje poprzez rozwijanie linek, w skutek czego segmenty zjeżdżają w dół po prowadnicach.



Zdj. 3 Napęd, przekładnia ślimakowa

Sprawozdanie z realizacji części elektrycznej projektu bramy garażowej.

Zespół elektroniki: Gut Paweł, Jakielaszek Miłosz, Knap Paweł

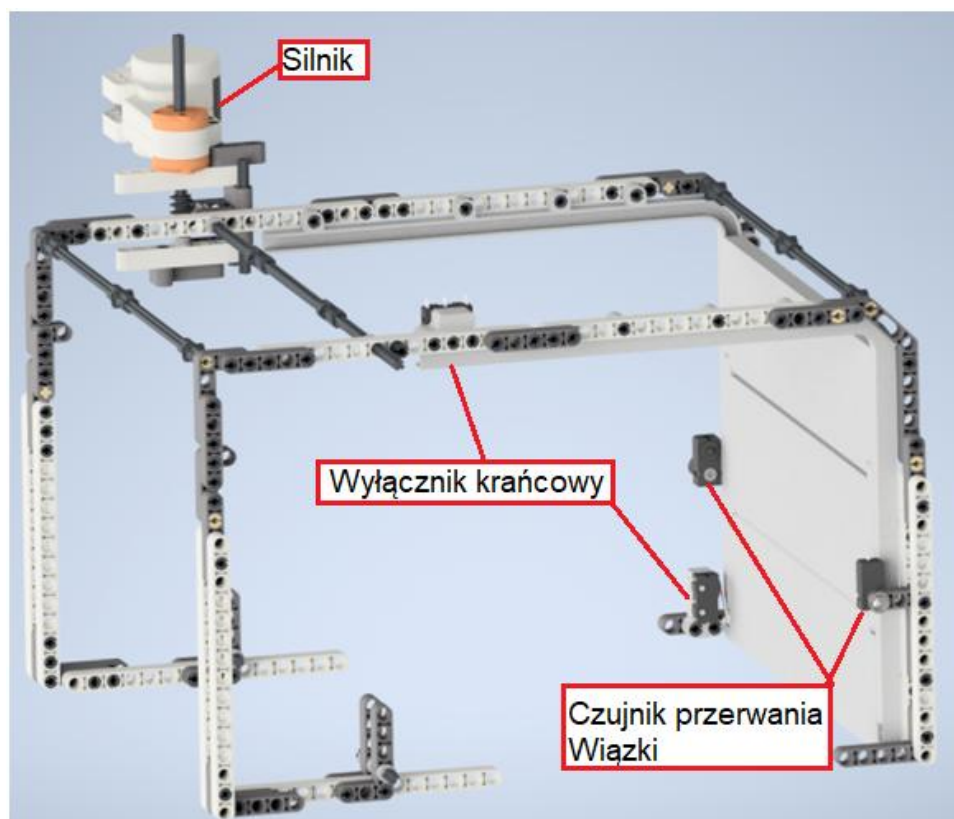
W części elektronicznej celem było stworzenie koncepcji układu sterowania bramą garażową, dobranie niezbędnych podzespołów oraz stworzenie schematu połączeń.

Brama garażowa sterowana jest za pomocą Arduino Uno. Do mikrokontrolera zostały podłączone następujące elementy

- ESP32 – jednostka komunikacyjna
- L293D - sterownik silnika
- Lego NXT Motor - napęd bramy
- Bateria 9V – zasilanie silnika
- WK625 - wyłącznik krańcowy
- QT 50CM - czujnik przerwania wiązki

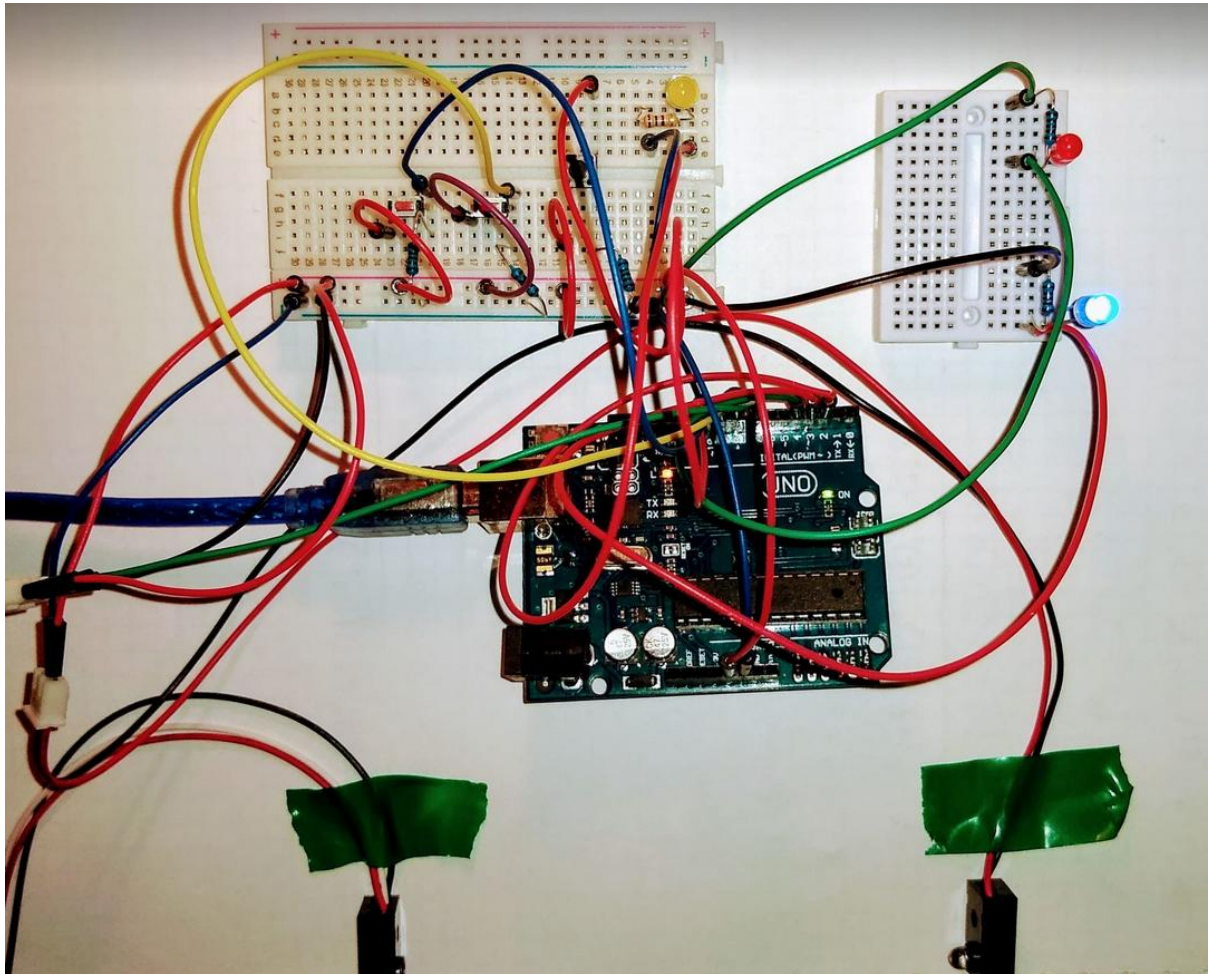
Jako element napędowy bramy został wykorzystany silnik prądu stałego Lego NXT Motor, podłączony do sterownika silnika L293D. Jest to układ scalony zawierający mostek H dzięki niemu istnieje możliwość sterowania kierunkiem obrotów silnika. Do określenia, czy brama jest otwarta, czy zamknięta zastosowano wyłączniki krańcowe z rolką oznaczone symbolem WK625. Istotnym elementem w bramie garażowej jest układ zapobiegania zamknięcia bramy, gdy człowiek lub samochód znajduje się w strefie poruszania się bramy. Do realizacji tego zadania zastosowano barierę świetlną czyli czujnik przerwania wiązki o oznaczeniu QT 50CM. Do otwarcia bramy służy pilot generujący sygnał w zakresie podczerwieni, który odbierany jest przez dedykowane urządzenie znajdujące się w widocznym miejscu. W układzie elektrycznym przewidziano również możliwość komunikacji bezprzewodowej z urządzeniami sieciowymi, na przykład w celu wysyłania komunikatów na temat awarii lub serwisu. W tym celu wykorzystany został mikrokontroler ESP32 posiadający dedykowane moduły do komunikacji bezprzewodowej. Wymiana danych pomiędzy pomiędzy Arduino oraz ESP32 odbywa się za pomocą interfejsu szeregowego UART.

Schemat rozmieszczenia czujników i napędu



Prototyp układu

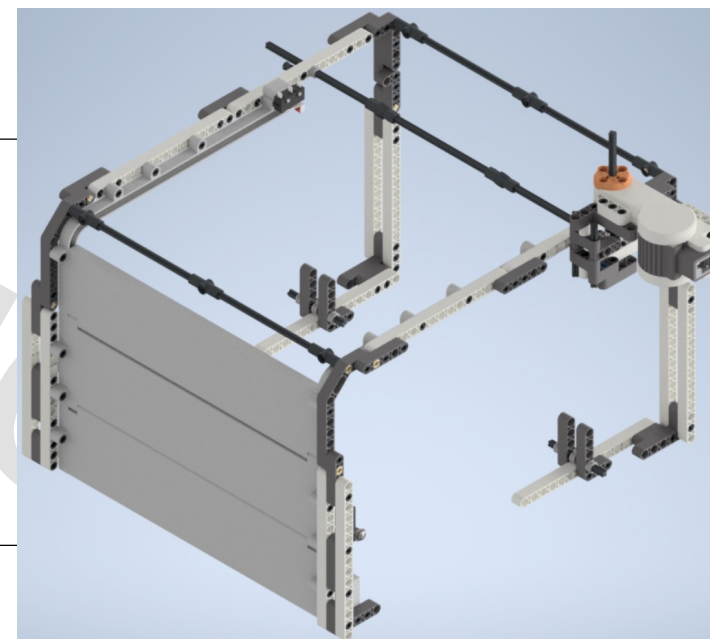
W ramach realizacji zadania stworzono układ testowy, gdzie zastąpiono brakujące podzespoły. Jako przełączniki krańcowe wykorzystano przyciski, a obroty silnika sygnalizowane są za pomocą dwóch diod. Prototyp pozwolił na przetestowanie oraz potwierdzenie poprawności stworzonego oprogramowania.



Wykonali: Gut Paweł
Jakielaszek Miłosz
Knap Paweł

Nazwa przedmiotu: Mechatronic Design

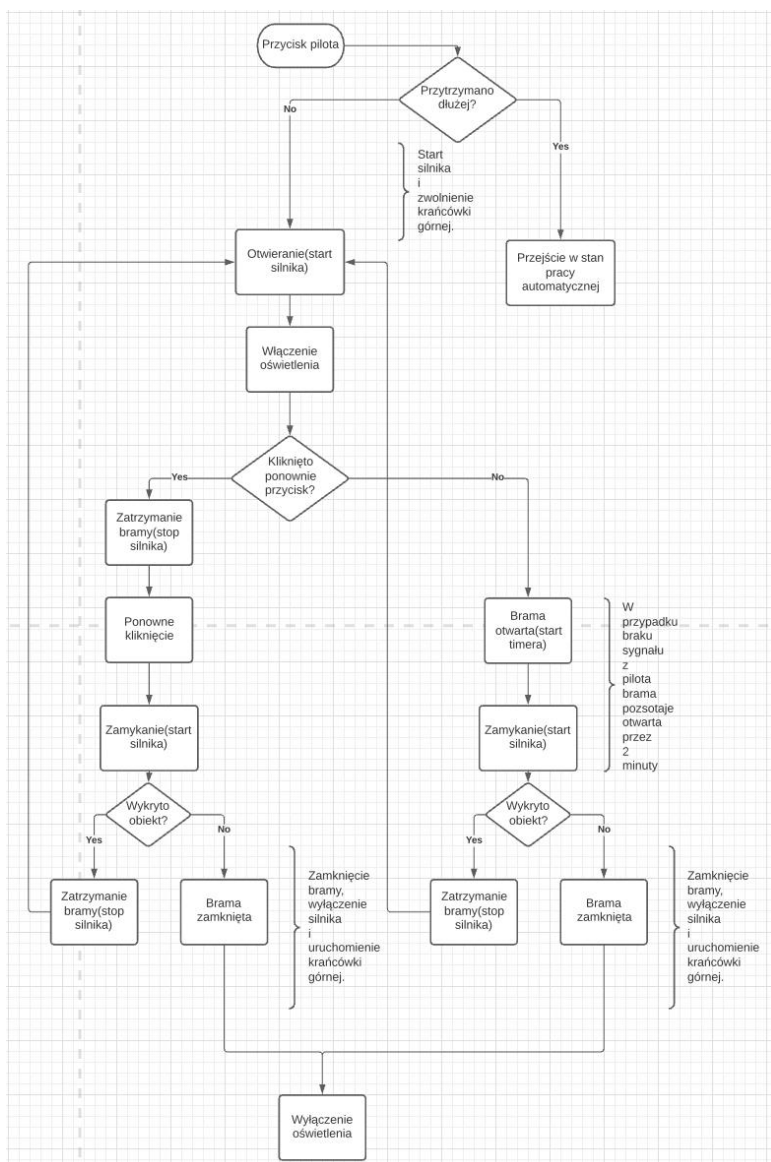
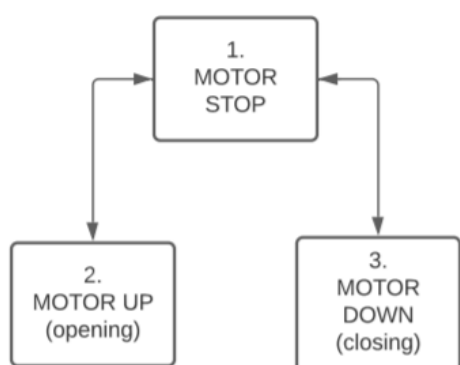
Utworzono dnia 29.11.2020



Sprawozdanie z realizacji części software projektu bramy garażowej.

Zespół software: Buda Krzysztof, Jordanek Jakub, Jurek Krzysztof

Nasza część zespołu zajęła się oprogramowaniem do bramy tworzonej przez pozostałych członków. Część software'ową podzielono na dwa etapy: podstawowe działanie i rozszerzenia(dodatkowa funkcjonalność). W pierwszym etapie skupiliśmy się nad praktycznym działaniem bramy, czyli: otwieraniem, zamykaniem, wykrywaniem przeszkód. Ta funkcjonalność zawarta została w głównej gałęzi kodu na naszym github'ie, którego link zamieszczono poniżej. Budowa programu oparta jest na maszynie stanów z wykorzystaniem instrukcji „switch case”. Na rysunku poniżej przedstawiony został ogólny diagram stanów (po lewej) oraz bardziej szczegółowy stworzony podczas planowania pracy (po prawej).



W stanie 1. silnik bramy jest zatrzymany. W zmiennej „PreviousState” przechowywany jest poprzedni stan. Po wciśnięciu przycisku w zależności od aktualnej wartości tej zmiennej program przechodzi w stan otwierania lub

zamykania bramy. Użytkownik może otworzyć bramę na stałe poprzez przytrzymanie przycisku dłużej niż sekundę. Krótkotrwałe wciśnięcie przycisku powoduje włączenie funkcji automatycznego zamykania, w tym przypadku brama zamknie po czasie zadany w zmiennej „GATE_OPENED_TIME”. W stanie 2. brama otwiera się do momentu ponownego wciśnięcia przycisku (zatrzymanie bramy w położeniu pośrednim) lub do napotkania krańcówki górnej, której sygnał powoduje powrót do stanu 1 i zatrzymanie silnika. W stanie 3. brama zamyka się do momentu napotkania krańcówki dolnej. Podobnie jak podczas otwierania możliwe jest też zatrzymanie w położeniu pośrednim poprzez wciśnięcie przycisku. Dodatkową funkcjonalnością jest wykrywanie przeszkody. Jeżeli podczas zamykania pojawi się sygnał z czujnika (przeszkoda w bramie) to program przejdzie w stan 1. a silnik zatrzyma się.

Drugi etap to dodatkowe funkcjonalności jak: kontakt z wifi, zliczanie uruchomień i informacja o serwisie oraz push notification z aktualnym stanem bramy. W tej części skorzystano z trzech dodatkowych bibliotek: „ESP8266WiFi.h, WiFiClient.h, Pushsafer.h”. Pierwsza z tych bibliotek pozwala na obsługę mikrokontrolera ESP8266, który posiada moduł WiFi niezbędny do połączenia z lokalną siecią wifi. Miejsce inicjalizacji połączenia oraz obsługi odpowiedzi otrzymanej przez mikrokontroler zaznaczone zostało w kodzie za pomocą komentarzy. Warto podkreślić, że mikrokontroler ten komunikuje się z Arduino za pomocą portu szeregowego UART. Tam też wysyłane są sformułowane w kodzie polecenia np. `Serial.println(WiFi.localIP());`. Tutaj skorzystano z polecenia dotyczącego realizacji połączenia wifi, za tą część odpowiada już biblioteka WiFiClient.h, która pozwala nam również sprawdzić dostępność adresu IP w sieci za pomocą pingu. Skorzystano możliwości pingu, aby spingować adres IP telefonu użytkownika, w celu sprawdzenia jego obecności przy otwarciu bramy. Ostatnia z wymienionych bibliotek zajmuje się wysyłaniem powiadomień typu push notification. Jej funkcjonalność została zastosowana w projekcie, aby wysłać na zdefiniowany w kodzie adres IP informację na temat wymaganej konserwacji oraz otwarcia bramy pod nieobecność właściciela. Podsumowując, moduły te zapewniają użytkownikowi kontrolę nad stanem bramy. Przypominają mu o niezbędnej konserwacji oraz informują gdy brama została otwarta, a adres IP jego telefonu nie znajduje się w sieci, w której jest mikrokontroler bramy.

Całość kodu znajduje się pod adresem: „<https://github.com/jordus6/Brama-software>”. Pod koniec pracy dodano jeszcze jedną gałąź z kodem, która odpowiada za odczytywanie informacji o naciśnięciu pilota otwierającego bramę. W tym celu użyto odbiornika podczerwieni TSOP31236 - 36 kHz. Umożliwia on odbieranie sygnałów np.

z pilota od telewizora, który mógłby symulować pilot od bramy. Do obsługi odbiornika wykorzystano bibliotekę IRremote.h. Daje ona możliwość zainicjalizowania komunikacji oraz odczytu odebranego kodu. W momencie, gdy wykryty zostaje sygnał oraz odczytany kod jest równy zdefiniowanemu w programie, brama reaguje w taki sposób jak na przyciśnięcie przycisku. Program w tej gałęzi kodu działa tak jak program w gałęzi podstawowej, jedynie przycisk pilota jest obsługiwany zdalnie.