

**Politechnika  
Warszawska**

**Robotyka i Automatyka  
STEROWNIK ZASUWY ZAMKA ROM-5**

**Wojciech Gajda 304494**

Promotor: dr hab. inż. Paweł Malczyk, prof. uczelni  
Data oddania: **18 lipca 2021**

# Spis treści

<b>1 Wstęp</b>	<b>4</b>
<b>2 Budowa zamka:</b>	<b>5</b>
2.1 Sekcja mechaniczna . . . . .	5
2.1.1 Zamek ROM-5 . . . . .	5
2.1.2 Opis sterownika . . . . .	6
2.1.3 Rzeczywisty wygląd sterownika z zamkiem . . . . .	7
2.2 Sekcja elektroniczna . . . . .	8
2.2.1 Wymagania stawiane sekcji elektronicznej . . . . .	8
2.2.2 Opis układu elektronicznego . . . . .	8
2.2.3 Mikrokontroler . . . . .	8
2.2.4 Schemat elektryczny sterownika . . . . .	8
2.2.5 Płytnica drukowana . . . . .	10
2.2.6 Mostek H . . . . .	11
2.2.7 Pomiar amperometryczny . . . . .	11
2.2.8 Obsługa portów wejściowych i wyjściowych . . . . .	12
2.2.9 Podkładki zasilania . . . . .	13
2.3 Sekcja programowa . . . . .	14
2.3.1 Wymagania stawiane sekcji programowej . . . . .	14
2.3.2 Model ideowy programu . . . . .	14
2.3.3 Warunek stopu . . . . .	15
2.3.4 Zabezpieczenie czasowe . . . . .	15
2.3.5 Debouncing - filtracja sygnałów wejściowych . . . . .	16
2.3.6 Sygnalizacja stanu . . . . .	16
<b>3 Instalacja zamka</b>	<b>17</b>
3.1 Rzeczywisty wygląd sterownika w miejscu montażu . . . . .	19
<b>4 Podsumowanie</b>	<b>20</b>

## Spis rysunków

1	Zamek ROM-5 . . . . .	5
2	Schemat zamka . . . . .	5
3	Schemat sterownika zamka . . . . .	6
4	Rzeczywiste zdjęcia sterownika . . . . .	7
5	Schemat elektryczny sterownika . . . . .	8
6	Schemat płytka drukowanej . . . . .	10
7	Wytrawione płytki . . . . .	10
8	Zlutowana płytka w obudowie . . . . .	10
9	Schemat mostka H wraz z L298 . . . . .	11
10	Przekładnik prądowo-napięciowy . . . . .	12
11	Porty wejścia i wyjścia . . . . .	12
12	Stabilizator liniowy 5V . . . . .	13
13	Zabezpieczenie układu zasilania . . . . .	13
14	Schemat blokowy programu . . . . .	14
15	Schemat ideowy montażu zamka . . . . .	17
16	Schemat ideowy montażu zamka ze sterownikiem . . . . .	18
17	Podłącznie do płytki drukowanej . . . . .	18
18	Drzwi ze zamontowanym sterownikiem . . . . .	19
19	Zbliżenie na sterownik . . . . .	20

## 1 Wstęp

Obecnie rynek urządzeń Smart-Home rozwija się bardzo prężnie i zaczyna dotyczyć niemal każdej dziedziny życia. Nowo powstałe urządzenia uwalniają nas od przeróżnych czynności, sprowadzając dla przykładu otwarcie drzwi do kliknięcia odpowiedniego przycisku w aplikacji. Co więcej, dzięki ciągłemu połączeniu z siecią nie ma potrzeby znajdowania się w pobliżu drzwi w celu ich otwarcia. Takie powiązanie zamka z internetem niesie co prawda pewne bezpieczeństwo, sprawiając, że drzwi strzegące naszego miru domowego stają się podatne nie tylko na atak wytrychem, ale również na cyber-włamanie. Nie mniej jednak integracja urządzeń codziennego użytku z siecią stanowi nieuchronne następstwo rozwoju idei Smart-Home.

Problemem obecnie dostępnych rozwiązań jest fakt, że kierowane są one dla osób w pełni zdrowych, aby ułatwić czynność które bez wiekszego problemu można wykonać ręcznie. Sprawa ma się jednak nieco inaczej w przypadku osób niepełnosprawnych, z ograniczonymi możliwościami ruchowymi. Potencjał tych urządzeń jako narzędzi służących ludziom chorym jest bardzo duży i może uprościć wiele codziannych czynności sprawiających im wiele problemów. Wymaga to jednak zmiany sposobu myślenia i indywidualnego podejścia do danego schorzenia. Trudno szukać tutaj uniwersalnego rozwiązania...

Motywacją do stworzenia sterownika zasuwy zamka była chęć ułatwienia otwarcia drzwi osobie niepełnosprawnej w stopniu znacznym, nieposiadającej pełnej władzy w rekach nogach i poruszającej się na wózku inwalidzkim. Dotychczas zamek obsługiwany był przy pomocy sprawniejszej stopy, co wymagało podjechania bezpośrednio pod drzwi. Dzięki sterownikowi możliwa stała się obsługa zamka z poziomu aplikacji lub przyciskiem zamontowanym w dogodnym miejscu. Chory nie został również pozbawiony możliwości ręcznego otwarcia zamka, fabryczną rączkę przedłużono i zakończono wygodną gałką co umożliwia dalsze otwarcie zamka w przypadku zaniku zasilania lub awarii.

## 2 Budowa zamka:

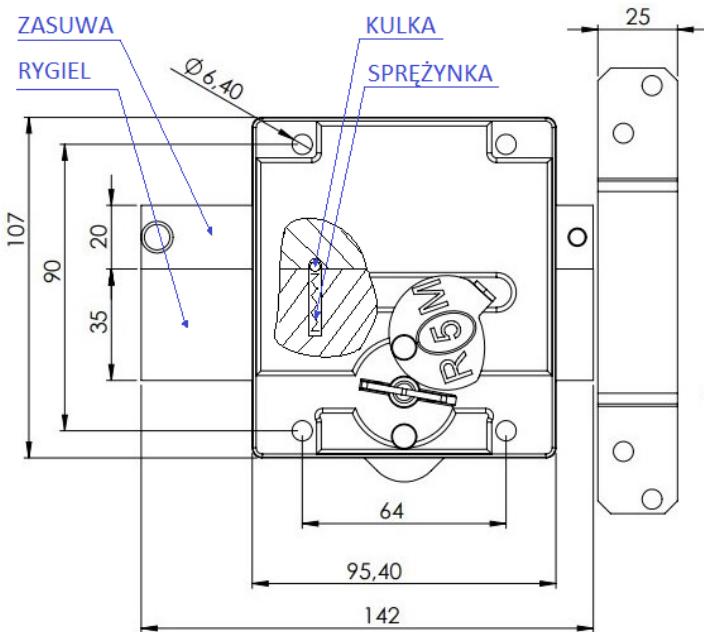
Opis budowy zamka podzielony został na trzy sekcje ze względu na dominującą dziedzinę techniki:

### 2.1 Sekcja mechaniczna

#### 2.1.1 Zamek ROM-5



Rysunek 1: Zamek ROM-5

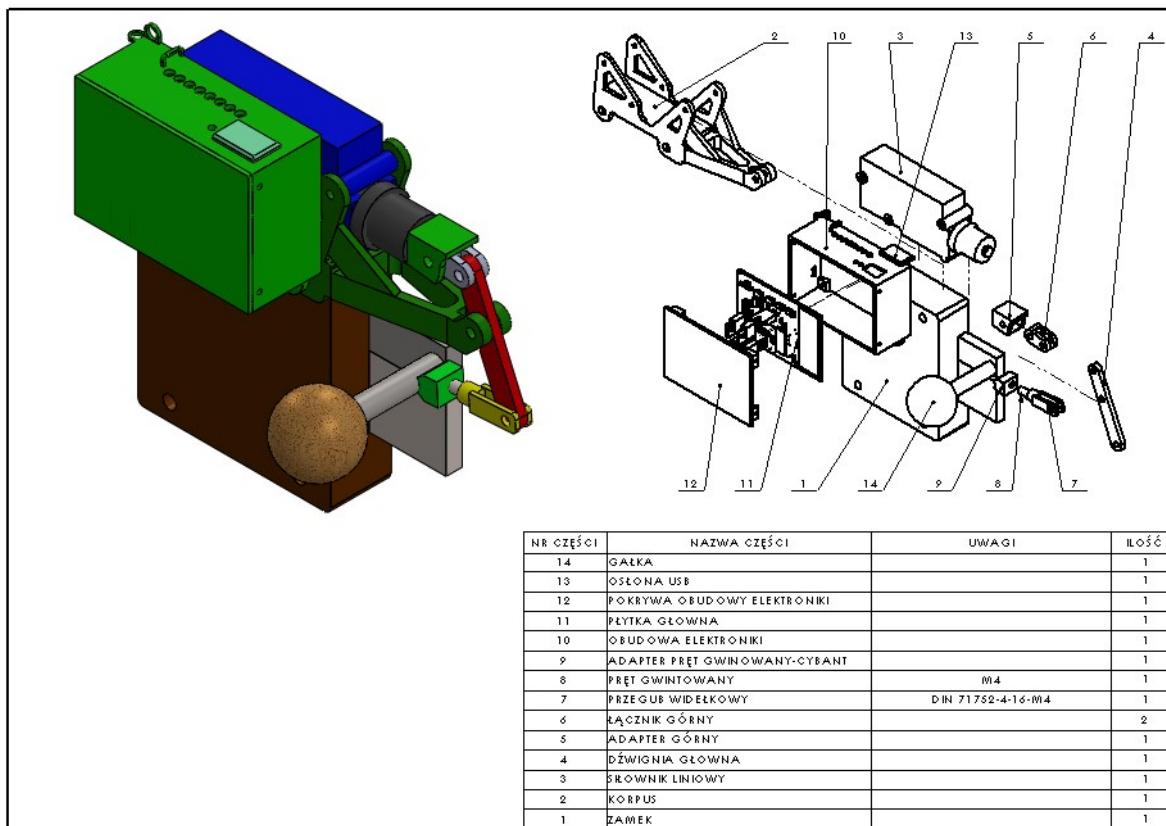


Rysunek 2: Schemat zamka

Zamek ROM-5 jest zamkiem nawierzchniowym zasuwkowym z zasuwią dzieloną. Jedna z części zasuwy jest częściowo swobodna co pozwala na ręczne zamknięcie drzwi. Zamek jest symetryczny - może być stosowany w przypadku drzwi lewych lub prawych. W omawianym przypadku zamek znajduje się na drzwiach prawych, toteż uchwyt przeniesiony jest na drugą stronę. Obrócenie kluczem pozwala bezpośrednio na przesunięcie dolnego rygla. Połączenie zasuwy i rygla zapewnia kulka docisnieta sprężynką. Zasuwe można zatem rozłączyć z głównym ryglem używając odpowiednio dużej siły. Aby przesunąć zasuwe przy pomocy klucza należy najpierw zrównać rygiel i zasuwę. Przesunięcie rygla bez klucza nie jest natomiast możliwe.

Dzięki zastosowaniu gotowego zamka unikamy problemu związanego z bezpieczeństwem – wytrzymałość zamka potwierdzona jest atestem producenta. Sterownik będący tematem tej pracy jest nieinwazyjną nakładką na już istniejący zamek, pozwalającą na przesuw wspomnianej wyżej zasuwy.

## 2.1.2 Opis sterownika



Rysunek 3: Schemat sterownika zamka

Głównym elementem wykonawczym sterownika jest siłownik liniowy stosowany w samochodowych zamkach centralnych. Siłownik pozwala na przesuw w zakresie 0-19mm z siłą do 80N, posiada wbudowane sprzęgło odśrodkowe i wyłączniki krańcowe.

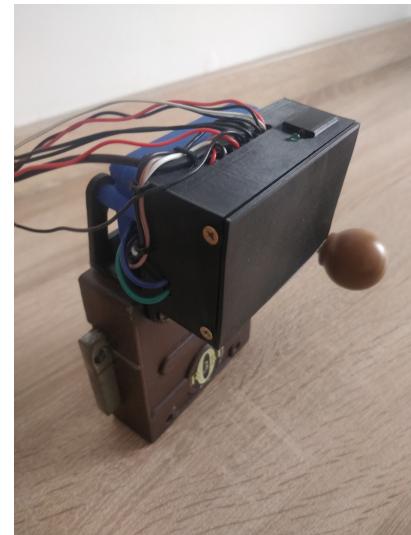
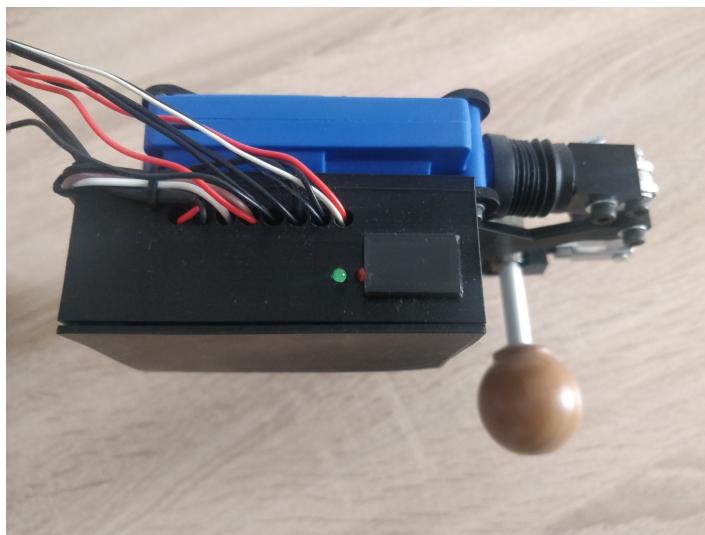
Mechanizm oparty jest na zasadzie dźwignii dwustronnej. Punkt podparcia zapewnia wysięgnik zintegrowany z korpusem. Ze względu na skok zamka wynoszący 28mm główna dźwignia nie jest symetryczna i stosunek jej ramion wynosi 3:5. Początkowo wydrukowana dźwignia, została zastąpiona elementem wyciętym ze stalowego płaskownika o gr. 4mm. Siłownik połączony jest z dźwignią poprzez wydrukowany adapter i dwa łączniki wycięte z alumionowego płaskownika o gr. 3mm. Zasuwa połączona jest z dźwignią poprzez opaskę ślimakową, wydrukowany adapter, pręt gwintowany umożliwiający regulację i przegub widełkowy DIN 71752-4-16-M4.

Z przodu siłownika umieszczona została płytka PCB stanowiąca elektroniczną część sterownika wraz z dedykowaną obudową. Zmianie uległa fabryczna dźwignia zasuwy: została wydłużona i zakończona gałką, aby ułatwić jej ręczną obsługę.

Projekt zamka został stworzony przy użyciu oprogramowania SolidWorks 2019, a elementy zaznaczone na schemacie na zielono zostały wydrukowane przy pomocy drukarki Creality Ender-3 z PLA.

### 2.1.3 Rzeczywisty wygląd sterownika z zamkiem

Cały proces projektowania i testowania oparty był na zamku ROM-3. Zamek ROM-3 zachowuje wymiary gabarytowe zamka ROM-5, a ze względu na mniejszą liczbę atestów jest znacznie tańszy od ROM-5



Rysunek 4: Rzeczywiste zdjęcia sterownika

## 2.2 Sekcja elektroniczna

### 2.2.1 Wymagania stawiane sekcji elektronicznej

Do najważniejszych parametrów i wymagań stawianych elektronice urządzenia zalicza się:

- całość zasilana napięciem stałym 12V z zasilacza sieciowego o odpowiedniej wydajności
- wysterowanie silnika DC (zmiana kierunku obrotu poprzez zmianę polaryzacji) o stosunkowo dużym zapotrzebowaniu (12V, prąd szczytowy deklarowany przez producenta to 5A, w praktyce nie przekracza 3A)
- sterowanie zamkiem przy pomocy przycisku dzwonkowego, styk bezpotencjałowy NO
- odczyt stanu wyłączników krańcowych (krańcówek) i kontraktonu położenia drzwi
- zapewnienie odpowiedniego poziomu bezpieczeństwa układu i sygnalizacja stanu zamka.

### 2.2.2 Opis układu elektronicznego

Głównym elementem sterownika jest mikrokontroler STM32 w wersji F303K8 zawarty na płytce rozwojowej zgodnej z Arduino Nano. Sterowanie silnikiem odbywa się za pośrednictwem mostka H. Pomiędzy mikrokontrolerem, a mostkiem umieszczony został układ L298N stanowiący 4 półmostki. Pozwala on zmianę poziomu logicznego i efektywne wysterowanie tranzystorów mostka. Cały układ zabezpieczony jest bezpiecznikiem polimerowym i diodą zabezpieczającą przed podłączeniem układu z odwrotną polaryzacją. Poprzez wyciągnięcie odpowiedniej zworki, prąd płynący przez mostek może zostać przepuszczony przez układ ACS712 stanowiący przekładnik prądowniapięciowy. Umożliwia to pomiar amperometryczny z poziomu mikrokontrolera. Wejścia sygnałowe pracują domyślnie w trybie pull-up, a wejście przycisku zostało dodatkowo zabezpieczone przez 2 diody Zenera. Cały układ został zawarty na płytce drukowanej i umieszczony w wydrukowanej obudowie.

### 2.2.3 Mikrokontroler

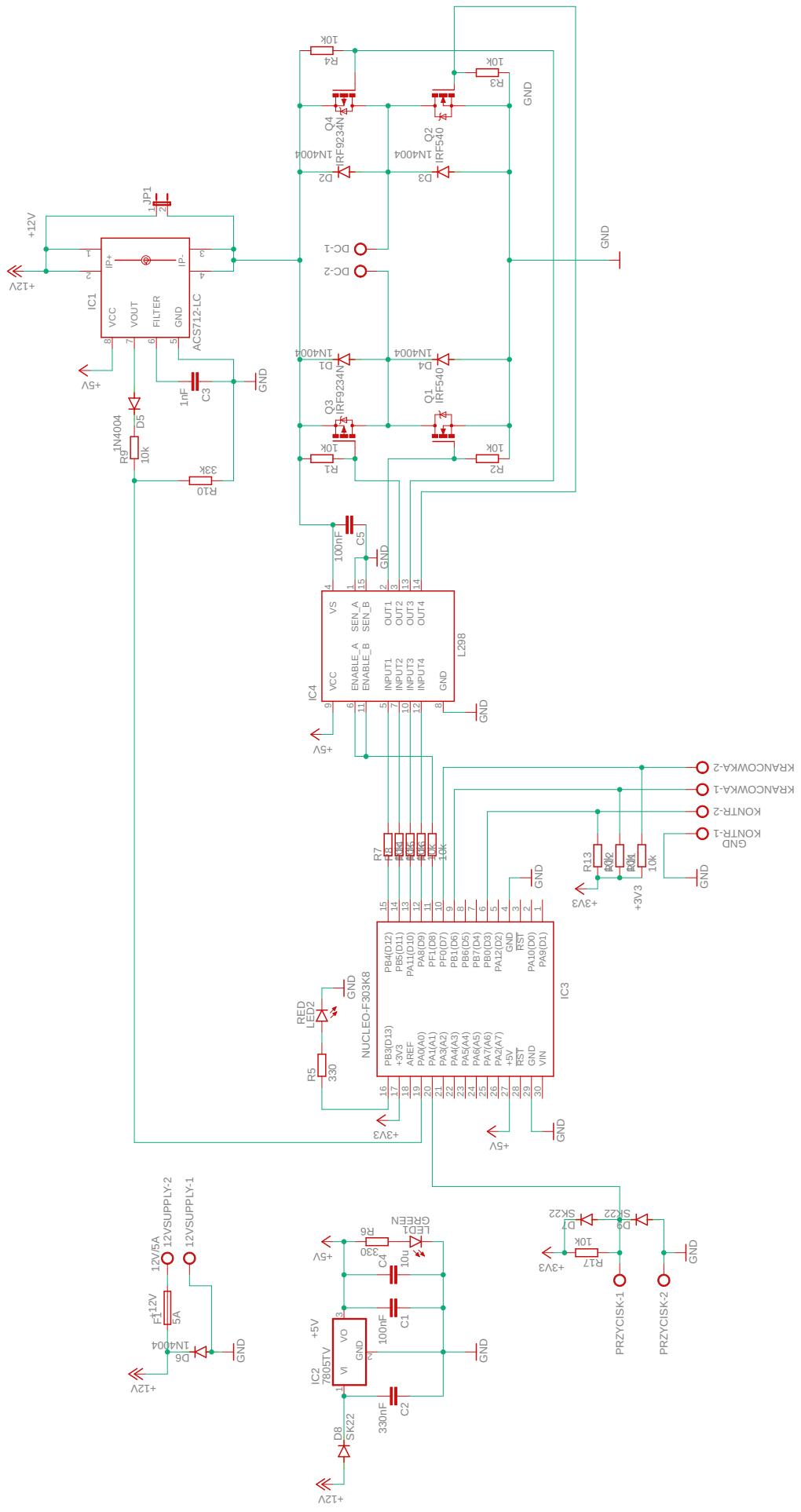
Wybrany mikrokontroler pracuje w takt zegara 8MHz oraz jego moc obliczeniowa znacznie przewyższa potrzeby tego projektu. Płytką rozwojową zapewnia wszystkie elementy pasywne niezbędne do pracy mikrokontrolera, wbudowany debugger i stabilizator napięcia. Wybór konkretnie modelu F303K8 podyktowany był głównie ceną, niewielkimi wymiarami i dużą liczbą portów I/O z obsługą PWM w porównaniu do innych płyt w standardzie Arduino Nano.

Ze względu na zasadniczość układu linią 5V zamiast przez port USB, wbudowany debugger nie uruchamia się co blokuje rozruch układu. Do poprawnej pracy należy odlutować odpowiednią zworę.

### 2.2.4 Schemat elektryczny sterownika

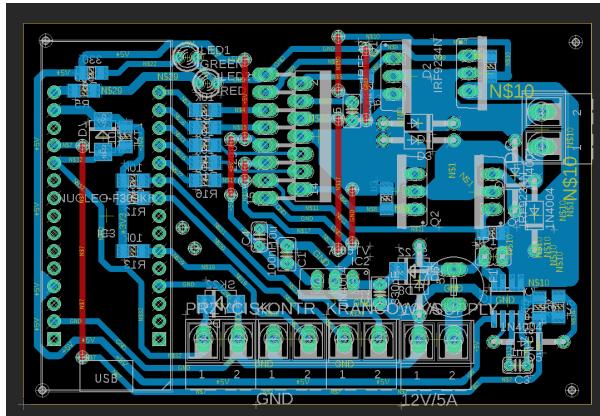
Na następnej stronie znajduje się schemat elektryczny sterownika. Do zaprojektowania układu elektronicznego i płytka drukowanej skorzystałem z programu EAGLE w wersji ograniczonej, przeznaczonej dla hobbistów.

Rysunek 5: Schemat elektryczny sterownika

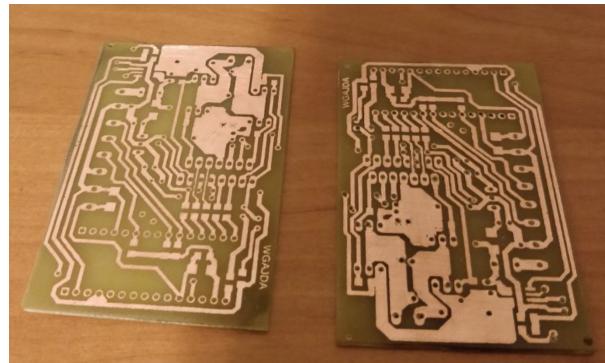


### 2.2.5 Płytki drukowane

Przedstawiony powyżej układ został przeniesiony na płytę drukowaną. Schemat płytki prezentuje poniższy rysunek:

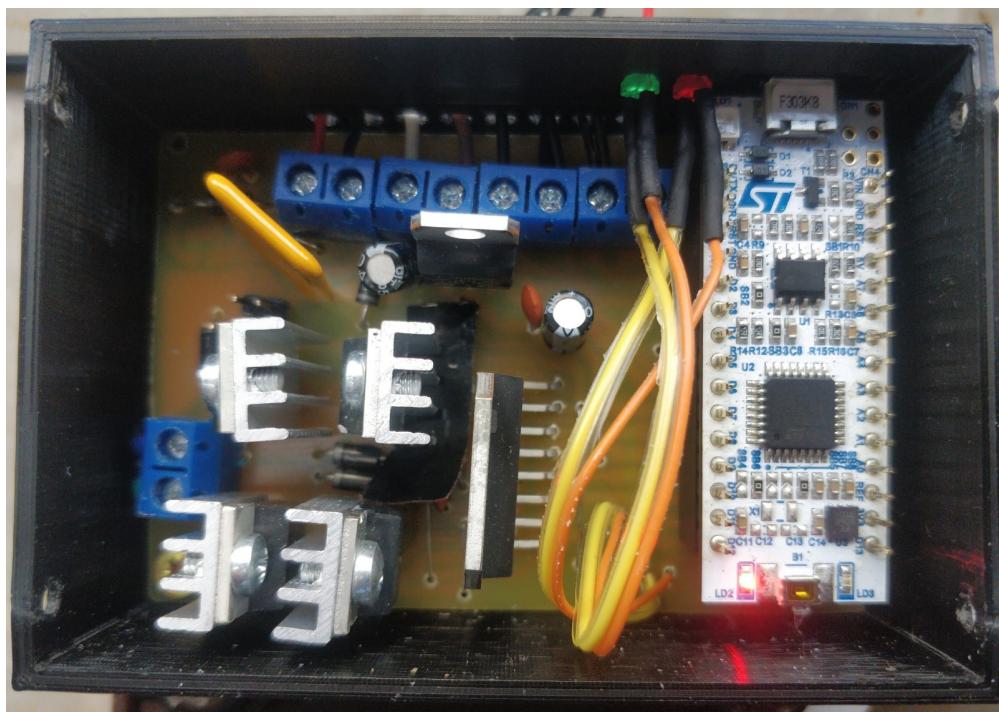


Rysunek 6: Schemat płytki drukowanej



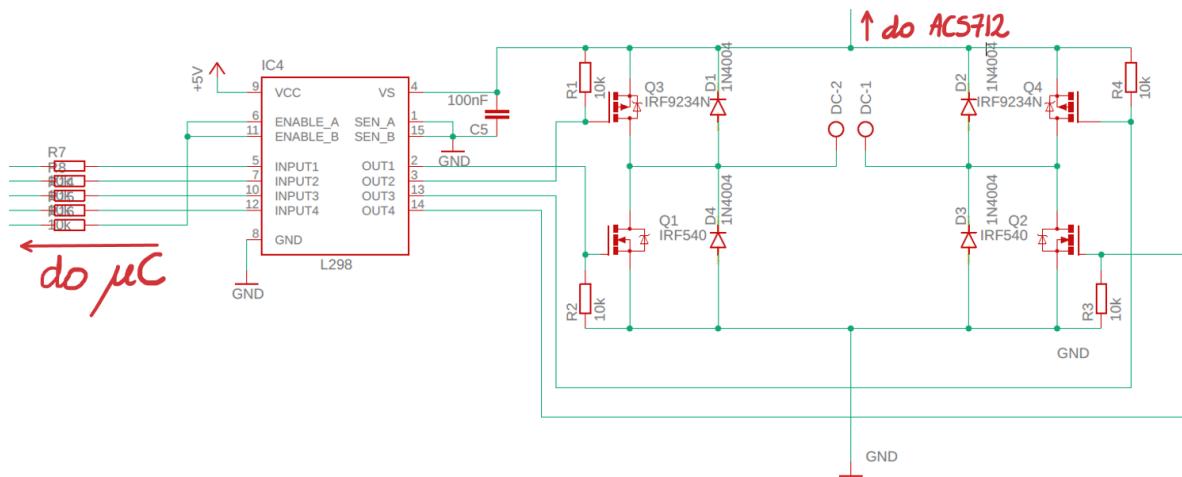
Rysunek 7: Wytrawione płytki

Płytki jest jednowarstwowa, na wierzchniej warstwie poprowadzone jest jedynie kilka zworek oznaczonych kolorem czerwonym. Płytki (wytrawiłem od razu 3 sztuki, z których jedna uległa zniszczeniu) wykonane zostały metodą termotransferu i wytrawione w domowych warunkach.



Rysunek 8: Zlutowana płytka w obudowie

## 2.2.6 Mostek H



Rysunek 9: Schemat mostka H wraz z L298

Mostek H pozwala na sterowanie prędkością i kierunkiem obrotów silnika DC napędzającego silownik. W układzie zostały użyte tranzystory MOSFET: typu P IRF9Z23N i typu N IRF540. Według dokumentacji pozwalają one przepływ stałego prądu o natężeniu odpowiednio 19A i 28A, oraz maksymalnemu napięciu źródła/dren 55V i 100V. Oznacza to znaczny zapas względem zapotrzebowania i po nieznacznych zmianach pozwala na prace układu z większym silnikiem lub silnikami 24V, wykorzystywanymi powszechnie w napędach bram garażowych.

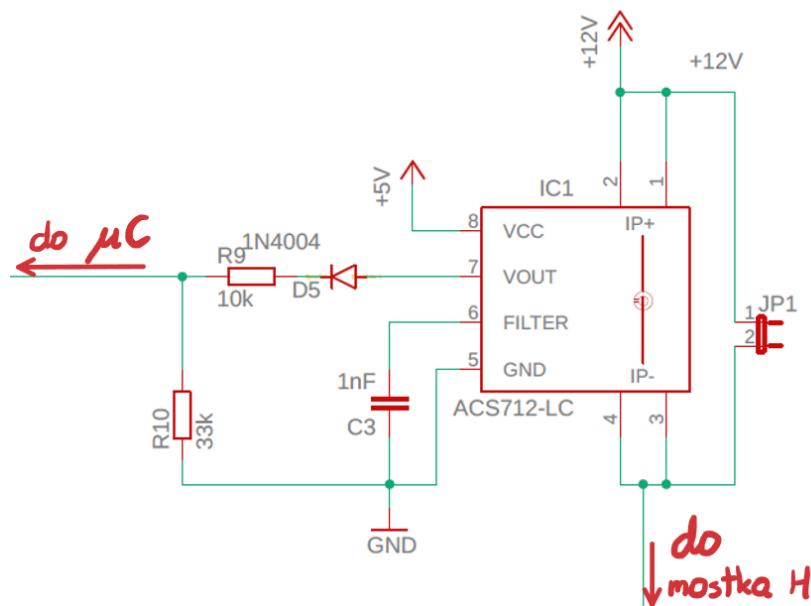
Diody D1-D4 mają na celu odciążyć diody zawarte w mosfetach, a oporniki R1-R4 powodują domyślne wyłączenie mosfetów gdy nie są one wysterowane. Znajdujący się pomiędzy mikrokontrolerem a mostkiem układ L298 składa się z czterech połmostków co pozwala wydajnie doprowadzić i odprowadzić ładunek z bramek MOSFET'ów. Dodatkowe wejścia ENABLE pozwalają w całości wyłączyć układ. Ma to zastosowanie w czasie inicjalizacji układu, spowodowanej np. zaniem zasialania. Cały układ L298 pozostaje wyłączony do czasu pełnego skonfigurowania pinów mikrokontrolera.

## 2.2.7 Pomiar amperometryczny

Celem pomiaru amperometrycznego jest wykrycie nagłego wzrostu natężenia prądu płynącego przez silnik. Pozwala to na wykrycie zablokowania wału silnika lub jego uszkodzenia.

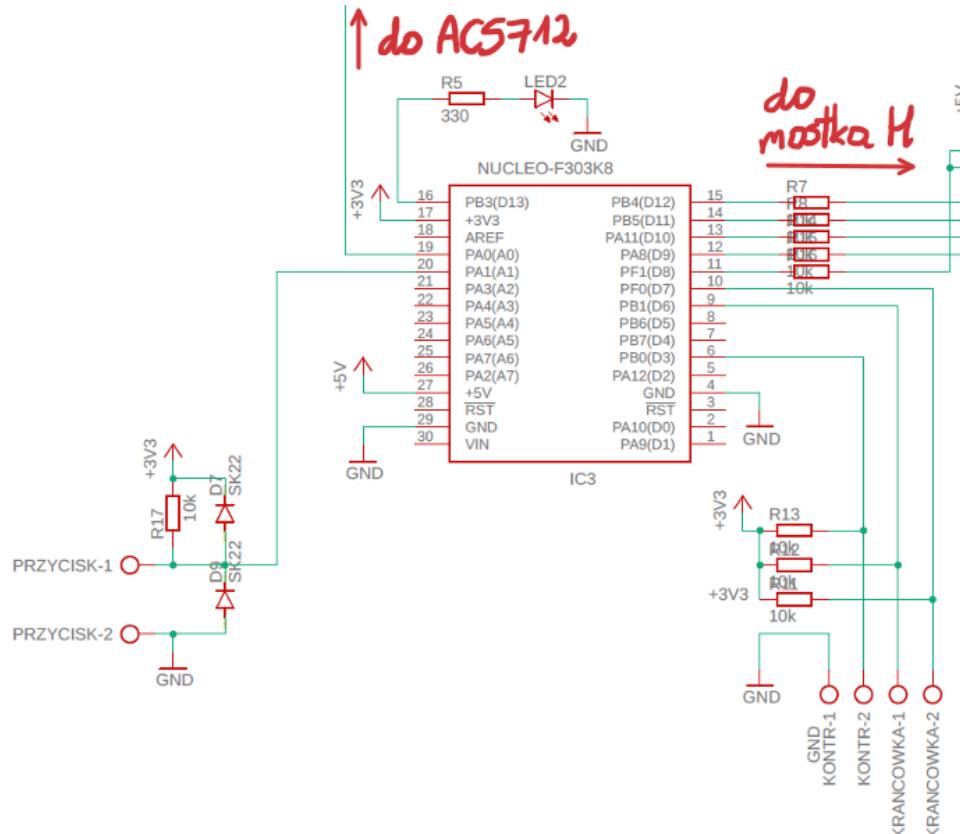
Szeregowo do mostka H włączony został układ pomiarowy. Pomiar może zostać pominięty poprzez włączanie zworki JP1. Układ ACS712 wykorzystuje efekt Halla do pomiaru napięcia w symetrycznym względem zera zakresie do 5A, 10A lub 25A w zależności od wersji. (np. w wersji 5A:  $5A \Leftrightarrow 5V$ ,  $0A \Leftrightarrow 2.5V$  i  $-5A \Leftrightarrow 0V$  na wyjściu VOUT). Podany w dokumentacji układu przykład użycia diody D5 i dzielnika napięcia opartego na R9 i R10 pozwala na pomiar prądu płynącego w kierunku zgodnym z polaryzacją układu i dostosowanie poziomu logicznego z 5V do 3.3V.

*Wykrywanie przeszkode nie jest niezbędnie w tym projekcie. Pomiar oscyloskopem wykazał, że układ pracuje prawidłowo jednak jego obsługa nie została zaimplementowana programowo.*



Rysunek 10: Przykładnik prądowo-napięciowy

### 2.2.8 Obsługa portów wejściowych i wyjściowych



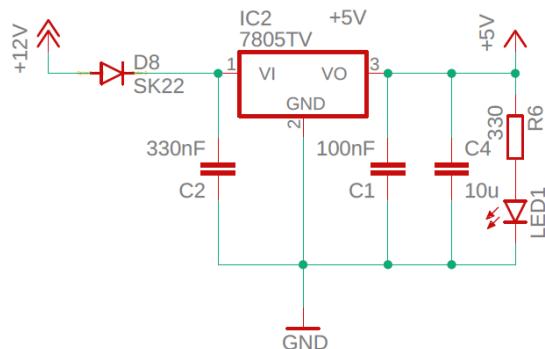
Rysunek 11: Porty wejścia i wyjścia

Oprócz wymienionych wcześniej podukładów do mikrokontrolera podłączone są jeszcze:

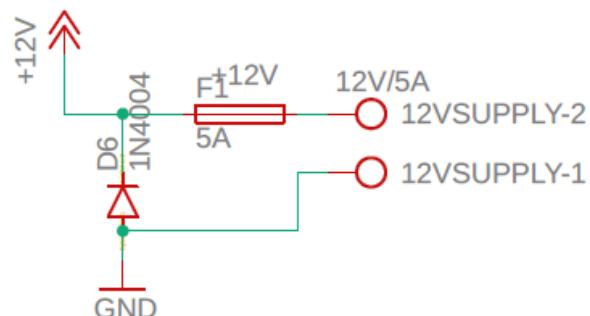
- wyłączniki krańcowe - piny w trybie pull-up
- kontrakton sprawdzający stan zamknięcia skrzydła drzwi - pin w trybie pull-up
- wejście sygnału przycisku - pin pull-up z diodami zabezpieczającymi
- wyjście sygnalizacyjne: zielona dioda LED.

### 2.2.9 Podukłady zasilania

Mikrokontroler i układ ACS712 zasilane są napięciem 5V. Mikrokontroler posiada co prawda wewnętrzny stabilizator i pracuje przy napięciu 3.3V jednak największą wydajność płytki rozwojowej uzyskujemy zasilając go linią 5V. Do obniżenia napięcia z 12V na 5V wykorzystany został stabilizator liniowy LM7805 wraz z odpowiednimi kondensatorami filtrującymi. Układ pozbawiony radiatora w tej konfiguracji jest przystosowany do ciągłego przepływu prądu o natężeniu 285mA i impulsu dochodzącego do 2.2A. Pomiar pokazał, że w trakcie pracy całkowity prąd pobierany z linii 5V oscyluje w okolicy wartości 50mA. Powszechnie wiadomo, że stabilizatory liniowe charakteryzują się niższą wydajnością niż np. przetwornice step-down jednak w tym projekcie wydajność stabilizatora jest wystarczająca, zapewnia prostotę konstrukcji oraz pozwala uniknąć problemu zakłóceń pochodzących od przetwornicy. Bezpośrednio do stabilizatora podpięta jest czerwona dioda sygnalizująca stan zasilania układu.



Rysunek 12: Stabilizator liniowy 5V



Rysunek 13: Zabezpieczenie układu zasilania

Rysunek po prawie stronie przedstawia sposób zabezpieczenia układu przed przeciążeniem zasilacza i odwrotną polaryzacją. Zasilacz sieciowy podłączony jest do złącza ARK oznaczonego na schemacie jako 12VSUPPLY. Pomiedzy zasilaczem a układem umieszczony został bezpiecznik polimerowy 30V/5A i dioda D6 pozwalająca zminimalizować ryzyko uszkodzenia sterownika przy podłączeniu zasilacza w odwrotnej polaryzacji

## 2.3 Sekcja programowa

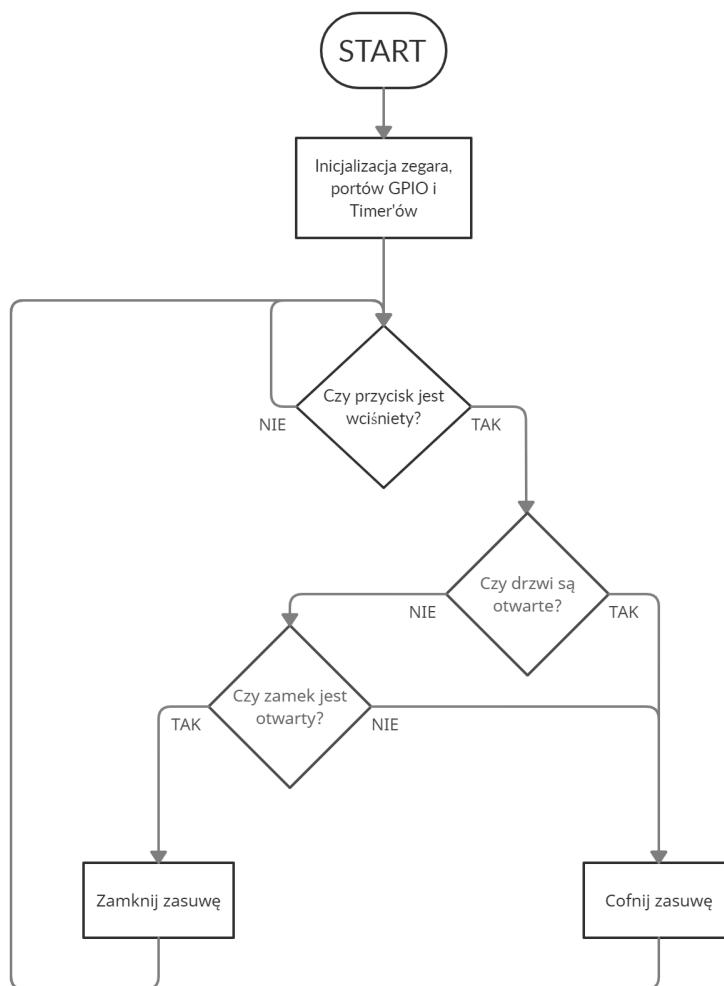
### 2.3.1 Wymagania stawiane sekcji programowej

Do najważniejszy wymagań stawianych programowi urządzenia zalicza się:

- przechodzenie do odpowiednich stanów w zależności od stanu wejść (opisane w następnym podrozdziale)
- odpowiednie załączanie kluczy tranzystorowych
- odfiltrowanie zakłopeń portów wejściowych
- sygnalizacja stanu pracy programu
- zabezpieczenie układu przed awarią lub przeciążeniem.

### 2.3.2 Model ideowy programu

W znacznym uproszczeniu program można przedstawić następującym schematem blokowym:



Rysunek 14: Schemat blokowy programu

Podczas tworzenia kolejnych stanów programu przyjęto następujący model bezpieczeństwa:

*W momencie wcisnięcia przycisku (bezpośrednio, bądź zdalnie) osoba upoważniona znajduje w pobliżu drzwi. W razie napotkania jakiegokolwiek błędu, otwórz zamek.*

Program podejmuję więc akcję sprawdzając czy wystąpiło odpowiednie zdarzenie. Można zapisać to w postaci tabeli gdzie pierwszy wiersz oznacza najwyższy priorytet:

Zdarzenie	Opis	Akcja
Kontrakton drzwi wyłączony	Skrzydło drzwi niedomkniete lub uszkodzony kontrakton	Otwórz zasuwę
Krańcówka otwarcia wyłączona	Zamek zamknity lub w nieznanej pozycji lub krancówka uszkodzona	Otwórz zasuwę
Krańcówka otwarcia włączona i Kontraktor drzwi włączony	Zasuwa cofnięta i drzwi domknięte	Zamknij zasuwę

### 2.3.3 Warunek stopu

Powyższy podrozdział dotyczył uruchomienia silnika DC z odpowiednią polaryzacją. Pozostawienie silnika w stanie uruchomionym spowodowałoby dotarcie do zadanej pozycji i uszkodzenie sterownika. W związku z tym należy wyłączyć silnik niedługo po osiągnięciu przez niego zamierzonego celu. Silnik powinien się również wyłączyć nawet jeżeli wyłącznik krańcowy uległ awarii.

W czasie poprawnej pracy zamka wyłączenie silnika następuje w przeciągu 200ms od zarejestowania aktywacji odpowiedniego wyłącznika krańcowego. W przypadku zastosowanego siłownika załączenie krańcówki następuje już w odległości ok. 1/3 całkowitego wysuwu od końca zakresu. Dodatkowy czas pozwala na całkowite wysunięcie lub wsunięcie tloczka.

### 2.3.4 Zabezpieczenie czasowe

W czasie pracy zamka może dojść do sytuacji w której podstawowy warunek stopu nie zostanie osiągnięty. Aby zapobiec zniszczeniu układu całkowity czas przez który silnik może być zasilany wynosi 1000ms. Po tym czasie, niezależnie od stanu programu, mostek H przełączy się w stan nieaktywny.

Warto nadmienić, że funkcje odpowiedzialne za sterowanie odpowiednimi tranzystorami nie pozwalają na realizację nowego zadania, jeżeli poprzednie nie uległo zakończeniu, tj. wszystkie klucze tranzystorowe nie zostały otwarte.

### 2.3.5 Debouncing - filtracja sygnałów wejściowych

Ze względu na mechaniczną niedoskonałość wyłączników krańcowych i przycisku, sygnał odbierany przez może być zakłócony i zawierać krótkotrwałe impulsy. Wiąże się to z istnieniem zjawiska drżenia styków (z ang. bounce). Debouncing ma na celu wykrycie takich drgań i zarejestrowanie wciśnięcia przycisku jako pojedyńczego przejścia pomiędzy stanami.

Każdy z portów wejściowych próbkowany jest z częstotliwością 1000Hz. Zmiana stanu portu rejestrowana jest dopiero w momencie gdy 16 kolejnych próbek w przypadku przycisku, 32 w przypadku wyłączników krańcowych jest zgodnych.

Dodatkową zaletą tego rozwiązania jest fakt wyzwalania akcji zamka przez zbocze. Dłuższe przytrzymanie przycisku (np. przez zwarcie) nie powoduje ciągłej pracy zamka pomiędzy pozycjami, jedynie pojedyńczą akcję.

### 2.3.6 Sygnalizacja stanu

Ze względu na brak połączenia sterownika z siecią, wyświetlaczem lub innym urządzeniem wyjścia sygnalizacja stanu programu odbywa się poprzez zieloną diode LED, zmostkowaną z diodą użytkownika znajdującej się na płytce rozwojowej. Wykorzystane zostało tutaj rozwiązanie znajdujące zastosowanie w pojazdach produkowanych pod koniec XX w. w okresie od czasu gdy elektroniczny sterownik silnika stał się standardowym elementem samochodu do czasu rozpowszechnienia standardu OBD2 (od 2001 w UE). W tym okresie wielu producentów, których sterowniki posiadały zdolność samodiagnostyki, decydowało się na przekazywanie kodu błędu przez kontrolkę "CHECK ENGINE". Po uruchomieniu funkcji (zwykle poprzez wstawienie zworki w sterowniku) kontrolka rozpoczęła wymigawianie sekwencji sygnałów których odczytanie pozwalał na odnalezienie opisu usterki w instrukcji.

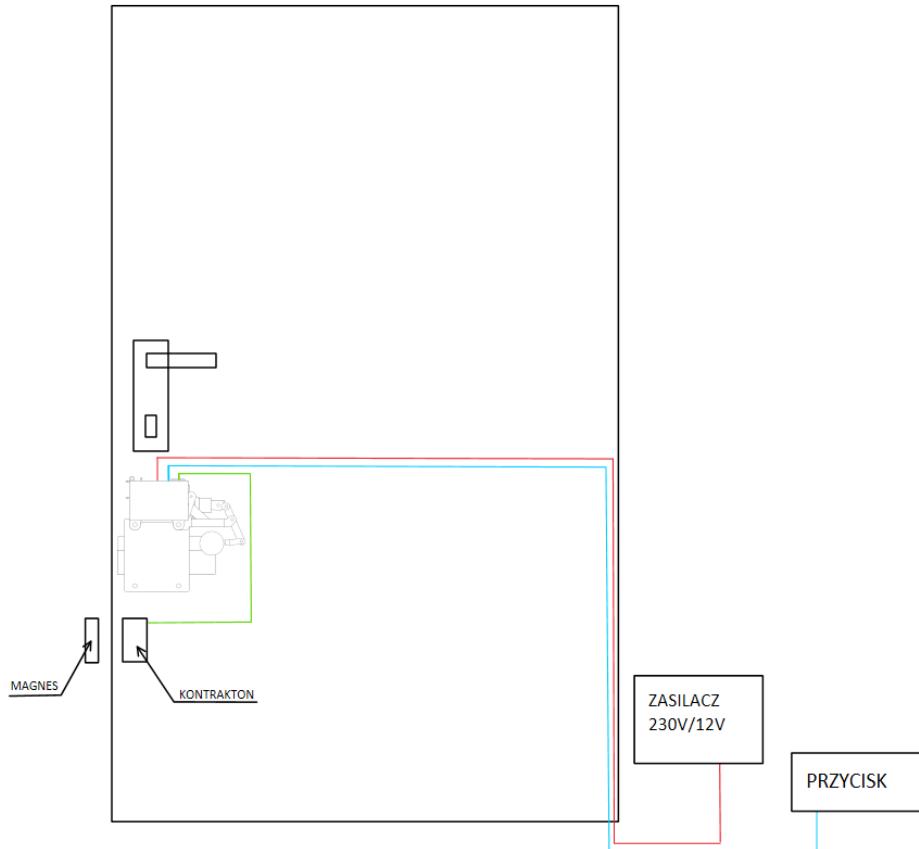
W przypadku sterownika przyjęta została następująca postawa czasu: sygnał krótki - 200ms, przerwa - 200ms, sygnał długi - 3x krótki - 600ms. Opisy zaimplementowanych sygnałów przedstawione zostały w poniższej tabeli:

Sygnal	Opis
krótki-krótki-krótki	Zakończenie inicjalizacji sterownika, gotowość
krótki	Zamykam zamek
krótki-krótki	Otwieram zamek
długi-długi	Wystąpił problem, otwieram zamek

Problem oznacza najprawdopodobniej niedomknięte drzwi lub uszkodzony kontrakton.

### 3 Instalacja zamka

Początkowo założono, że sterownik ma obsługiwać jedynie przewodowe sterowanie zamkiem. W tej konfiguracji schemat ideowy zamontowanego zamka wygląda w następujący sposób:

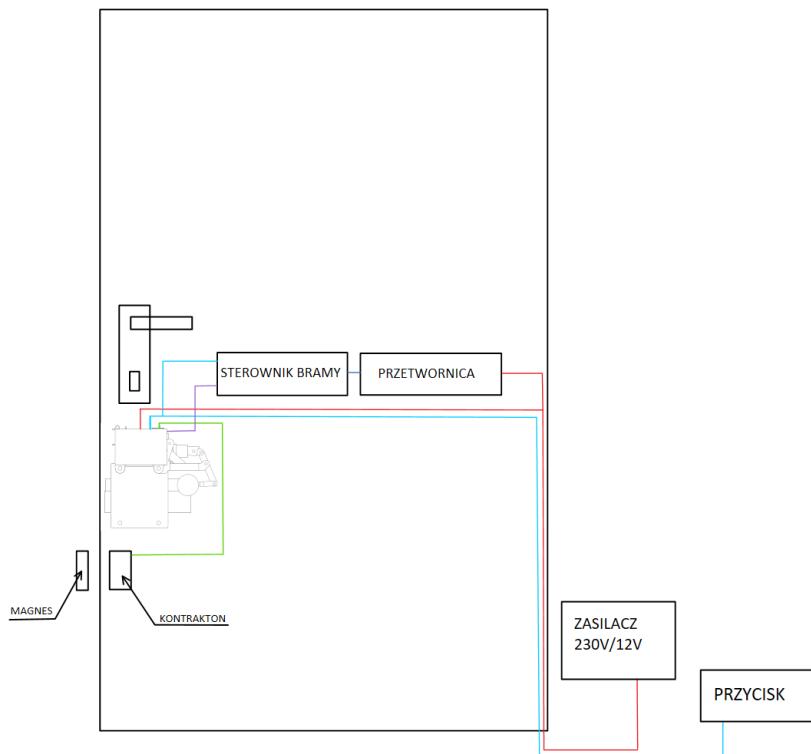


Rysunek 15: Schemat ideowy montażu zamka

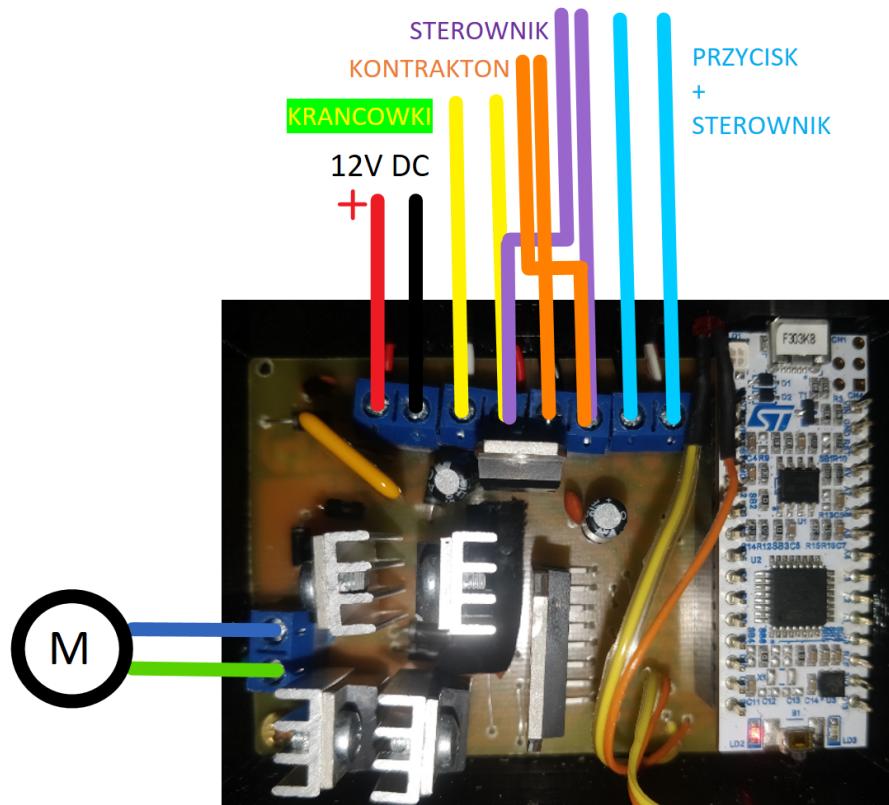
W koncowym etapie projektowania okazało się jednak, że wymagana jest również zdalna kontrola nad zamkiem. Do tego celu wykorzystane zostało urządzenie sprzedawane pod nazwą "sterownik bramy garażowej/furtki" zgodny ze standardem urządzeń już występujących w mieszkaniu.

Urządzenie zasilane z portu USB (5V/1A) posiada wyjście w postaci przekaźnika NO i wejście bezpotencjałowe do którego fabrycznie podłączony był kontraktor. Do zasilania układu skorzystamy z przetwornicy 12V→5V/1A. Układ taki można znaleźć m. in. w ładowarkach samochodowych. Wyjście z układu połączone jest równolegle do przycisku, a wejście odczytuje stan wyłącznika krańcowego aktywnego przy zamkniętym zamku. Ostatecznie schemat instalacji zamka wraz ze sterownikiem bezprzewodowym przedstawiony został na następnej stronie.

Na schemacie ideowym każde połączenie jest w rzeczywistości przewodem dwużyłowym. Poniższy rysunek przedstawia dokładne podłączenie przewodów do płytki głównej (na czas montażu zieloną diode można wysunąć z obudowy):



Rysunek 16: Schemat ideowy montażu zamka ze sterownikiem



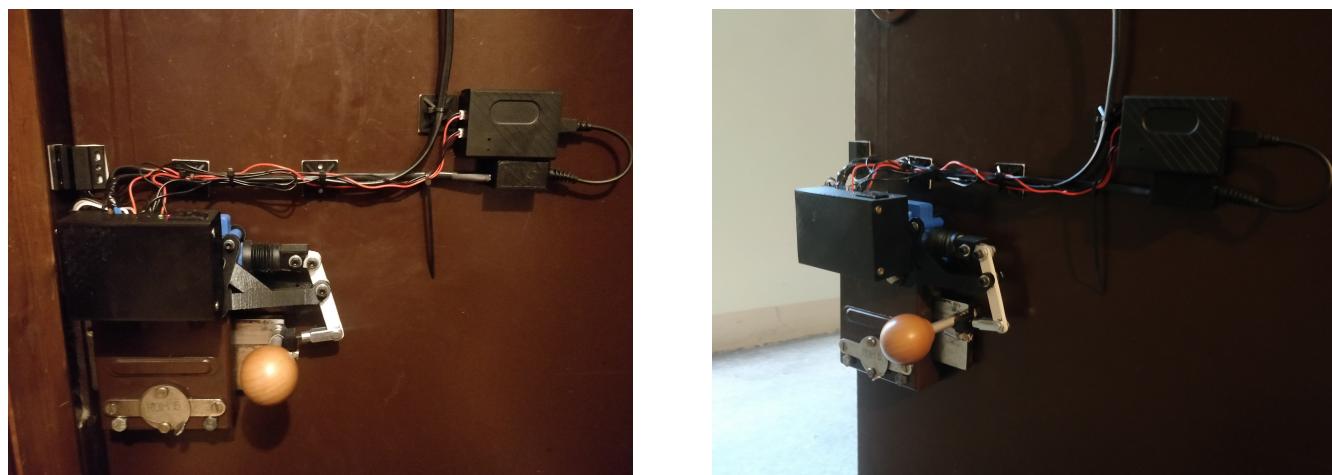
Rysunek 17: Podłączenie do płytki drukowanej

### 3.1 Rzeczywisty wygląd sterownika w miejscu montażu

W dn. 14.07.2021 sterownik zamka został zamontowany w mieszkaniu osoby niepełnosprawnej. Poniższe zdjęcia prezentują ostateczną realizację projektu:



Rysunek 18: Drzwi ze zamontowanym sterownikiem



Rysunek 19: Zbliżenie na sterownik

## 4 Podsumowanie

Powyższa praca prezentuje urządzenie, które mamy nadzieję chociaż w pełnym stopniu ułatwi otwieranie zamka drzwi osobie niepełnosprawnej. Sterownik spełnia indywidualne wymagania, którym cieźko sprostać obecnie dostępnymi rozwiązaniami komercyjnymi. Jako dodatkowy efekt tej pracy można wyekstrahować jeszcze jedno urządzenie:

Efektem pracy nad tym projektem, pomijając sekcje mechaniczną jest programowalny sterownik silnika DC o dużej mocy, kompaktowych rozmiarach i stosunkowo niskiej cenie. W przyszłości opisany sterownik można rozwinąć i dostosować np. do sterowania bramą garażową lub prostą suwnicą. Niedużym nakładem pracy można zmieniać liczbę zarządzanych silników (przez dodanie kolejnych mostków H), dostosować go do sterowania silnikiem krokowym lub zapewnić obsługę większej liczby wejść i wyjść sygnałowych. Znając wymagania dotyczące komunikacji ze sterownikiem (port szeregowy, magistrala CAN, transmisja bezprzewodowa Wi-Fi, Bluetooth lub radiowa 868MHz itd.) można dobrać tak mikrokontroler, aby zapewnić sprzętową obsługę wymaganych protokołów. Wreszcie myśląc o produkcji masowej można zrezygnować z płytka rozwojowych na rzecz czystych chipów i zapewnić im odpowiednim obsługę na płytce głównej.

Część tych funkcjonalności można osiągnąć jedynie zmianami programowymi np. regulację prędkości obrotowej sygnałem PWM, pomiar amperometryczny przez przetwornik ADC lub komunikacje szeregową przez port USB. Jednak siłownik użyty w projekcie na ogół sterowany jest zero-jedynkowo przy pomocy przekaźników. Nie ma więc potrzeby implementacji tych funkcji.