



**POLITECHNIKA ŚLĄSKA**  
**WYDZIAŁ AUTOMATYKI, ELEKTRONIKI I INFORMATYKI**

**Projekt z Systemów Mikroprocesorowych**

**Smart Room**

Autorzy: Kamil Woźniak, Wojciech Kołodziej

Rok akademicki: 2020, semestr: 5, grupa: 1, sekcja: 4:

Kierujący pracą: dr. inż. Małgorzata Michalczyk

Gliwice, październik 2020

## **Spis treści**

1. Wstęp	3
1.1. Cel i zakres projektu	3
2. Harmonogram	4
2.1. Harmonogram zatwierdzony	4
2.2. Harmonogram wykonany	4
3. Kosztorys	5
4. Urządzenie wraz z aplikacją (odpowiednio zmodyfikować!!!)	6
4.1. Określenie problemu	6
4.2. Analiza rozwiązań	6
4.3. Zaproponowane rozwiązanie	6
4.4. Wykonanie (zmienić na właściwe słowa)	7
4.5. Problemy w trakcie tworzenia sprzętu i aplikacji (niepotrzebne usunąć!)	7
5. Podsumowanie	8
Literatura	9
Załączniki	10

## 1. Wstęp

My ludzie, jako istota rozumna od zawsze dążymy do ułatwienia sobie nawet najprostszych czynności. Większość narzędzi oraz przedmiotów użytku codziennego zostało wynalezione właśnie z tej naturalnej potrzeby człowieka, a wszystko to ma na celu wprowadzenie w naszą codzienność coraz większej ilości urządzeń i rozwiązań poprawiających naszą wygodę i komfort. Nawet sam Albert Einstein zwykł mawiać, że „wszystko powinno być tak proste, jak to tylko możliwe, ale nie prościej”.

Właśnie tutaj pojawia się projekt wychodzący naprzeciw tym wymaganiom. SMART ROOM jest to system mający na celu zwiększenie komfortu i łatwości użytkowania przestrzeni pokoju. Łączy on w sobie możliwość zdalnego sterowania i regulowania elementów pokoju za pośrednictwem telefonu komórkowego, co dodatkowo podwyższa wygodę użytkownika. Ważnym do poruszenia tematem jest jeszcze istota wprowadzania określonego nastroju w pokoju. Któż po stresującym dniu w przestrzeni publicznej nie chciałby wejść do strefy prywatnej z wcześniej włączonym zielonym, relaksującym oświetleniem. Podstawowymi elementami takiego pokoju są rolety okienne, oświetlenie główne (lampa sufitowa), oświetlenie dodatkowe (LED) w wybranych częściach pokoju. Po skończeniu elementów podstawowych zajęto się zwiększeniem bezpieczeństwa poprzez realizację elementu zamka otwieranego na kartę z tagiem RFID.

### 1.1. Cel i zakres projektu

Celem projektu jest ułatwienie życia użytkownika poprzez umożliwienie zdalnego sterowania swoim otoczeniem. Sterowanie zrealizowane zostało poprzez odbiornik podczerwieni podłączony do mikrokontrolera oraz wygodny i łatwy w obsłudze pilot podczerwieni.

Pierwszym elementem otoczenia są dwa systemy oświetlenia, w których można wyróżnić oświetlenie główne oraz oświetlenie dekoracyjne. Na oświetlenie dekoracyjne składa się taśma LED RGB umiejscowione nisko przy podłodze i służące do pracy

wieczorowo-nocnej oraz wprowadzające relaksującą i miłą dla oka atmosferę. Użytkownik jest w stanie sterować kolorem oświetlenia w zależności od nastroju czy też indywidualnych preferencji.

Elementem funkcjonalnym w naszym projekcie jest roleta okienna. Za jej pomocą możliwe jest regulowanie ilości światła wpadającego do pokoju oraz zapewnienie dodatkowej prywatności. Sterowanie nią możliwe jest dzięki zastosowaniu silniczka krokowego. Do silniczka doczepiony został podłużny element na którym osadzony został fragment rolety.

Przy próbie wejścia do pokoju odbywa się autoryzacja na podstawie danych przenoszonych poprzez elementy zawierające w sobie tag RFID. Aby uzyskać dostęp należy przyłożyć odpowiedni tag RFID do czujnika, co spowoduje otworzenie się zamka sterowanego elektrycznie, a w rezultacie możliwość otwarcia drzwi. Użytkownik każdorazowo przykładając tag RFID do drzwi na podstawie kolorowych diod będzie w stanie określić czy tag jest poprawny (dioda zielona) czy niepoprawny (dioda czerwona), bez konieczności każdorazowego ciągnięcia za klamkę.

Projekt zostanie zrealizowany na makiecie z uwagi na sytuację epidemiologiczną, ograniczający nas budżet, stosunkowo mała ilość czasu oraz możliwość łatwiejszej prezentacji. Makieta zostanie zbudowana z płyty osb z uwagi na łatwość montażu oraz względną odporność na zniszczenia. Ze względu na osadzenie projektu w makiecie odpowiednio przeskalowane zostaną urządzenia wykonawcze co pozwoli na ograniczenie wydatków poprzez potrzebę na przykład mniejszego silniczka czy też mniejszej ilości diod led i kabli łączących wszystko.

Podsumowując główne wymagania sprzętowe potrzebny będzie mikroprocesor Arduino Uno, silniczek DC (potem zamieniony na krokowy) wraz ze sterownikiem oraz dodatkowe zasilanie 12V, moduł podczerwieni oraz pilot do obsługi, taśma LED RGB, dioda czerwona i zielona, czytnik RFID oraz elementy posiadające tagi RFID, serwomechanizm do obsługi zamka, przekaźniki do zamykania i otwierania obwodów elektrycznych (co pozwala sterować świeceniem żarówki oraz oświetlenia LED), wszelkie materiały potrzebne do zbudowania samej makiety. Do tego na pewno przyda się dużo komponentów elektronicznych takich jak kabelki zarówno męsko-męskie jak i męsko-żeńskie, diody do sprawdzenia poprawności kodu, płytka stykowa, baterie.

W tym punkcie opisać cel i zakres projektu oraz do czego jej wyniki mogą być wykorzystane. Określić wymagania sprzętowe i programowe potrzebne do realizacji projektu.

## 2. Harmonogram

Przedstawić opis i założenia harmonogramu pracy z podziałem na dwie części:

### Tydzień 1

1. Zapoznanie się z problematyką projektu.
2. Wstępny zakup części potrzebnych do wykonania założonych zadań (wraz z kolejnymi realizowanymi krokami zakup dodatkowych części).
3. Zgromadzenie informacji (z dokumentacji technicznych udostępnionych w Internecie przez producentów) o zakupionych produktach oraz schematów połączeń potrzebnych do zrealizowania projektu.

### Tydzień 2

4. Zapoznanie się z możliwościami kontroli silniczka DC przy pomocy sterowników oraz pierwsze próby stworzenia rolety.
5. Implementacja pozwalająca na sterowanie ruchem silniczka w obie strony.
6. Próba poprawnego podłączenia przekaźnika elektromagnetycznego.
7. Stworzenie prototypu oświetlenia pokoju aby spełniał nasze oczekiwania (oświetlenie główne i LED).
8. Wstępne zapoznanie się z możliwościami sterowania silnikiem i oświetleniem bezprzewodowo.

### Tydzień 3

9. Pełne zapoznanie się z możliwościami sterowania silnikiem i oświetleniem bezprzewodowo.
10. Parowanie odbiornika podczerwieni z pilotem RC. Nawiązanie poprawnej komunikacji między urządzeniami.
11. Oprogramowanie modułu komunikacyjnego w sposób pozwalający na realizację wyznaczonych zadań zdalnie.

### Tydzień 4

12. Czynności związane z optymalizacją kodu oraz ulepszeniem współpracy sterowników i modułu komunikacyjnego.
13. [OPCJONALNE] Próba zrealizowania drzwi otwieranych za pomocą czytnika RFID
14. [OPCJONALNE] Próba wykonania automatycznego odświeżacza powietrza.

15. [OPCJONALNE] Próba stworzenia systemu zapamiętywania ustawień oraz możliwość dopasowania indywidualnych trybów pokoju dla użytkownika np. tryb dzienny/nocny czy też tryb pracy/relaksu.

16. Ewentualne zlutowanie wszystkich elementów ze sobą w celu zwiększenia odporności projektu na wszelkie upadki, szturchnięcia i tym podobne.

17. Osadzenie projektu na makiecie.

#### Tydzień 5

18. Prezentacja stworzonego projektu.

### 2.1. Harmonogram zatwierdzony

Podać dokładnie taki harmonogram jaki został zatwierdzony na początkowych zajęciach.

#### Tydzień 1

1. Zapoznanie się z problematyką projektu.
2. Wstępny zakup części potrzebnych do wykonania założonych zadań (wraz z kolejnymi realizowanymi krokami zakup dodatkowych części).
3. Zgromadzenie informacji (z dokumentacji technicznych udostępnionych w Internecie przez producentów) o zakupionych produktach oraz schematów połączeń potrzebnych do zrealizowania projektu.

#### Tydzień 2

4. Zapoznanie się z możliwościami kontroli silniczka DC przy pomocy sterowników oraz pierwsze próby stworzenia rolety.
5. Implementacja pozwalająca na sterowanie ruchem silniczka w obie strony.
6. Próba poprawnego podłączenia przekaźnika elektromagnetycznego.
7. Stworzenie prototypu oświetlenia pokoju aby spełniał nasze oczekiwania (oświetlenie główne i LED).
8. Wstępne zapoznanie się z możliwościami sterowania silnikiem i oświetleniem bezprzewodowo.

#### Tydzień 3

9. Pełne zapoznanie się z możliwościami sterowania silnikiem i oświetleniem bezprzewodowo.

10. Parowanie odbiornika podczerwieni z pilotem RC. Nawiązanie poprawnej komunikacji między urządzeniami.

11. Oprogramowanie modułu komunikacyjnego w sposób pozwalający na realizację wyznaczonych zadań zdalnie.

#### Tydzień 4

12. Czynności związane z optymalizacją kodu oraz ulepszeniem współpracy sterowników i modułu komunikacyjnego.
13. [OPCJONALNE] Próba zrealizowania drzwi otwieranych za pomocą czytnika RFID
14. [OPCJONALNE] Próba wykonania automatycznego odświeżacza powietrza.
15. [OPCJONALNE] Próba stworzenia systemu zapamiętywania ustawień oraz możliwość dopasowania indywidualnych trybów pokoju dla użytkownika np. tryb dzienny/nocny czy też tryb pracy/relaksu.
16. [OPCJONALNE] Próba stworzenia automatycznie otwieranego i zamykanego okna pod wpływem zmiany temperatury.
17. Ewentualne zlutowanie wszystkich elementów ze sobą w celu zwiększenia odporności projektu na wszelkie upadki, szturchnięcia i tym podobne.
18. Osadzenie projektu na makiecie.

#### Tydzień 5

19. Prezentacja stworzonego projektu.

## 2.2. Harmonogram wykonany

Harmonogram powinien obejmować faktycznie zrealizowany program projektu. Powinny także być przedstawione powody jego ewentualnej modyfikacji wraz z uzasadnieniem.



### Tydzień 1

1. Zapoznanie się z problematyką projektu.
2. Wstępny zakup części potrzebnych do wykonania założonych zadań (wraz z kolejnymi realizowanymi krokami zakup dodatkowych części).
3. Zgromadzenie informacji (z dokumentacji technicznych udostępnionych w Internecie przez producentów) o zakupionych produktach oraz schematów połączeń potrzebnych do zrealizowania projektu.

### Tydzień 2

4. Zapoznanie się z możliwościami kontroli silniczka DC przy pomocy sterowników oraz pierwsze próby stworzenia rolety.
5. Próba poprawnego podłączenia przekaźnika elektromagnetycznego.
6. Stworzenie prototypu oświetlenia pokoju aby spełniał nasze oczekiwania (oświetlenie główne).
7. Parowanie odbiornika podczerwieni z pilotem RC. Nawiązanie poprawnej komunikacji między urządzeniami.
8. Pełne zapoznanie się z możliwościami bezprzewodowego sterowania za pomocą pilota i odbiornika podczerwieni.

### Tydzień 3

9. Implementacja pozwalająca na sterowanie ruchem silniczka w obie strony.
10. Pełne zapoznanie się z możliwościami sterowania oświetleniem bezprzewodowo.
11. Oprogramowanie modułu komunikacyjnego w sposób pozwalający na realizację wyznaczonych zadań zdalnie.
12. Stworzenie makiety na której zostanie osadzony projekt.

### Tydzień 4

12. Czynności związane z optymalizacją kodu oraz ulepszeniem współpracy sterowników i modułu komunikacyjnego.
13. Wymiana zepsutego silniczka na silniczek krokowy.
14. Podłączenie, implementacja i test działania silniczka krokowego.
15. Zrealizowanie dodatkowego elementu w postaci drzwi otwieranych za pomocą czytnika RFID
16. Ewentualne zlutowanie wszystkich elementów ze sobą w celu zwiększenia odporności projektu na wszelkie upadki, szturchnięcia i tym podobne.
17. Osadzenie projektu na makiecie.

**Tydzień 5**

18. Prezentacja stworzonego projektu.

**3. Kosztorys**

W tym rozdziale należy umieścić w tabeli wszystkie elementy z których składa się projekt wraz nazwą, producentem, ceną jednostkową, ich liczbą oraz kosztem. Musi być podany całkowity koszt projektu.

\* Elementy znajdowały się w zestawie startowym Arduino, więc ich cena jest szacunkowa.

lista przewidywanych zakupów:

Element	Ilość	Koszt [pln]
klon Arduino Uno + shield	1	30*
Kabel USB dla UNO R3	1	w zestawie
Płytką stykowa	1	3*
Zestaw przewodów	1	6*
diody	~2	1.8
moduł podczerwieni + Pilot	1	14*
przełącznik elektromagnetyczny	2	12
czytnik RFID + karta + brelok	1	10*
makieta	1	0
zasilacz 12V	1	z odzysku (16)
sterownik ULN2003 + silniczek	1	8.5
taśma LED 1m	1	z odzysku (15)
	<b>Suma</b>	<b>85,3</b>

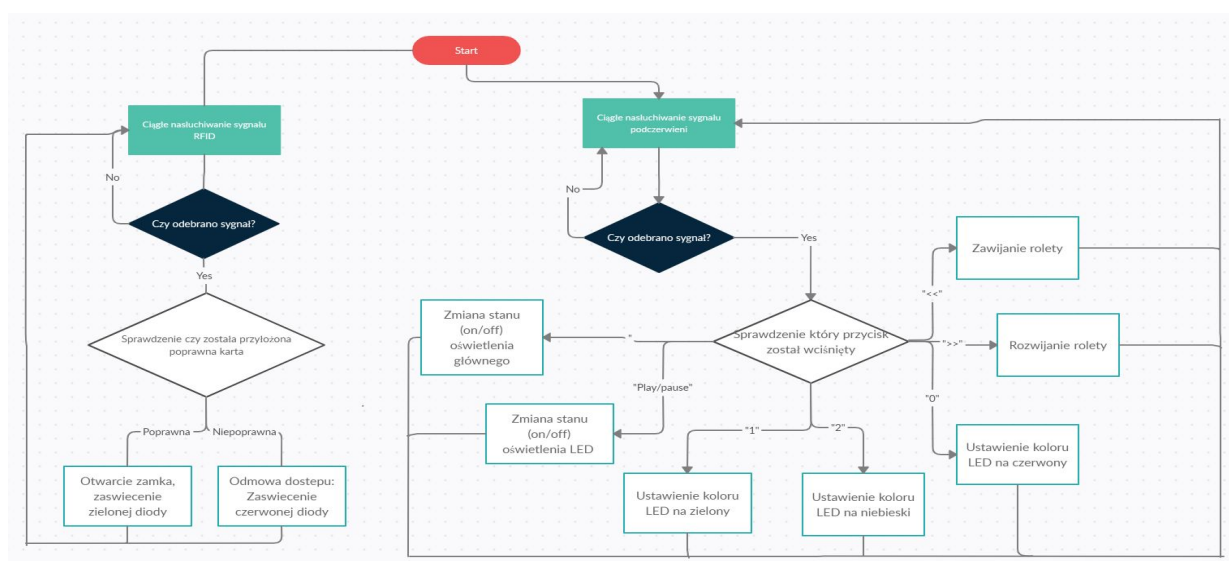
## 4. Urządzenie wraz z aplikacją

Należy podać krótki opis celu który chcemy osiągnąć wykonując projekt.

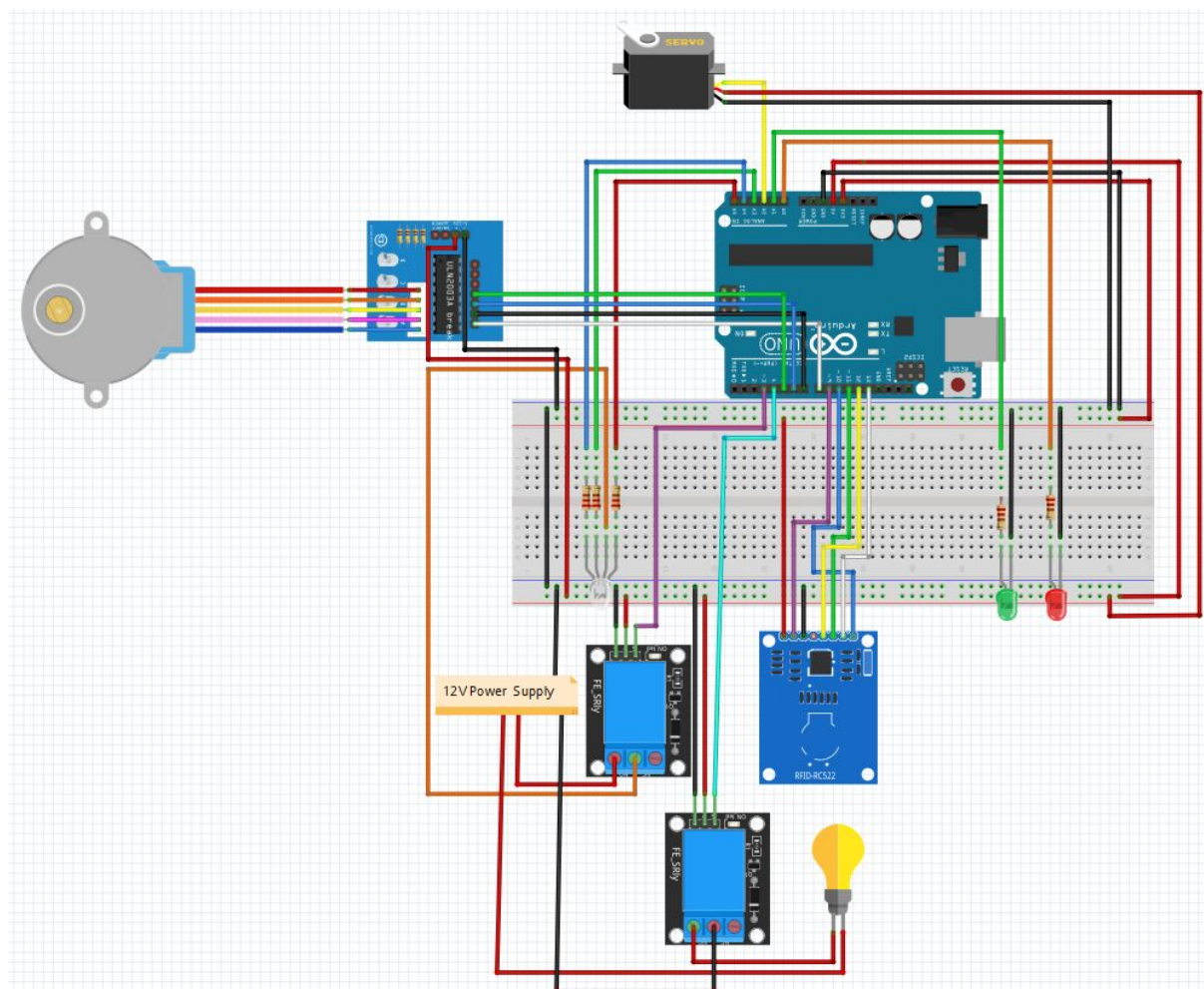
### 4.1.Określenie problemu oraz schematy.

Stworzenie układu obsługującego zdalne sterowanie oświetleniem i roletą oraz zamek do drzwi otwierany poprzez sprawdzenie karty RFID wymaga zastosowania odpowiednich elementów elektronicznych oraz oprogramowania. Poniższe schematy przedstawiają zastosowane urządzenia oraz objaśniają implementację programu.

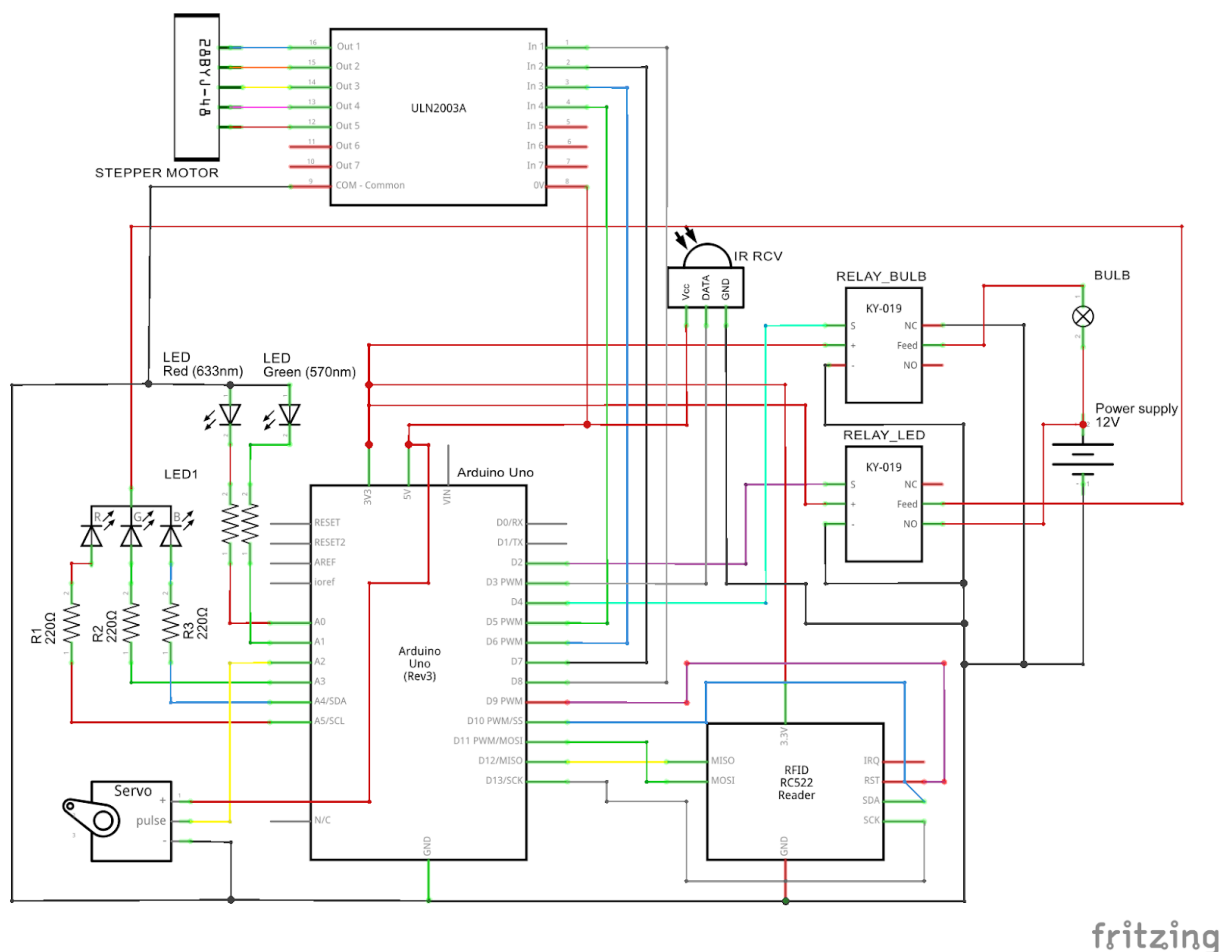
Należy zawsze odwoływać się do tego co jest na Rysunek 1.



**Rys.1 - Schemat blokowy (schemat w pełnej rozdzielczości znajduje się w pliku.zip, w którym został wysłany projekt)**



**Rys. 2-** Schemat ideowy układu wykonany w aplikacji "Fritzing" dla większej czytelności



**Rys.3 - Klasyczny schemat ideowy układu**

## 4.2. Analiza rozwiązań

### Zdalna komunikacja

Kryteria: możliwie mała trudność wykonania, niski koszt oraz duża dostępność.

Możliwości zrealizowania zdalnej komunikacji pomiędzy użytkownikiem a mikrokontrolerem były następujące:

- moduł Bluetooth: komunikacja poprzez telefon użytkownika za pośrednictwem dedykowanej aplikacji twórców Arduino o nazwie "Arduino Commander" z modulem bluetooth podłączonym do mikroprocesora
- moduł wifi: komunikacja poprzez telefon użytkownika za pośrednictwem dedykowanej aplikacji twórców Arduino o nazwie "Arduino Commander" z modulem wifi podłączonym do mikroprocesora
- moduł radiowy: komunikacja radiowa między urządzeniem nadawczym częstotliwości radiowych a odbiornikiem radiowym podłączonym do mikrokontrolera

- odbiornik podczerwieni: komunikacja między pilotem na podczerwień a odbiornikiem podczerwieni podłączonym do wejścia

### **4.3.Zaproponowane rozwiązanie**

W tym punkcie należy przedstawić wybrane rozwiązanie problemu, wraz z uzasadnieniem wyboru na podstawie kryteriów z poprzedniego punktu.

#### **Przełącznik**

W przypadku przełącznika zdecydowaliśmy się na przełącznik dwumodułowy SRD-05VDC-SL-C.

Przełącznik ten był dołączony do starter kitu naszego Arduino. Kolejnym przełącznikiem jest przełącznik tego samego typu, który został dokupiony w późniejszym terminie.

#### **Zdalna komunikacja**

Z uwagi na kryteria wyboru zdecydowano, iż komunikacja zdalna odbywać się będzie za pomocą odbiornika podczerwieni (VS1838B) i pilota (nadajnika) podczerwieni. Taka decyzja podjęta została po konsultacji z Kierującym pracą. Z uwagi na ograniczony czas wybrano takie rozwiązanie, biorąc pod uwagę trudności w poprawnej obsłudze pozostałych rozwiązań.

#### **Obsługa silniczka**

Obsługa silniczka została zrealizowana za pomocą sterownika dwukanałowego L293D, który pozwala na swobodne kontrolowanie prędkości i kierunku obrotu silniczka. Do całego układu ze sterownikiem zostało podpięte zewnętrzne zasilanie w postaci paczki czterech baterii 1.5V połączonych szeregowo oraz przycisku z diodą LED, który uruchamia silniczki, a razem z nim włącza się dioda LED.

Po spaleniu się sterownika L293D, użyty został pięcio woltowy silniczek krokowy kontrolowany przez sterownik ULN2003, który pozwala na wiele dokładniejsze sterowanie roletą. Sterownik pozwala na zmianę prędkości oraz kąta obrotu silniczka.

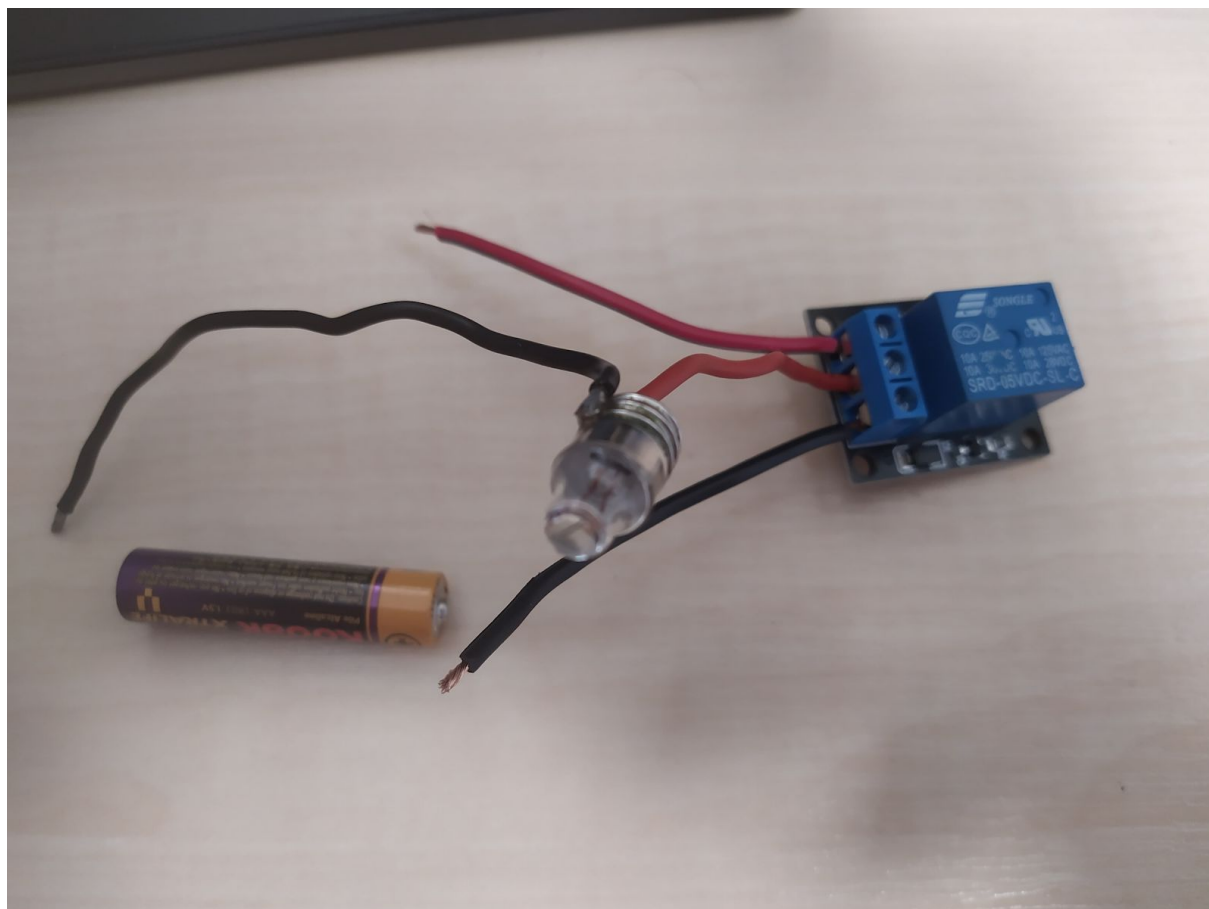
#### **Oświetlenie**

W celach ograniczenia budżetu jako oświetlenie dodatkowe (taśma LED) użyto kawałka pozostałej taśmy po remoncie i przylutowano do niej poszczególne piny, które można podłączyć do arduino i zasilania. Jako zasilanie użyto starego zasilacza 12V znalezione na strychu. Jako oświetlenie główne użyto małej żarówki 1.5V.

### **4.4.Wykonanie poszczególnych funkcjonalności**

### Przekaźniki sterujące oświetleniem

Z uwagi na fakt iż, zastosowany przekaźnik został dołączony do zestawu z Arduino postanowiono z niego skorzystać. Niefortunnie nie miał on opisanych wyjść, więc zanim przystąpiono do pracy z nim, należało eksperymentalnie wyznaczyć wyjścia przekaźnika. Do tego celu użyto niskonapięciowej żarówki zasilanej baterią. Testując różne konfiguracje wyznaczono, które wyjścia są zwarte przy braku sterowania zewnętrznego. Zwarte wyjścia zostały oznaczone czarnymi kablami na zdjęciu poniżej.



**Rys. 4-** *Elementy i sposób podłączenia użyty do sprawdzenia niepodpisanych wyjść przekaźnika*

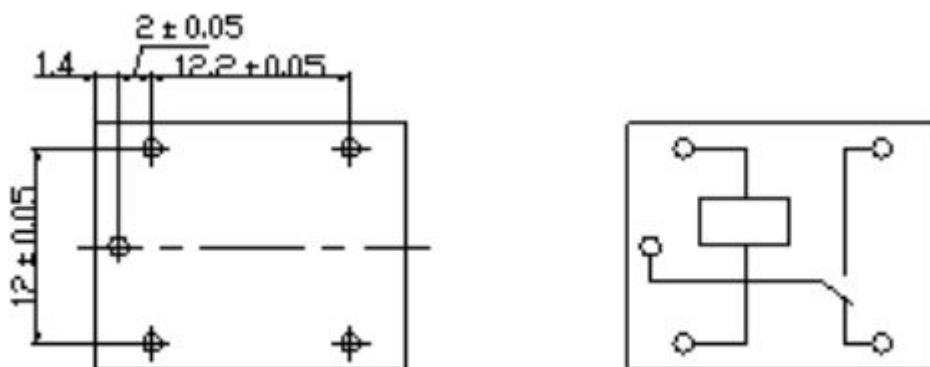
Następnie przystąpiono do podłączenia przekaźnika poprzez płytkę stykową do Arduino Uno. Nie było z tym większego problemu, gdyż chiński producent zdecydował się podpisać wejścia do przekaźnika (w przeciwieństwie do wyjść). Urządzenie zostało podłączone w następujący sposób:

Oznaczenie na przekaźniku: " s "	-	Arduino Pin
Oznaczenie na przekaźniku: " + "	-	Arduino GND

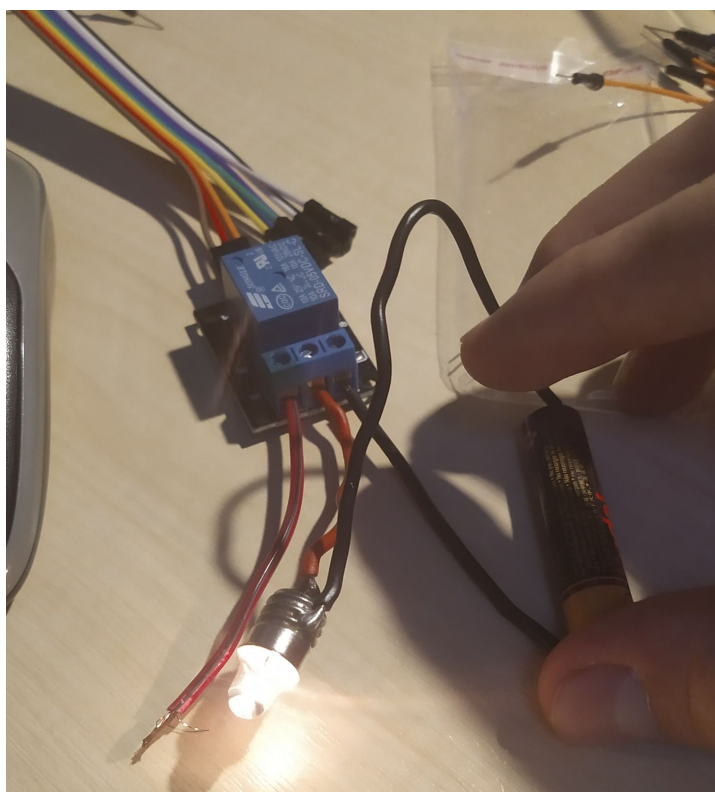


Oznaczenie na przekaźniku: “ - ” - Arduino 5V

Sterowanie przekaźnikiem odbywa się poprzez przypisanie stanu “HIGH” lub “LOW” na wyjściu pinu podpiętego do Arduino. Na początku przekaźniki są nieaktywne i na wyjściu ich pinu wystawiany jest sygnał LOW. W momencie naciśnięcia przycisku odpowiadającego za włączenie oświetlenia na wyjściu pinu wystawiany jest sygnał HIGH i układ się zamyka co pozwala na dopływ prądu do żarówki.

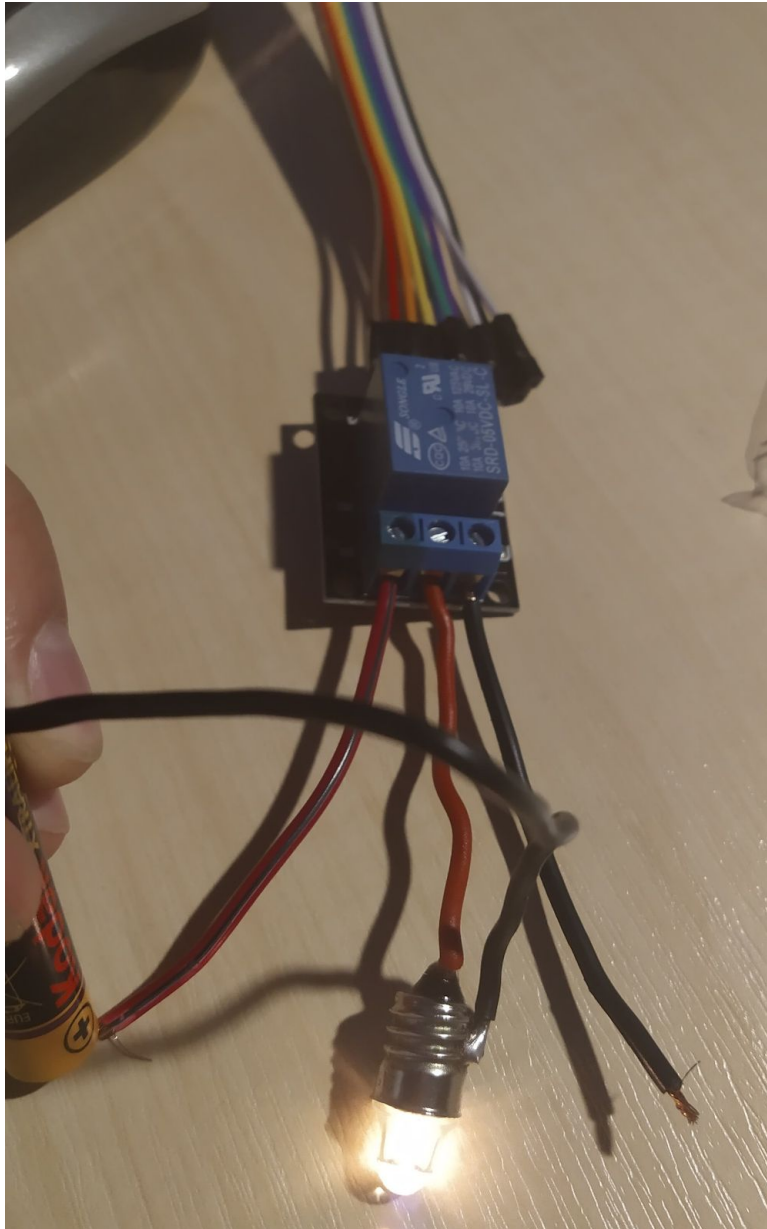


**Rys. 5-** Schemat przekaźnika



**Rys. 6-** Test przed przełączeniem przekaźnika





**Rys. 7-** *Test po przełączeniu przekaźnika*

### **Zdalna komunikacja**

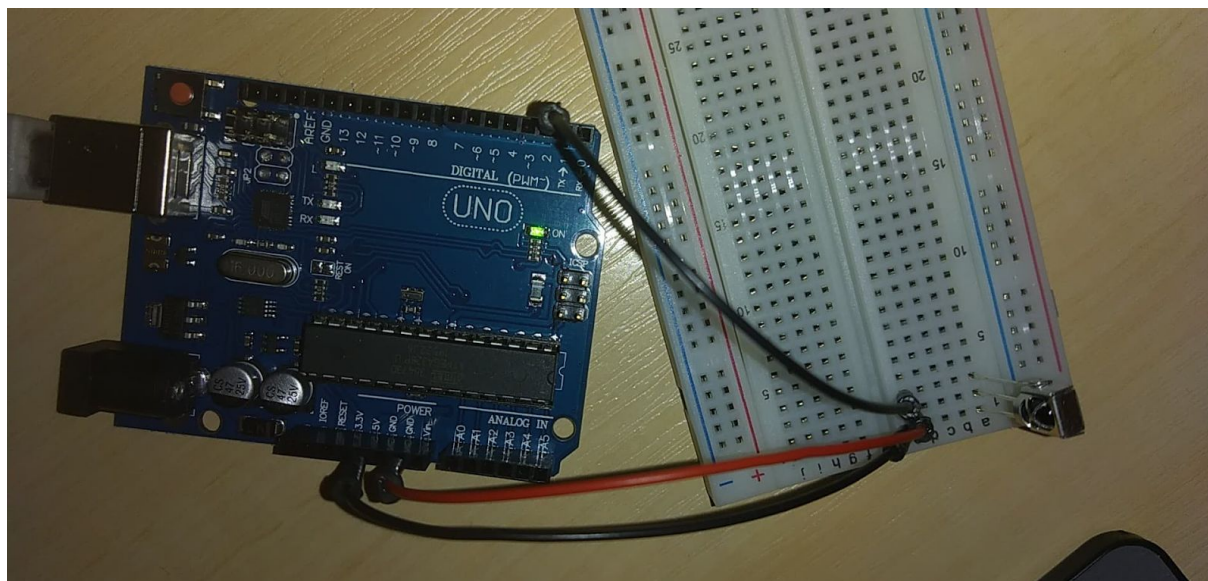
Zdecydowaliśmy się użyć odbiornika podczerwieni VS1838B oraz pilota podczerwieni pierwotnie używanego do sterowania systemem nagłośnienia w samochodzie. Odbiornik i pilot działały razem bez konieczności dodatkowej konfiguracji.

Odbiornik podczerwieni został podłączony poprzez płytkę stykową z arduino w następujący sposób:

VS1838B pin 1 - Arduino Pin 1

VS1838B pin2 - Arduino GND

VS1838B pin3 - Arduino 5V



**Rys. 8 -** Podłączenie odbiornika podczerwieni z Arduino Uno

Do oprogramowania poprawnej komunikacji posłużono się funkcjami zawartymi w bibliotece IRremote.h. Włączenie odbierania danych odbywało się za pomocą funkcji `irrecv.enableIRIn()`, która została wywołana w void `setup()`. Następnie w void `loop()` następowało sprawdzenie instrukcji warunkowej, która wykonywała się, gdy odbiornik otrzymał dane wysłane z pilota (wciśnięcie klawisza). Aby w pełni zrozumieć paczkę danych otrzymywanych od pilota, zdecydowano wyświetlić jej zawartość na ekranie serial monitora dla różnych przycisków.

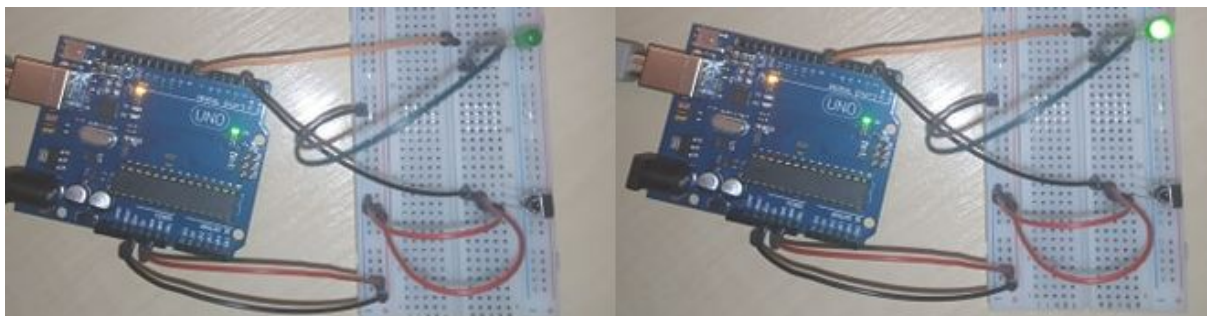
```
13:53:51.295 -> -1
13:53:51.295 -> wynik = 0xC101E57B
13:53:51.295 -> 32
14:02:07.761 -> -1
14:02:07.761 -> wynik = 0xB3E2CB05
14:02:07.761 -> 32
14:09:03.144 -> -1
14:09:03.144 -> wynik = 0x33D1A034
14:09:03.144 -> 32
```

**Rys. 9 -** Paczki danych otrzymane dla różnych klawiszy

Na podstawie analizy ramek odebranych przez odbiornik wysnuto wniosek, iż pierwsza linijka oznacza typ nadajnika (w tym przypadku zawsze -1), druga to wynik wciśniętego klawisza w

kodecie szesnastkowym, trzecia linijka to ilość przesyłanych bitów za jednym razem. Aby umożliwić uzależnienie działania programu w zależności od wciśniętego klawisza należy posłużyć się kodem szesnastkowym wciśniętego klawisza.

Pierwszy test działania komunikacji zdalnej przeprowadzono na diodzie, która podłączono wraz z rezystorem na płytkę stykową.



**Rys. 10-** Pierwszy test, przed i po wciśnięciu klawisza na pilocie podczerwieni

## Silniczek DC



**Rys. 11-** Schemat rozkładu pinów na sterowniku

Sterownik L293D posiada dwa wejścia na zasilanie - **Vcc1** które jest używane do kontrolowania wszystkich wewnętrznych obwodów logicznych i powinno być pięć wołtów. Owy pin podpięty jest do wyjścia 5V na arduino. Drugim pinem zasilania jest **Vcc2**, który używany jest jako źródło prądu dla podpiętych silniczków i może być od 4.5V do 36V, w naszym przypadku jest to około 6V pozyskane z zlutowanych ze sobą czterech baterii 1.5V. Oba piny mają wspólną masę na pinie numer 4.

Piny **OUT1,OUT2** są bezpośrednio podpięte do silniczka i przekazują one zasilanie oraz informacje o kierunku w którym ma obracać się silniczek.

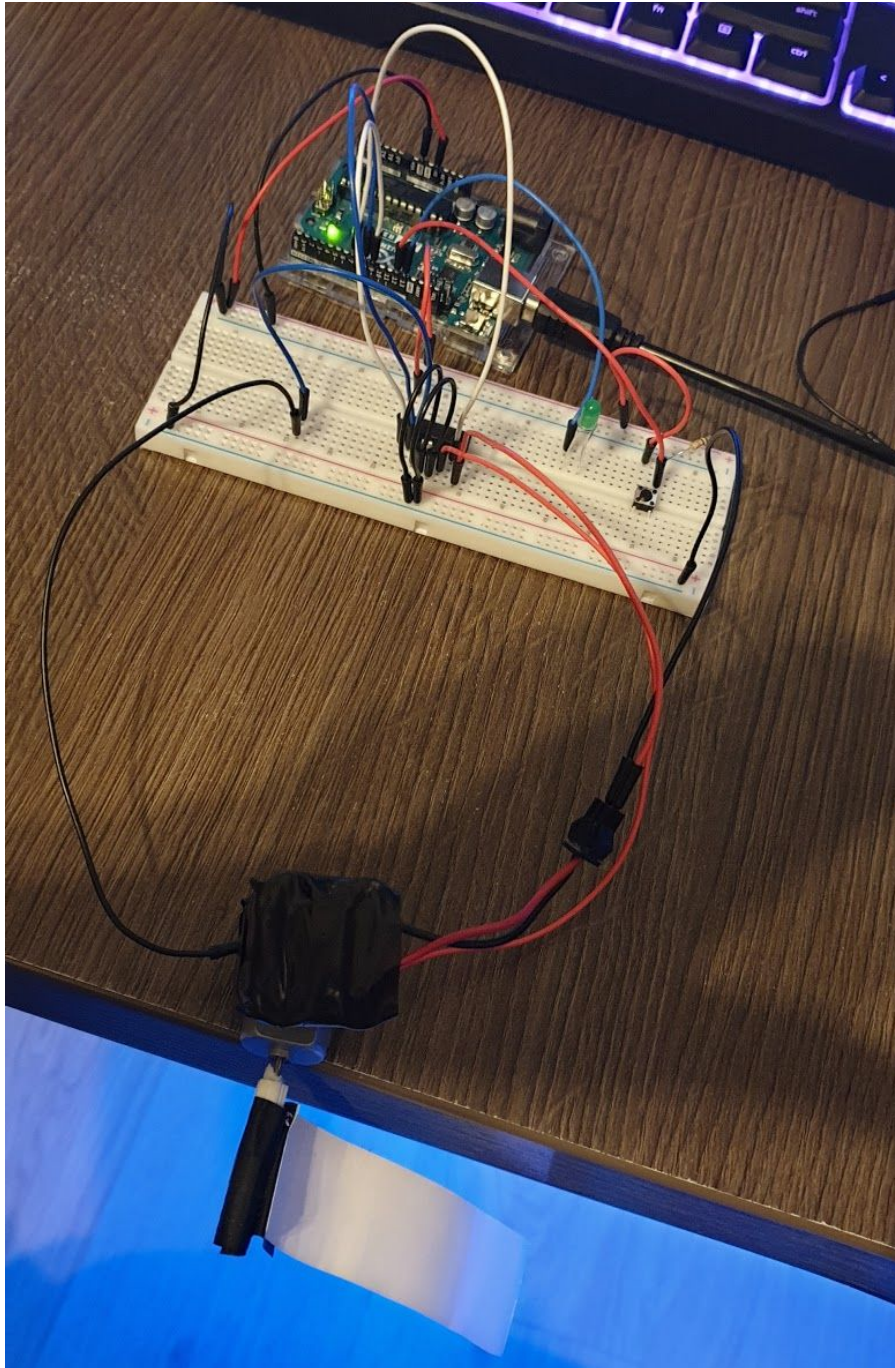
Pin **ENA**, **IN1**, **IN2** są odprowadzane do arduino i pozwalają odpowiednio na kontrolę prędkości silniczka oraz samo uruchomienie go. Zmienna w kodzie odpowiadająca pinowi **ENABLE** może przyjmować wartości od 0 do 255, natomiast piny **IN1** oraz **IN2** przyjmują wartości logiczne HIGH lub LOW z odpowiednimi kombinacjami:

IN1	IN2	Spinning Direction
Low(0)	Low(0)	Motor OFF
High(1)	Low(0)	Forward
Low(0)	High(1)	Backward
High(1)	High(1)	Motor OFF



**Rys. 12-** *Lutowanie paczki baterii*



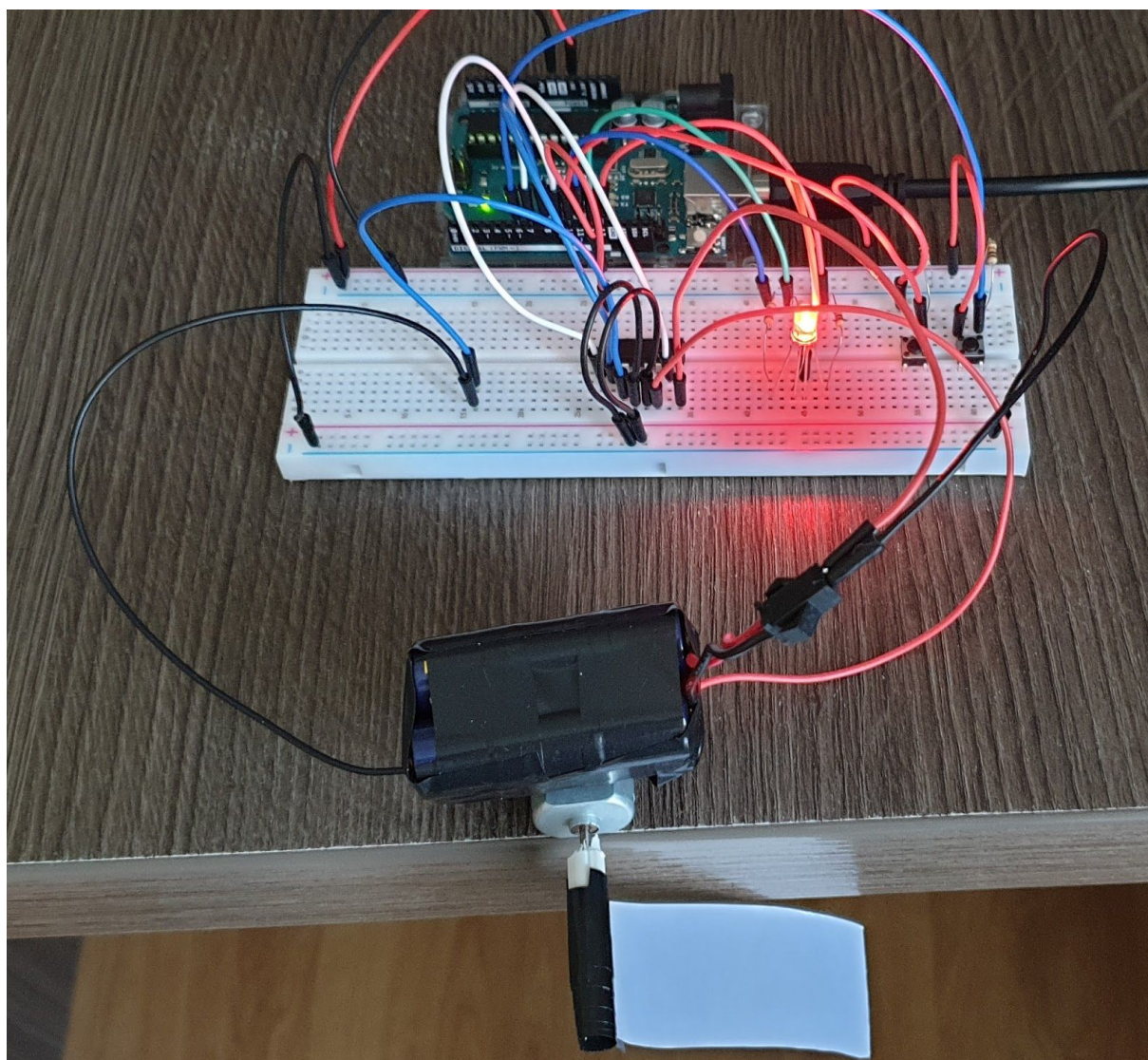


**Rys.13** - cały układ silniczka i sterownika

Powyższa konstrukcja przyczepiona do końcówki silnika jest tymczasowa i pozwala na łatwiejsze zaobserwowanie obrotu oraz jego kierunku. Docelowo będzie tam mechanizm obsługujący opuszczanie i podnoszenie rolety, jednakże wstępne podejścia skonstruowania takiego mechanizmu pokazały, że nie będzie to takie proste.

**Silniczek DC tydzień 2:**

W tym tygodniu dodano drugi przycisk, który pozwala na obracanie się silniczka w drugą stronę. Ponadto dodano diodę RGB (zamiast jednokolorowej diody, która była poprzednio), która zmienia kolor w zależności od stanu silniczka: **czerwony** - silniczek w spoczynku, **zielony** - silniczek obraca się w kierunku przeciwnym do wskazówek zegara, **niebieski** - w kierunku zgodnym do wskazówek zegara. Wymyślono sposób przekazania momentu obrotowego silniczka na mechanizm rolety poprzez koła zębate, które na dzień dzisiejszy opóźniają dalsze prace nad makietą. Pojawiło się też pytanie jak zachowa się zwykły silniczek DC w zastosowaniu praktycznym (w mechanizmie rolety) - czy będzie wystarczająco dokładny przy takim rozwiązaniu. Jeżeli nie spełni oczekiwań wymagana będzie zmiana całego układu na silnik krokowy i odpowiedni dla niego sterownik, co powinno zająć kilka godzin pracy. [link do filmiku z przykładem działania układu](#)



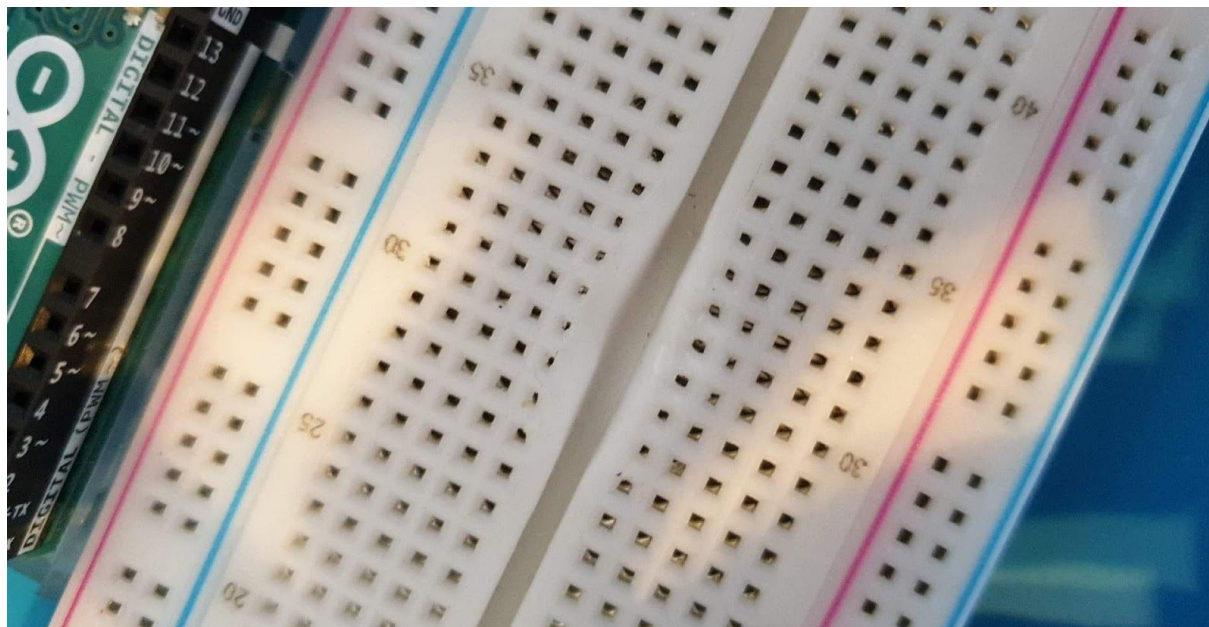
**Rys. 14-** układ silniczka i sterownika z dwoma przyciskami



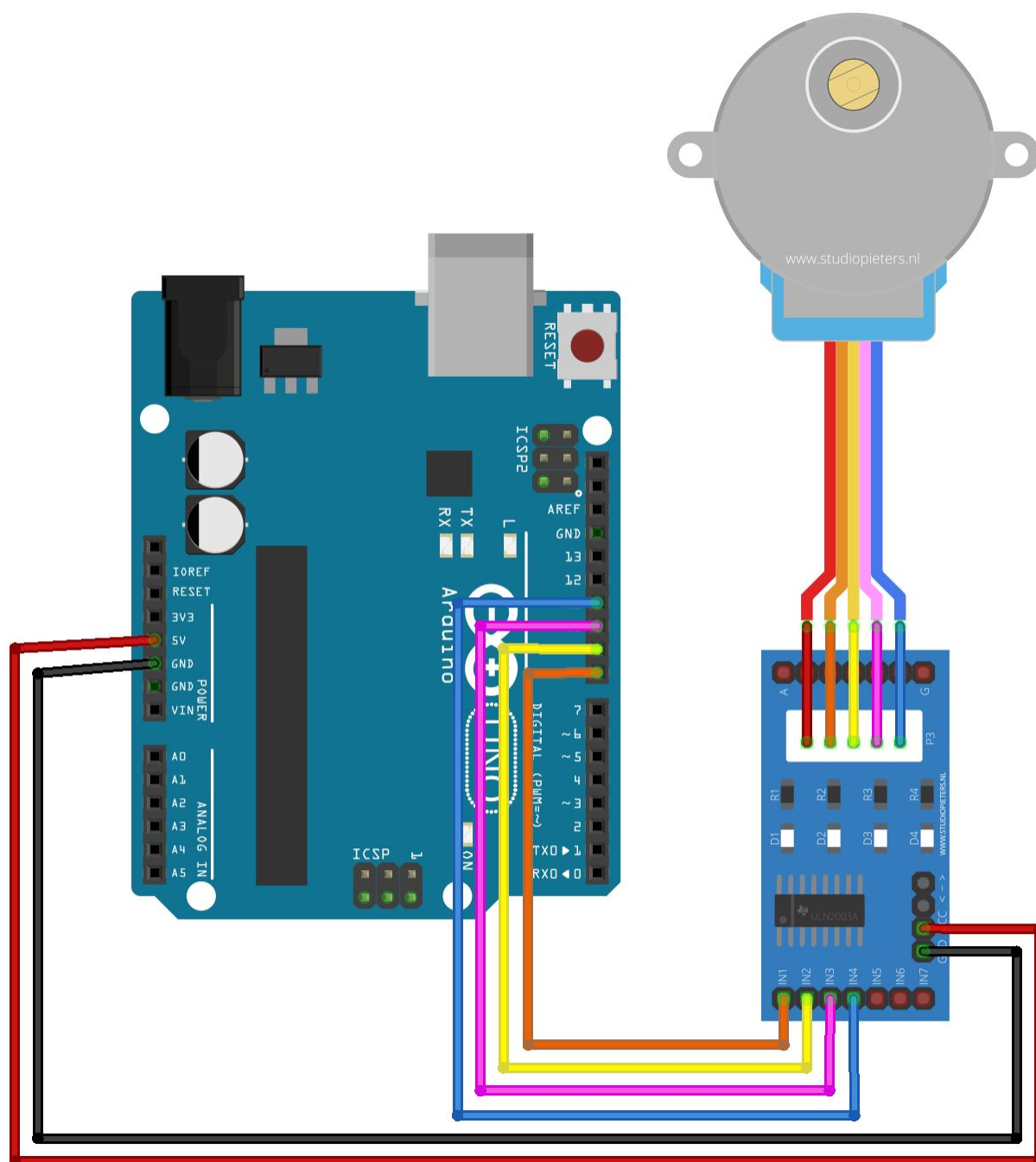
**Silniczek tydzień 3:**

W poprzednim tygodniu rozważano wymianę silniczka DC na silniczek krokowy, decyzja ta została jednak niejako podjęta za nas w momencie kiedy z powodu błędnego podłączenia jakiegoś pinu spalił się sterownik L293D i obracanie się silniczka w drugą stronę przestało działać. Przystąpiono do realizacji układu na silniczku krokowym i dostarczonym do niego sterowniku. Po owej realizacji zaprojektowanie modelu rolety okazało się dużo prostsze niż przypuszczano i nie było konieczne użycie kół zębatach, ponieważ silniczek krokowy porusza się wystarczająco wolno i dokładnie, że można do niego bezpośrednio przyczepić roletę. Implementacja zdalnego sterowania silniczkiem nie sprawiła większych problemów i jej wstępna wersja była stosunkowo prosta w realizacji. Pozostaje jedynie wprowadzić usprawnienia ułatwiające łatwość i uniwersalność obsługi sterowania - przykładowo wprowadzić to, aby roleta nie mogła zacząć się zwijać gdy jest w całości zwinięta.

[Prezentacja działania mechanizmu rolety](#)



**Rys. 15-** Zdjęcie szkód spowodowanych nagrzaniem się sterownika

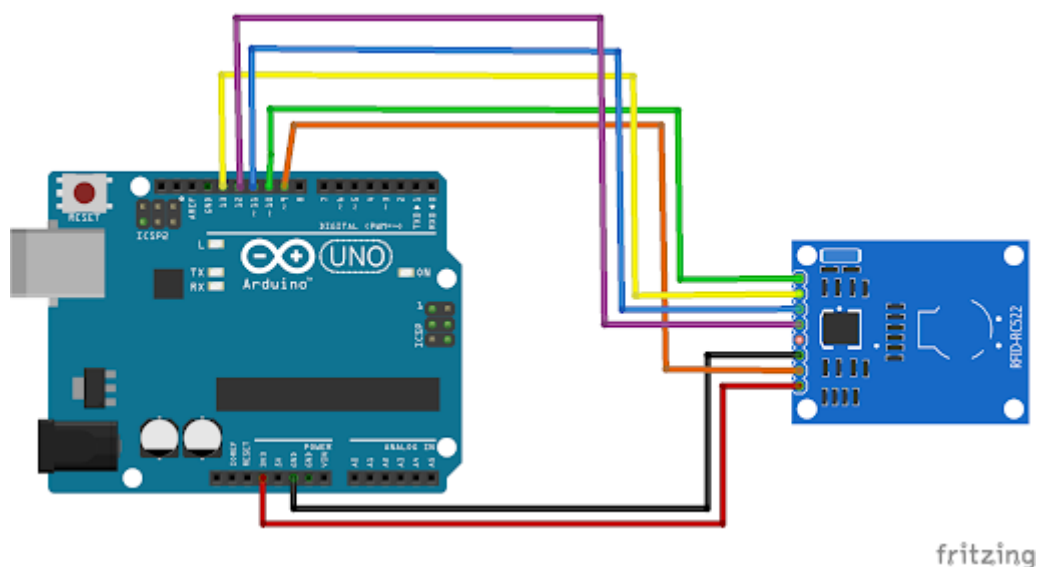


**Rys.16** - Schemat podłączenia silniczka krokowego i sterownika do Arduino

### Czytnik RFID

Czytnik został użyty w opcjonalnym wariantcie naszego projektu, tj. projektu zamka do drzwi otwieranego chipem lub namagnesowaną kartą PICC (ang. Proximity Integrated Circuit Card). W celu poprawy niezawodności komunikacji pierwszym krokiem było przylutowanie kabelków męsko-męskich do urządzenia. Czytnik pracuje na zasilaniu 3,3V.





**Rys. 17-** Schemat podłączenia czytnika RFID

Po podłączeniu czytnika zaimplementowano kod przykładu zawartego w bibliotece MFRC522.h, którego celem było sprawdzenie poprawności podłączenia oraz działania urządzenia oraz zdobycie unikalnego kodu uid naszego breloczka oraz karty. Poprzez zaimplementowanie wyświetlania informacji w monitorze portu szeregowego sprawdzono poprawność otrzymywanych danych, co umożliwiło przejście do implementacji kodu obsługującego serwomechanizm oraz diody na podstawie przyłożonej karty magnetycznej lub breloczka.

```
This code scan the MIFARE Classic NUID.
Using the following key: FF FF FF FF FF FF PICC type: MIFARE 1KB
A new card has been detected.
The NUID tag is:
In hex: 5C 9F 38 23
In dec: 92 159 56 35
PICC type: MIFARE 1KB
A new card has been detected.
The NUID tag is:
In hex: 09 7F C8 99
In dec: 09 127 200 153
PICC type: MIFARE 1KB
Card read previously.
```

RFID TAG

RFID CARD

**Rys. 18-** Informacje zawarte w monitorze portu szeregowego po przyłożeniu karty  
[Prezentacja działania zamka do drzwi na podstawie czytnika RFID](#)

### Serwomechanizm

Serwomechanizm posłuży nam jako zamek do drzwi, który po przyłożeniu prawidłowej karty do zamka będzie się otwierał. W implementacji obsługi serwomechanizmu posłużono się biblioteką servo.h, co umożliwiło bardzo precyzyjną kontrolę obrotu serwomechanizmu. Podczas tworzenia zamka do drzwi dodatkowo zostały umieszczone dwie diody LED sygnalizujące o poprawności przyłożonej karty.

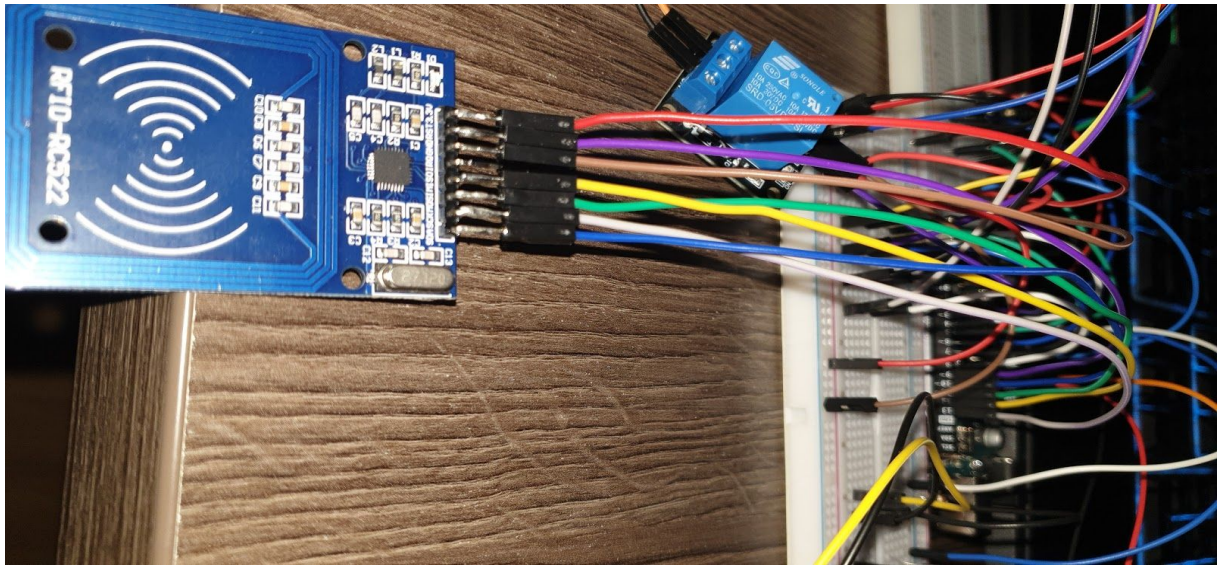


**Rys. 19-** Zastosowany serwomechanizm

### Końcowe prace nad całym układem

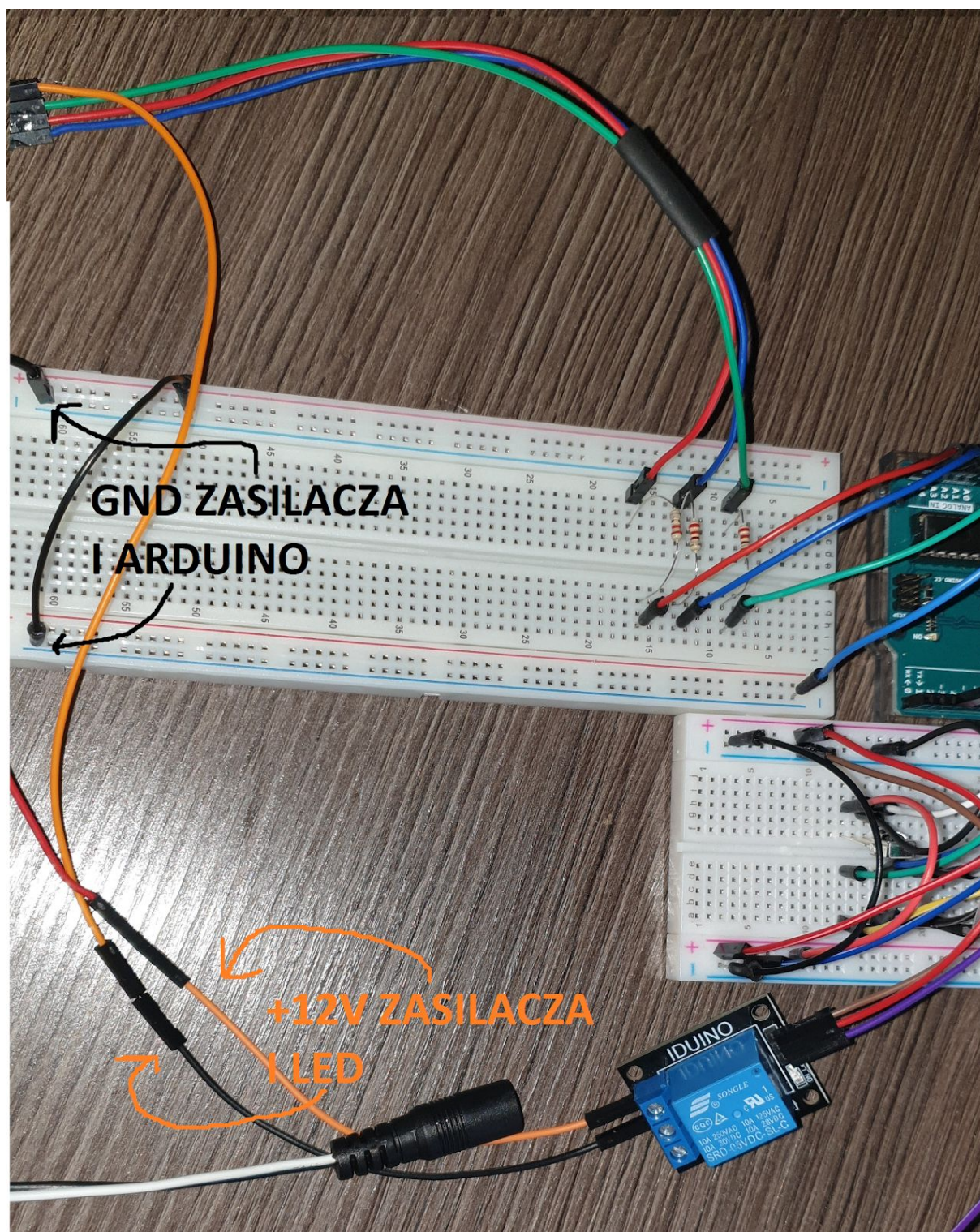
W ostatnim tygodniu skupiono się na osadzeniu wszystkich elementów na jednym Arduino, tak aby działały ze sobą współbieżnie. Okazało się to nie lada wyzwaniem ze względu na biblioteki które używano, a mianowicie biblioteki Servo.h, IRremote.h oraz Tune. Wymienione biblioteki korzystają ze wspólnych timerów i można przez to użyć jedynie dwóch z trzech wymienionych plików nagłówkowych. Zdecydowano się na pozbycie się Buzzera podczas otwierania drzwi i w kodzie źródłowym pozostałych bibliotek zmieniono nazwy timerów na niekolidujące ze sobą. Dodatkowo wprowadzono usprawnienie sterowania roletą - teraz układ wie gdy roleta jest rozwinięta lub zwinięta i nie pozwala na dodatkowe zwijanie już zwiniętej rolety, co jest zaprezentowane na poniższym filmiku prezentacji układu. Jako dodatkowe oświetlenie pokoju użyto taśmy LED RGB zasilanej poprzez 12V z sieci. Wyłączanie i włączenie oświetlenia zostało rozwiązane dzięki przekaźnikowi - jeden przycisk na pilocie włącza i wyłącza taśmę (dzięki funkcji switch-case) i dodatkowo możliwa jest zmiana kolorów diod dzięki pozostałym przyciskom na pilocie. Podobne rozwiązanie zostało zastosowane przy oświetleniu głównym pomijając możliwość zmiany kolorów.

[PREZENTACJA UKŁADU](#)



*Rys. 20- Sposób podłączenia czytnika RFID*

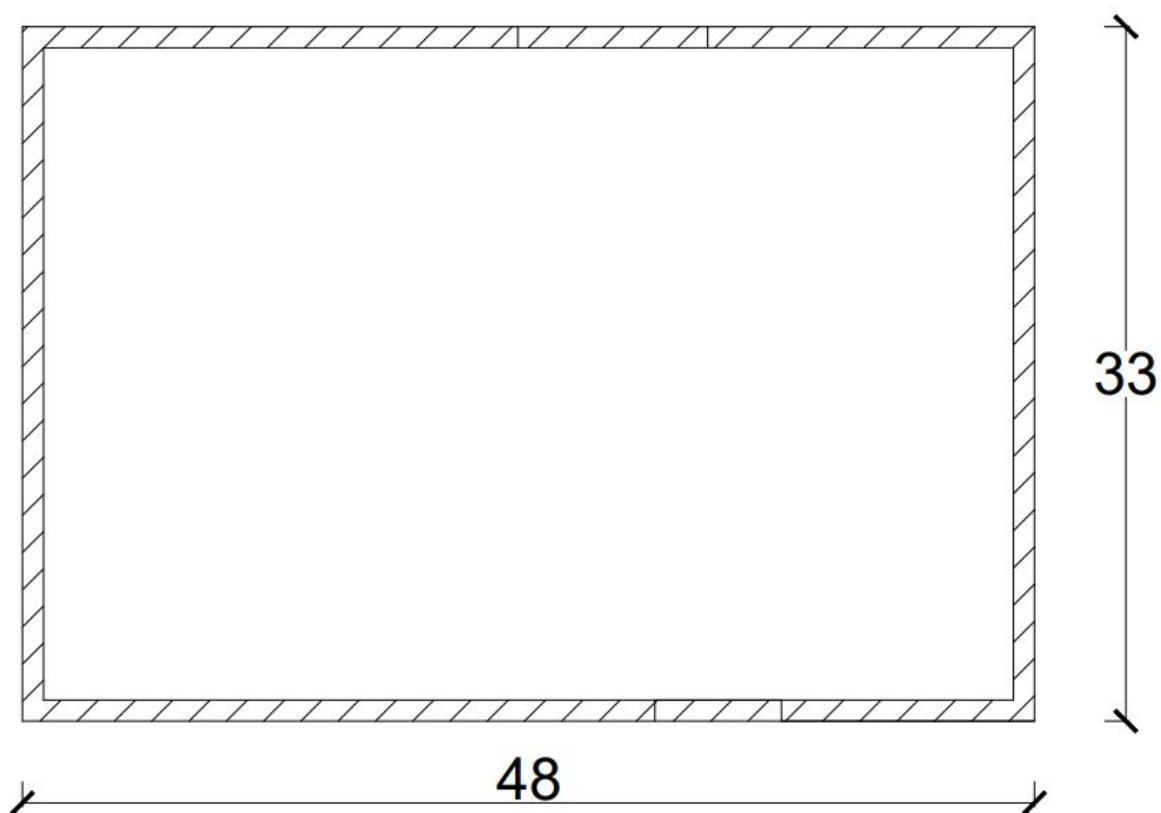




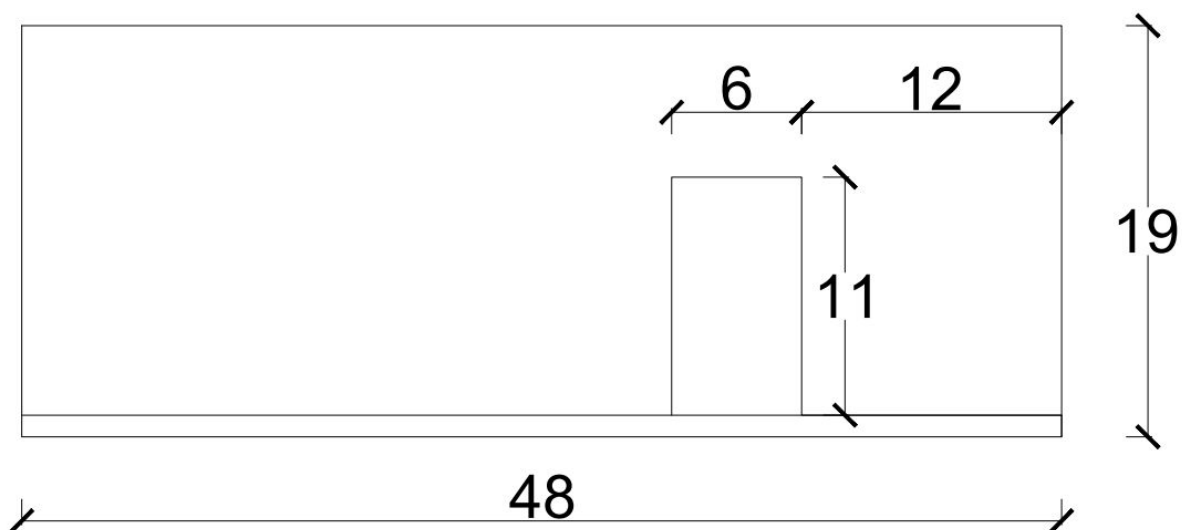
*Rys. 21- Sposób podłączenia taśmy LED*

### Makieta

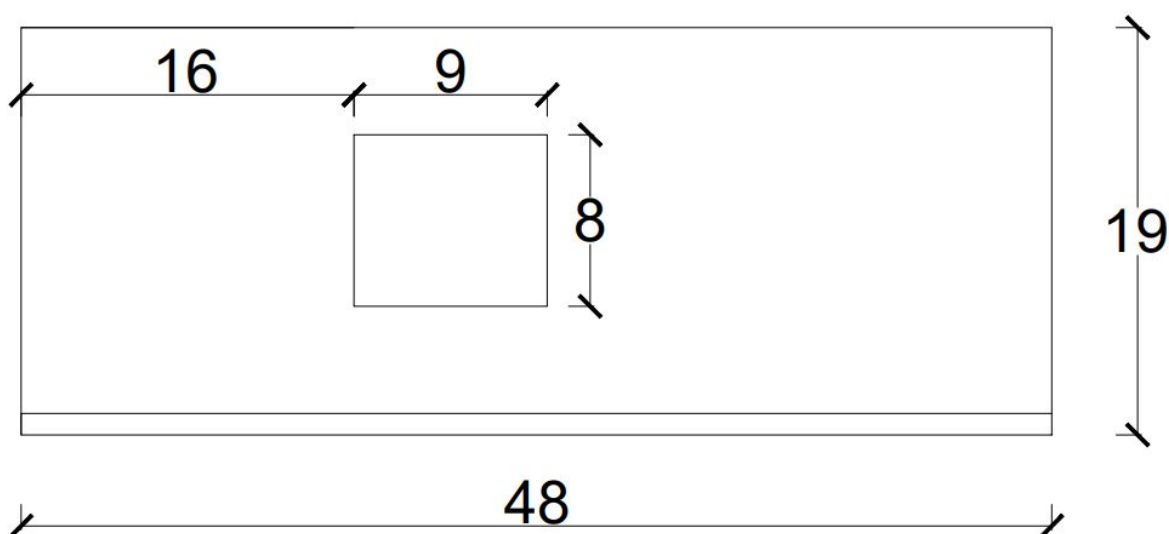
Projekt został osadzony na makiecie w celu łatwiejszej prezentacji. Makieta została stworzona ze starych fragmentów mebli, przez co jej koszt, który jest znikomy, nie został uwzględniony w kosztorysie. Poniżej zamieszczono rzuty makiety wykonane w programie AutoCad. Wartości liczbowe podane są w centymetrach.



**Rys. 22-** Rzut górny makiety wykonany w programie AutoCad



**Rys.23** - Rzut frontowy makiety z uwzględnionymi drzwiami wykonany w programie AutoCad



*Rys.24 - Rzut tylny makiety z uwzględnionym oknem wykonany w programie AutoCad*

#### 4.5.Problemy w trakcie tworzenia sprzętu

W tym punkcie przedstawiono jakie były problemy w trakcie jego realizacji i jak je rozwiązano.

- kabel do arduino zepsuty - **użycie kabla z drukarki;**
- brak oznaczeń w chińskich komponentach - **wyszukanie dokumentacji online;**
- lutowanie paczki baterii zamiast kupienia jednej 6V - **poświęcona dodatkowa godzina;**
- nieumiejętność przeniesienia momentu obrotowego silniczka na roletę - **wykorzystanie kół zębatach, a potem przejście na silniczek krokowy;**
- kwarantanna;
- umiejscowienie diody RGB w złym kierunku - **aby otrzymać pożądaną kolor, należy jego wartość ustawić na 0, a nie 255;**
- spalenie się sterownika L293D silniczka DC - **przejście na silniczek krokowy;**
- kolidacja dodatkowych bibliotek - **usunięcie jednej z bibliotek i modyfikacja kodu źródłowego pozostałych;**
- błędne podłączenie pinów czytnika kart RFID;
- niemożność implikacji włączenia i wyłączenia światła na jednym przycisku - **zmiana "if" na "switch-case";**
- zbyt słabe świecenie taśmy LED - **podłączenie masy zasilacza do masy Arduino;**



## 5. Podsumowanie

Z założonych w harmonogramie wytycznych udało się wykonać prawie wszystko łącznie z większością opcjonalnych celów poza oknem otwieranym automatycznie pod wpływem zwiększenia się temperatury oraz automatycznego odświeżacza powietrza z powodu braku czasu. Najtrudniejszy w implementacji okazał się skaner kart RFID z powodu wielokrotnego złego podłączenia pinów oraz tworzenia się zwarcia, ponieważ dwa kabelki się stykały. Implementacja wszystkich modułów na jednym kodzie okazała się trudna do wykonania z powodu kolidujących bibliotek, co powodowało, że niestety konieczne było pozbycie się buzzera, który wydawał dźwięk podczas przykładania karty, ale finalnie wyszło to na dobre, ponieważ nie było konieczne użycie rejestru przesuwanego, ponieważ po pozbyciu się buzzera zwolnił on ostatni potrzebny pin, do którego został podłączony przekaźnik od żarówki. Usprawnieniami których można by dokonać jest przede wszystkim ulepszenie komunikacji zdalnej, tak aby nie odbywała się ona na podstawie modułu podczerwieni, tylko na przykład przekaźnika i odbiornika fal radiowych. Dodatkowo wszystko można by umieścić i przylutować do płytki PCB w celach większej kompaktowości. Z mniej istotnych rzeczy można by wymienić na przykład dodanie większej ilości kolorów taśmy LED, użycie mocniej świecącej żarówki, a do tego potrzebny byłby mocniejszy zasilacz.

[Humorystyczny film prezentujący wszystkie funkcjonalności projektu](#)

<https://youtu.be/JAgjxOQAriw>

## Literatura

1. [Arduino RFID Sensor](#)
2. [Schemat sterownika L293D zaczerpnięta z tej strony](#)
3. [Funkcjonowanie przekaźników](#)
4. [Podczerwień](#)

dostęp do stron na dzień 30.11.2020

## Załączniki

zawartość załącznika:

Kod źródłowy (z komentarzami)

Schematy

Prezentacja.

**WŁASNA OCENA PROJEKTU: 4.5**

**OCENA PROJEKTU Z FILMEM KOŃCOWYM: 5**