



Politechnika  
Wrocławska



Politechnika Wrocławska

# Urządzenie do nakładania powłok grafenowych na powierzchnię roboczą ciągnadła stalowego

Interdyscyplinarny projekt zespołowy 2023/24  
Mechatronika - II stopień

**Prowadzący Projektu:**

Dr hab. inż. Zbigniew Zimniak

**Autorzy:**

Jakub Daroń

Michał Dubaniewicz

Piotr Grabowski

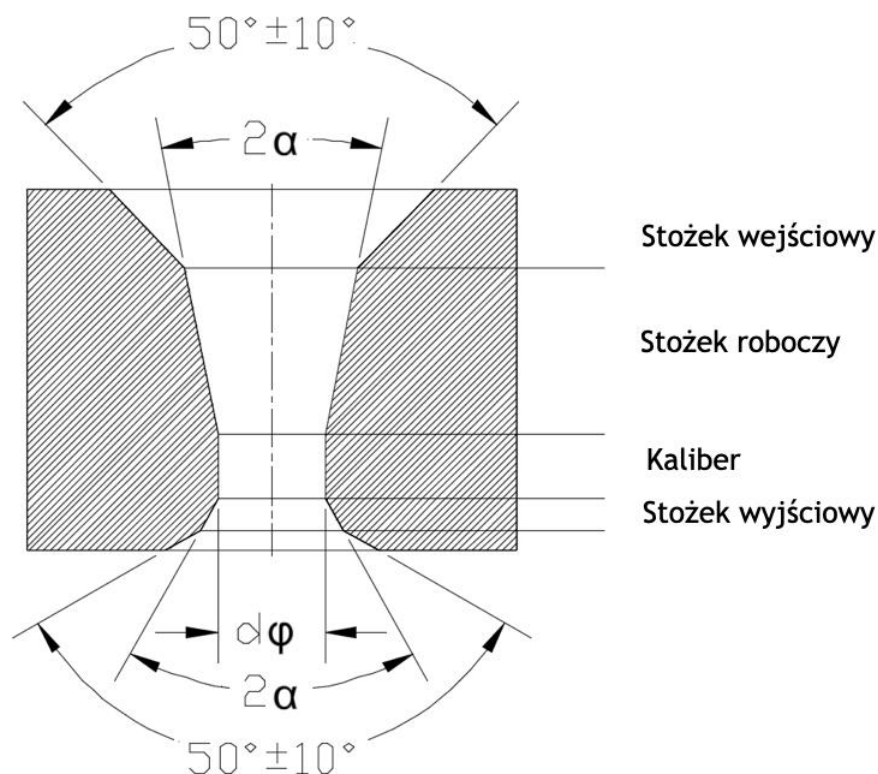
Wojciech Kargul

Jan Konior

Mateusz Lisoń

## 1. Wstęp

W procesie ciągnięcia drutów na ciągarkach wykorzystywane narzędzia – ciągadła – są poddawane są znaczącym siłom i z czasem degenerują pod wpływem ścierania. Z tego tytułu stosuje się materiały wysoce odporne na ścieranie do pokrycia powierzchni stożka roboczego. Typowe stosowane materiały to diament i węgliki spiekane. Na Rys. 1 przedstawiono schemat budowy ciągadła.



Rysunek 1 - schemat budowy ciągadła

Prowadzący projektu, dr hab. inż. Zbigniew Zimniak w swoim patencie nr PL 239477, zatytułowanym „Sposób wytwarzania trwałych cienkowarstwowych pokryć grafenowych na metalach” opisał metodę wytwarzania pokryć cienkowarstwowych z grafenu na metalach. Jak wiadomo grafen jest materiałem wysoce odpornym na ścieranie, w ramach projektu postanowiono zatem stworzyć urządzenie które pozwoli wykorzystać tą metodę do wykonywania powłok na elementach roboczych ciągadła stalowego.

W związku z tym określono następujące założenia projektowe:

- Minimalizacja kosztów – ze względu na brak funduszy wszystkie środki pochodzić będą z kieszeni wykonawców, określono budżet na poziomie ok. 600 zł.
- Maksymalizacja wykorzystania elementów typowych i wykonanych metodą druku 3D – ze względu na charakter naukowy budowanego urządzenia i niski budżet pewne elementy mogą wymagać wymiany lub modyfikacji z czasem. Zastosowanie druku 3D i elementów typowych ułatwi pracę w tego typu sytuacjach.
- 1 oś przesuwu liniowego pozwalająca uzyskać nacisk ok. 2kg, siła nacisku musi być sterowalna w pewnym zakresie – Parametry nacisku są powiązane z wymaganiami procesu nakładania powłok.

- 1 oś obrotowa – umożliwia obrót cięgadła i podział obrotu na x cykli w celu powtórzenia operacji pokrywania w innych pozycjach.
- Operacja półautomatyczna – ze względu na to że zasilacz generujący wyładowanie elektryczne jest stosowany również w innych instalacjach zdecydowano że nie będzie on sterowany z urządzenia tylko wyzwalany ręcznie.
- Sterowanie za pomocą aplikacji okienkowej – ze względu na ograniczanie kosztów zrezygnowano z fizycznego sterowania.

## 2. Przegląd Rozwiązań

Ze względu na to że urządzenie nie realizuje szeroko rozpowszechnionego procesu, nie ma rozwiązań które można wykorzystać jako bezpośrednie odniesienie. Z tego powodu poszczególne elementy projektu odnoszono do różnych urządzeń.

W przypadku osi przesuwu liniowego, jednym z bardziej powszechnych rozwiązań tego typu są drukarki 3D. Wykorzystują one śruby trapezowa lub kulowe drobnozwojne w celu uzyskania dokładności pozycjonowania. Całość jest napędzana napędami krokowymi. Przykład takiego urządzenia przedstawiono na Rys. 2.



Rysunek 2 - Przykładowa drukarka 3D (Prusa MINI+)

Innym przykładem urządzenia które przewijało się jako potencjalna inspiracja w przypadku przesuwu liniowego jest wiertarka kolumnowa, ze względu na to że są to konstrukcje przenoszące duże obciążenia. Na Rys. 3 przedstawiono przykładową konstrukcję.



*Rysunek 3 - przykładowa wiertarka kolumnowa*

Ostatecznie, w przypadku projektowanego urządzenia konieczne będzie częściowe połączenie konkretnych cech obu rozwiązań – precyzji drukarek 3D i wytrzymałości którą cechuje się konstrukcja wiertarki kolumnowej.

W przypadku osi obrotowej, ciekawą inspiracją były stoły obrotowe do frezarek CNC (Rys. 4). Urządzenia te pozwalają uzyskać obrót o zadany kąt i są sterowane. Kwestią którą należy rozwiązać w celu adaptacji tego typu rozwiązania do projektu jest rozmiar i metoda mocowania obiektu w sposób zapewniający centryczność cięgadła i elektrody.



*Rysunek 4 - przykładowy stół obrotowy do frezarki CNC*

W przypadku pomiaru nacisku, na rynku są dostępne 2 główne rodzaje rozwiązań – czujniki membranowe i belki tensometryczne (Rys. 5). Belki tensometryczne pozwalają na uzyskanie odczytów dla dużych nacisków (powyżej 10 kg), natomiast zajmują dużo miejsca i w zastosowaniu takim jak w projekcie konieczne byłoby wykorzystanie minimum 2 takich belek. Czujniki membranowe z kolei są nieco bardziej wrażliwe na punkt przyłożenia siły i z reguły mają niższe zakresy pracy.



Rysunek 5 - a.) belka tensometryczna b.) czujnik membranowy

### 3. Sekcja Mechaniczna

#### a. Ruch w osi pionowej

Pierwszym zagadnieniem mechanicznym do realizacji było zapewnienie ruchu elektrody w osi pionowej, jednocześnie z możliwością generowania jej docisku do ciążadła z siłą odpowiadającą wartości 2 [N]. W tym celu zamówiono prowadnicę liniową THK KR20A 200mm CNC o skoku śruby wynoszącym 1[mm] (Rys. 6). Ruch obrotowy śruby będzie generowany za pomocą silnika krokowego JK42HS40 – 0404, który zostanie szerzej przedstawiony w sekcji elektronicznej.



Rysunek 6 - Prowadnica liniowa THK KR20A 200 mm CNC

Szkieletem całego układu jest profil aluminiowy, do którego przymocowana została prowadnica oraz mocowanie stalowe typu L do silnika krokowego (Rys. 7). Napęd na prowadnicę przenoszony będzie za pomocą sprzęgła osiowo elastycznego (Rys. 8).



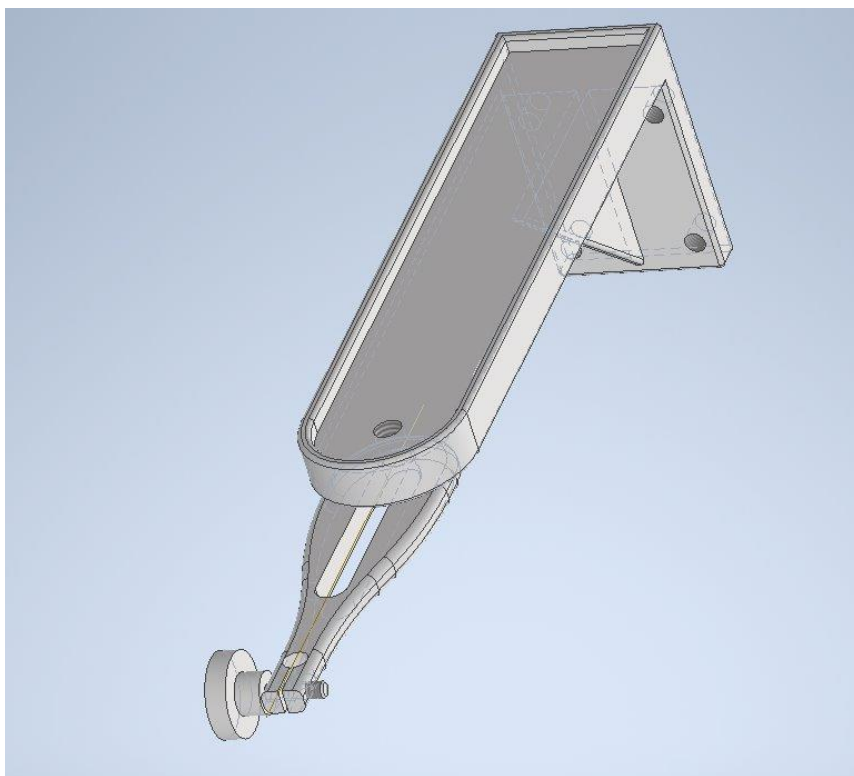
*Rysunek 7 - Stalowe mocowanie silnika typu L*



*Rysunek 8 - Sprzęgło osiowo elastyczne do silnika krokowego*

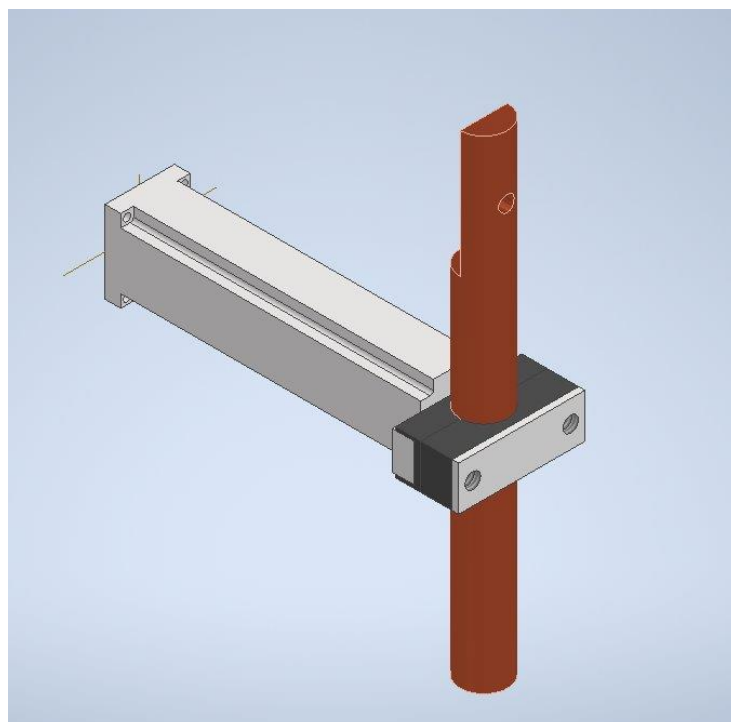
### **b. Koncepcje ramienia trzymającego elektrodę**

Kolejnym założeniem do realizacji była stabilizacja, przełożenie ruchu w osi pionowej na elektrodę oraz utrzymanie jej w osiowości z ciągadłem. W tym celu zaprojektowano ramię, które zostanie połączone z prowadnicą oraz utrzyma elektrodę w odpowiedniej pozycji. Pierwsza koncepcja ramienia zakładała, że w pełni zostanie ono wydrukowane za pomocą drukarki 3D, ze względu na fakt, że elektroda musi zostać odizolowana od reszty układu, ponieważ ma przez nią przepływać prąd. Założono, że ramię będzie składać się z dwóch elementów, dzięki czemu możliwa była regulacja jego długości. Zamodelowane rozwiązanie przedstawiono na Rys. 9.



*Rysunek 9 - Pierwsza koncepcja ramienia*

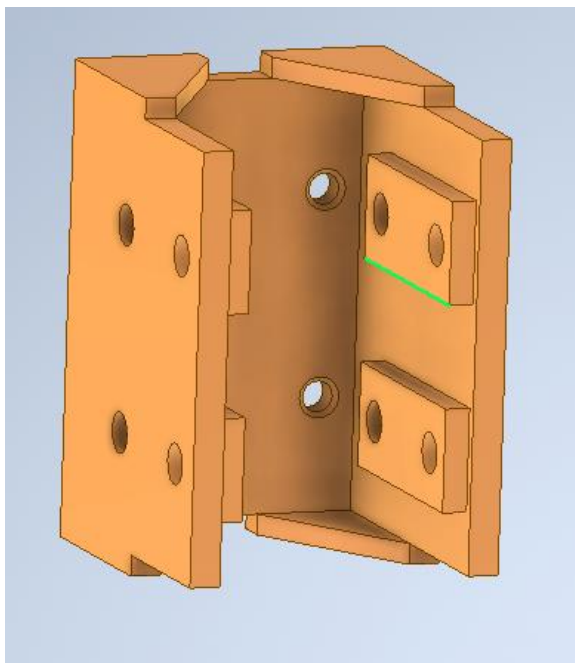
Szybko jednak zrezygnowano z tego pomysłu na dużą podatność tego układu, co nie zapewniało odpowiedniej stabilności oraz wytrzymałości. Wtedy zrodziła się myśl wykorzystania profilu aluminiowego jako ramienia, aby bardziej ustabilizować elektrodę. Tak więc powstało nowe rozwiązanie przedstawione na Rys. 10.



*Rysunek 10 - Druga koncepcja ramienia*

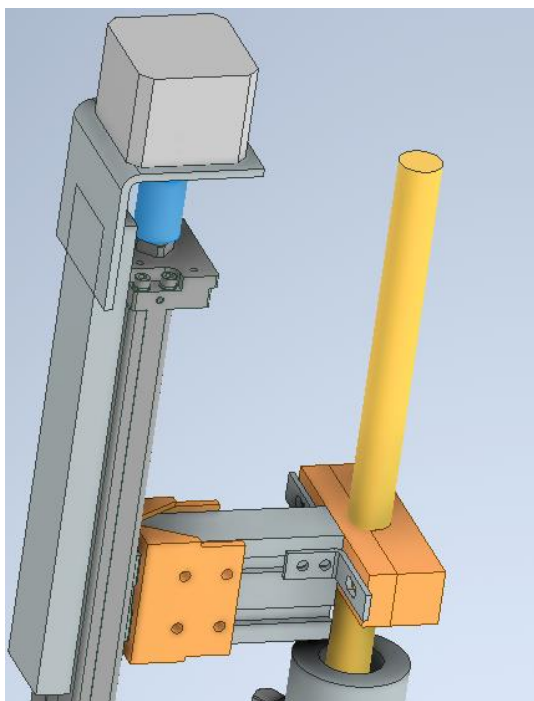


Pozostał wówczas problem z zamocowaniem profilu do prowadnicy. W tym celu zaprojektowano specjalistyczny uchwyt wydrukowany na drukarce 3D (Rys. 11).



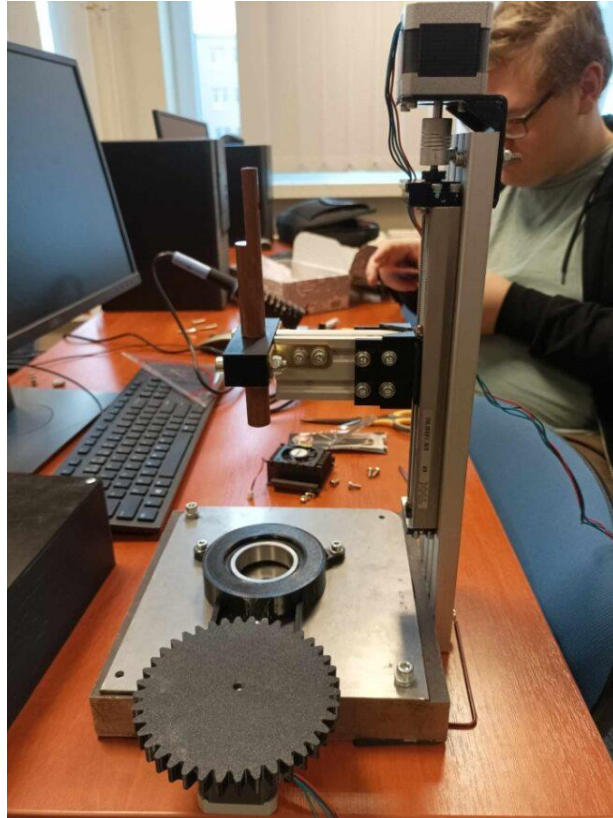
*Rysunek 11 - Mocowanie ramienia do prowadnicy*

W ostateczności model został zrewidowany tak aby uwzględnić zaprojektowane mocowanie. Bazując na tym projekcie wykonano fizyczne ramie. Model po poprawkach przedstawiono na Rys. 12, a fizyczną realizację na Rys. 13.



*Rysunek 12 - Finalna koncepcja ramienia*





Rysunek 13 - Realizacja ramienia z prowadnicą zamocowana na ramie układu

### c. Stół obrotowy pod ciągadło

Ostatnim zagadnieniem do rozpatrzenia było umiejscowienie ciągadła oraz umożliwienie mu obrotu. W tym celu zaprojektowano stół obrotowy, którego ruch wykonywany będzie za pomocą takiego samego silnika krokowego, jak w przypadku prowadnicy. Moment obrotowy przenoszony będzie za pomocą przekładni zębatej. Przętożenie obliczono jako stosunek liczby zębów obu kół zębatych, korzystając ze wzoru:

$$i = \frac{z_2}{z_1}$$

gdzie:

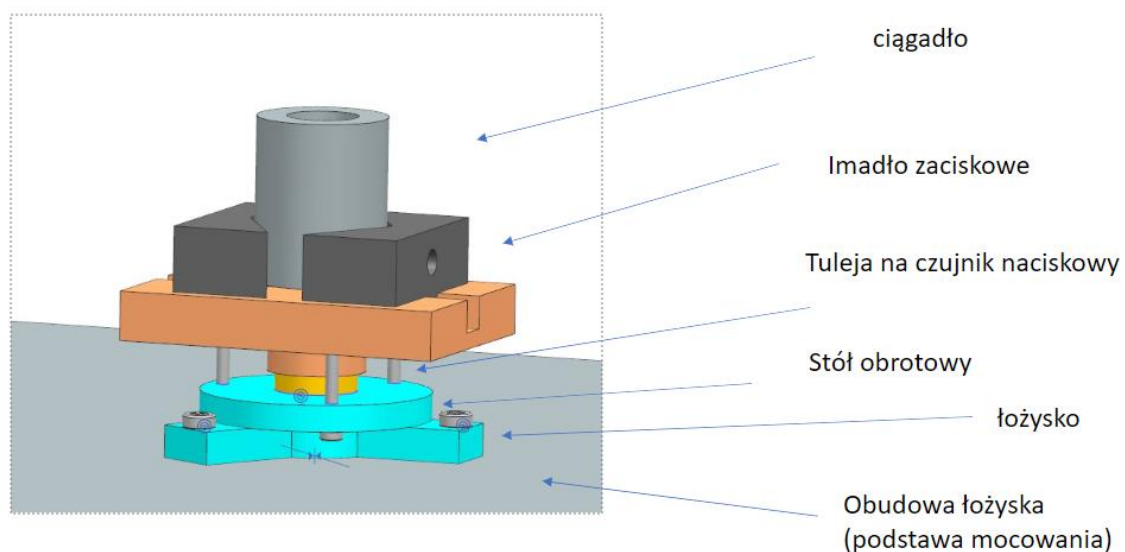
$z_1$  – liczba zębów koła napędzającego

$z_2$  – liczba zębów koła napędzanego

W wyniku czego otrzymano następujące przętożenie:

$$i = \frac{64}{40} = 1,6$$

Początkowa koncepcja stołu została przedstawiona na Rys 14.



*Rysunek 14 Początkowa koncepcja mocowania ciągadła*

W dalszych etapach rozwoju koncepcji, ze względu na dostępność i ułatwienie centrowania osiowego względem elektrody, zrezygnowano z imadła zaciskowego, a w jego miejsce pojawił się uchwyt trójszczękowy (Rys 15).



*Rysunek 15 - Ciągadło w uchwycie trójszczękowym*

Układ wyposażony został w łożysko kulkowe jednorzędowe 6008 2RS (Rys. 16), którego zadaniem jest przede wszystkim przenoszenie obciążeń osiowych, spowodowanych dociskiem elektrody na ciągadło.



### Łożysko kulkowe zwykłe 6008 2RS F&T 40x68x15

Indeks: L0101-001-1115

Kategoria: Łożyska kulkowe zwykłe

Waga: 0.2 kg

#### Parametry:

Średnica wewnętrzna d [mm]: 40

Średnica zewnętrzna D [mm]: 68

Szerokość B [mm]: 15

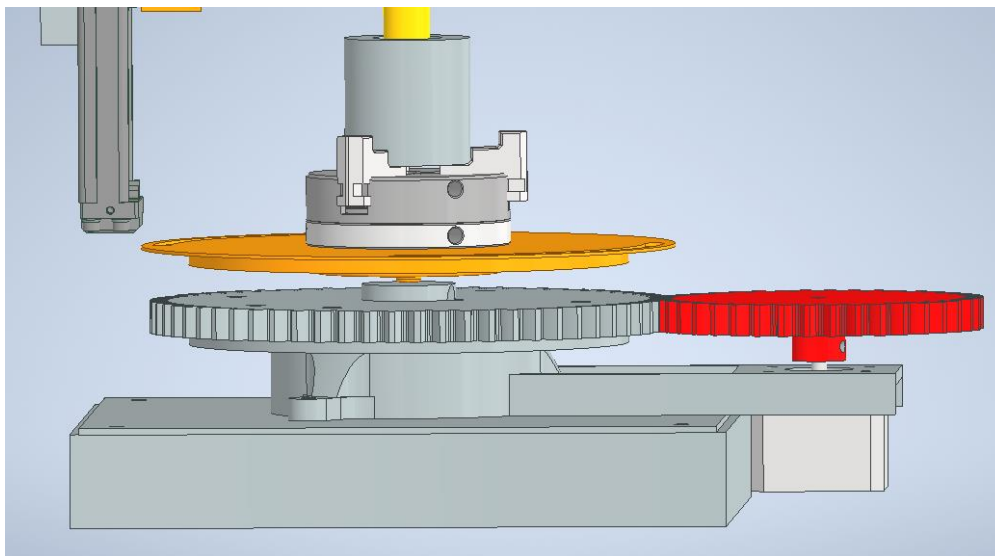
Zabudowa: 2RS

Luz: normalny

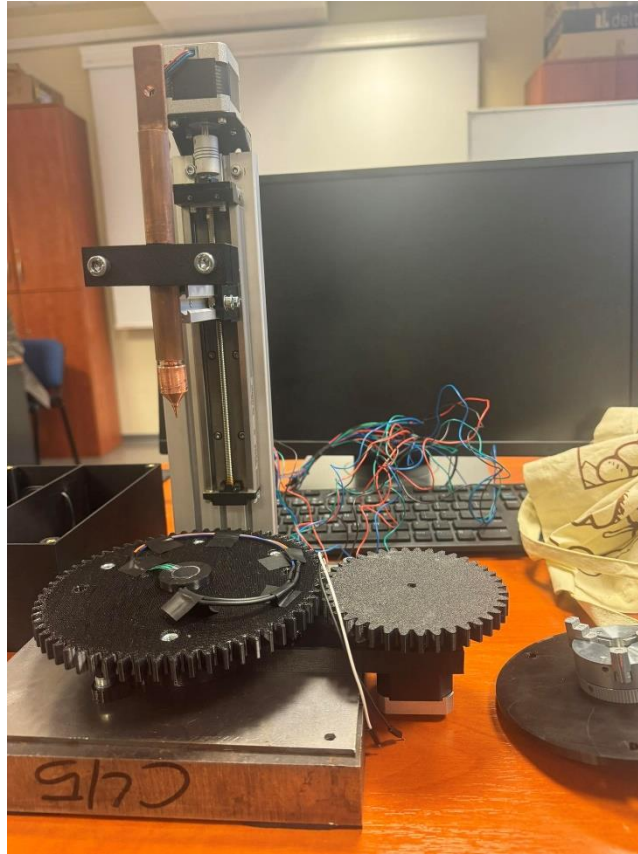
Zdjęcia poglądowe, mogą różnić się od stanu rzeczywistego.

Rysunek 16 - Łożysko kulkowe jednorzędowe 6008 2RS

Na Rys. 17 przedstawiono model finalnej koncepcji stołu obrotowego wraz z pełną przekładnią zębatą. Na Rys 18 przedstawiono realizację tego konceptu.



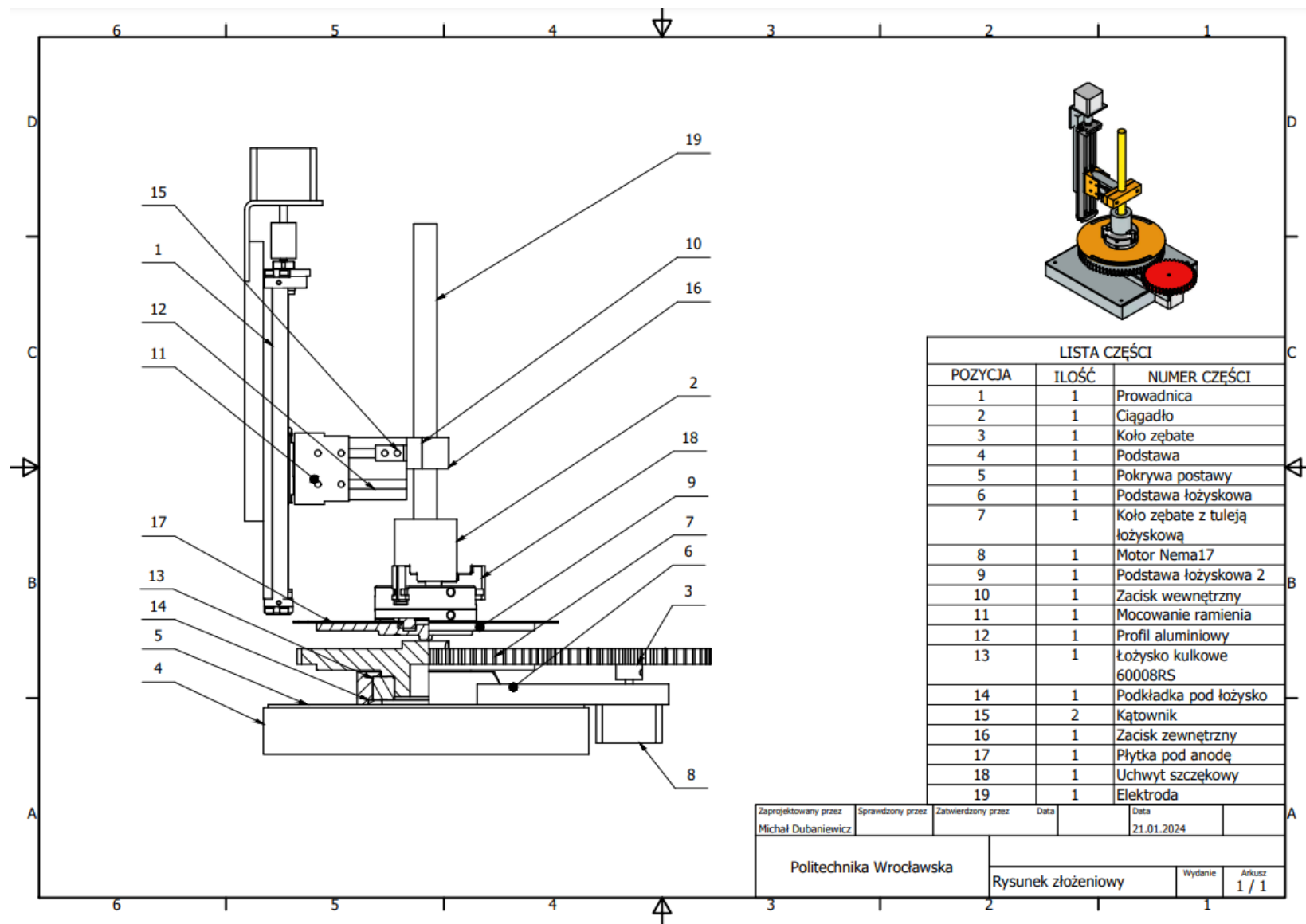
Rysunek 17 - Finalna koncepcja stołu obrotowego wraz przekładnią zębatą



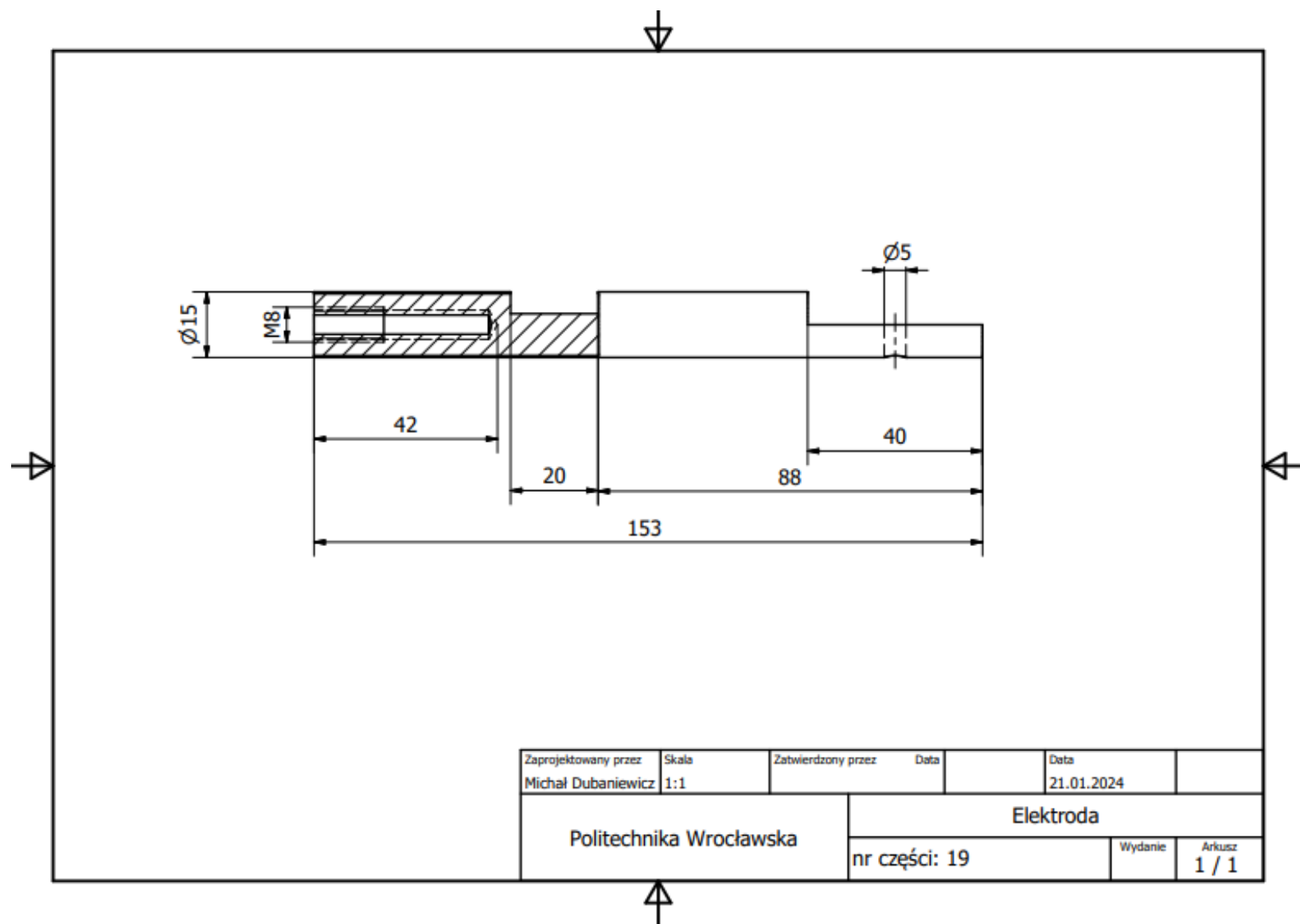
*Rysunek 18 - Realizacja fizyczna stołu obrotowego*

#### **d. Części niestandardowe**

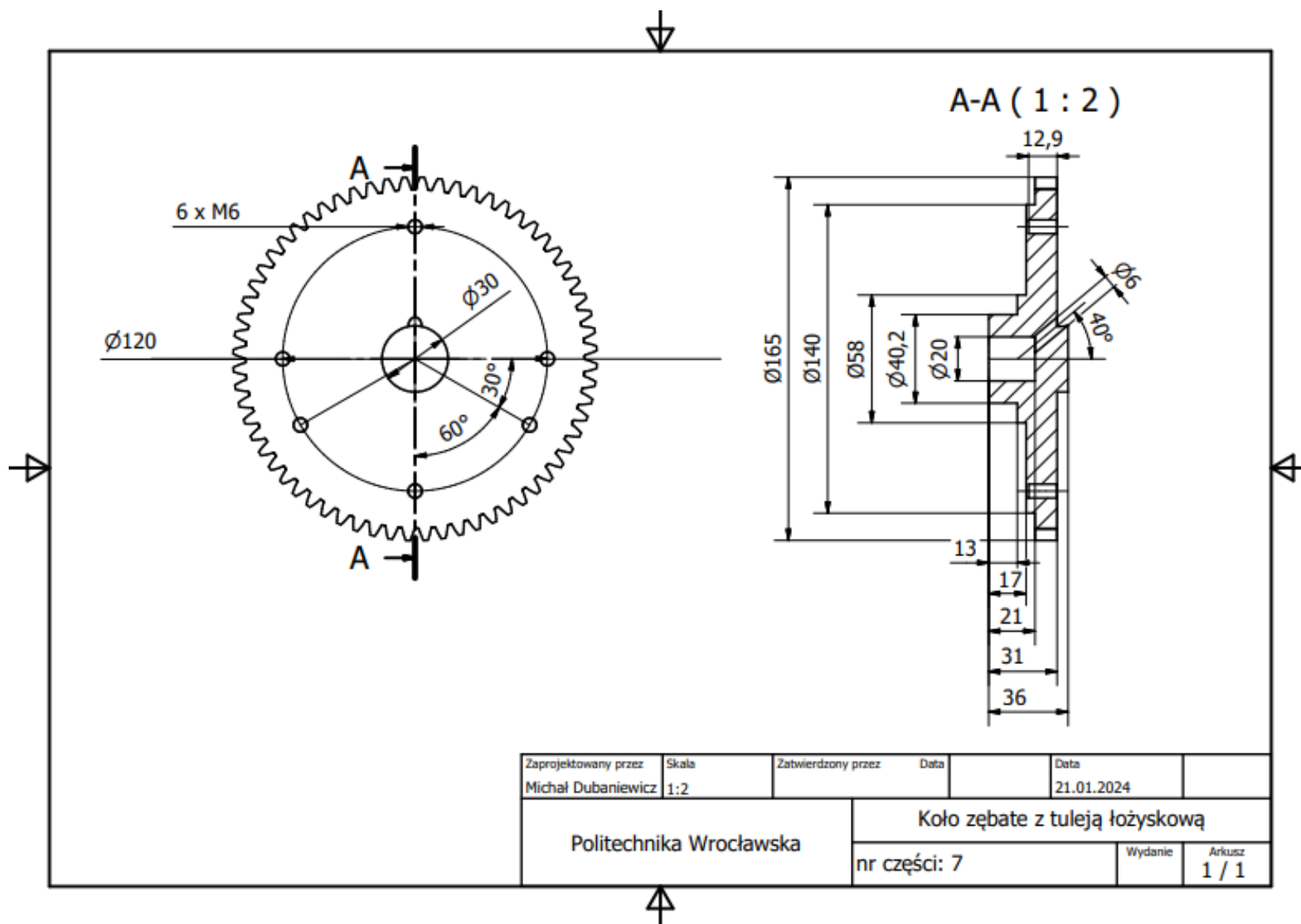
Ze względu na specyficzne cechy wymagane w projekcie sporo części zostało wykonane wg. dedykowanego projektu. Duża część elementów została wykonana za pomocą druku 3D. Na Rys. 14 przedstawiono rysunek złożeniowy układu. Poniżej, na Rys. 19 – Rys. 29 przedstawiono rysunki wykonawcze części niestandardowych.



Rysunek 19 - Rysunek złożeniowy układu

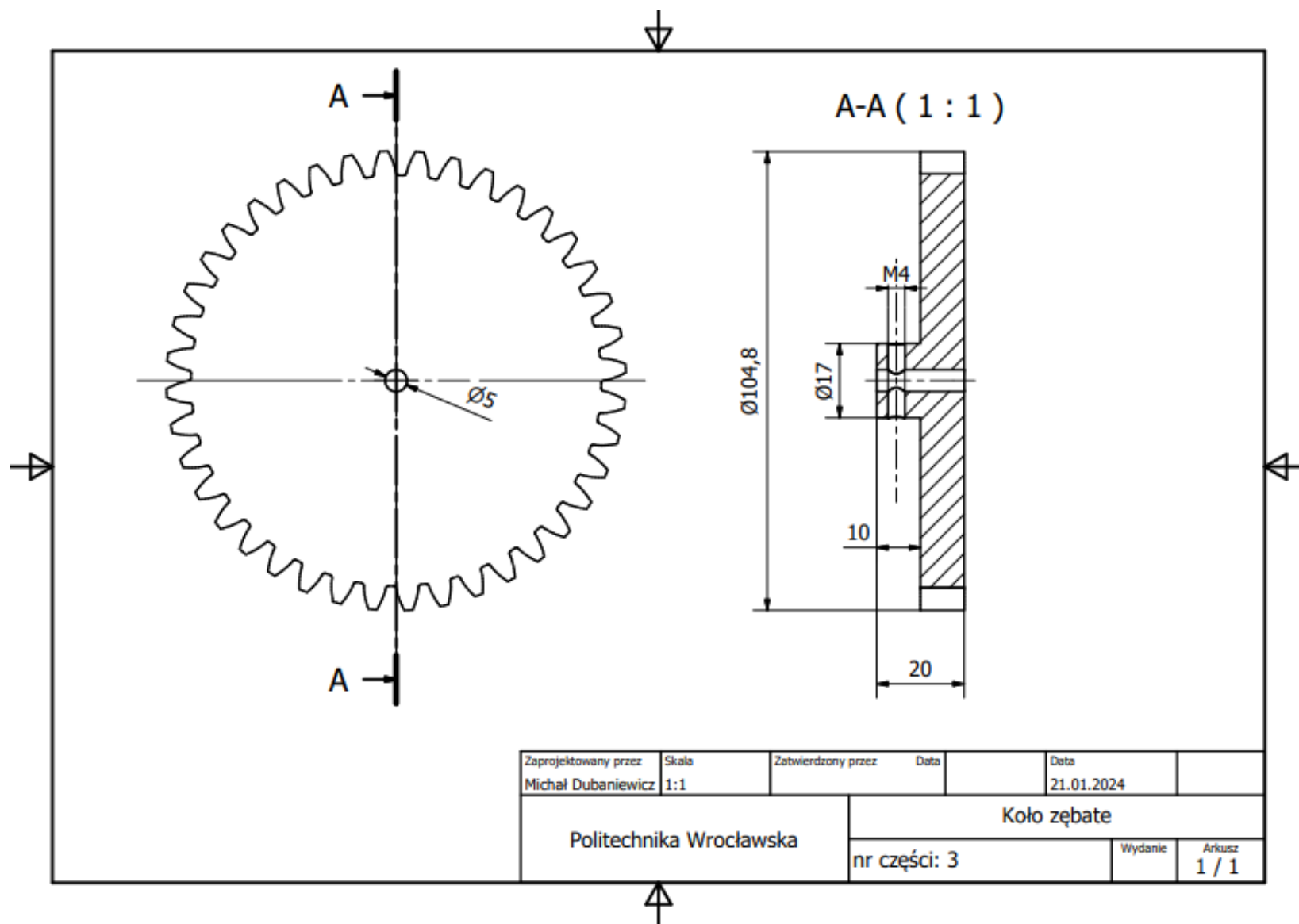


Rysunek 20 - Rysunek wykonawczy trzonu elektrody

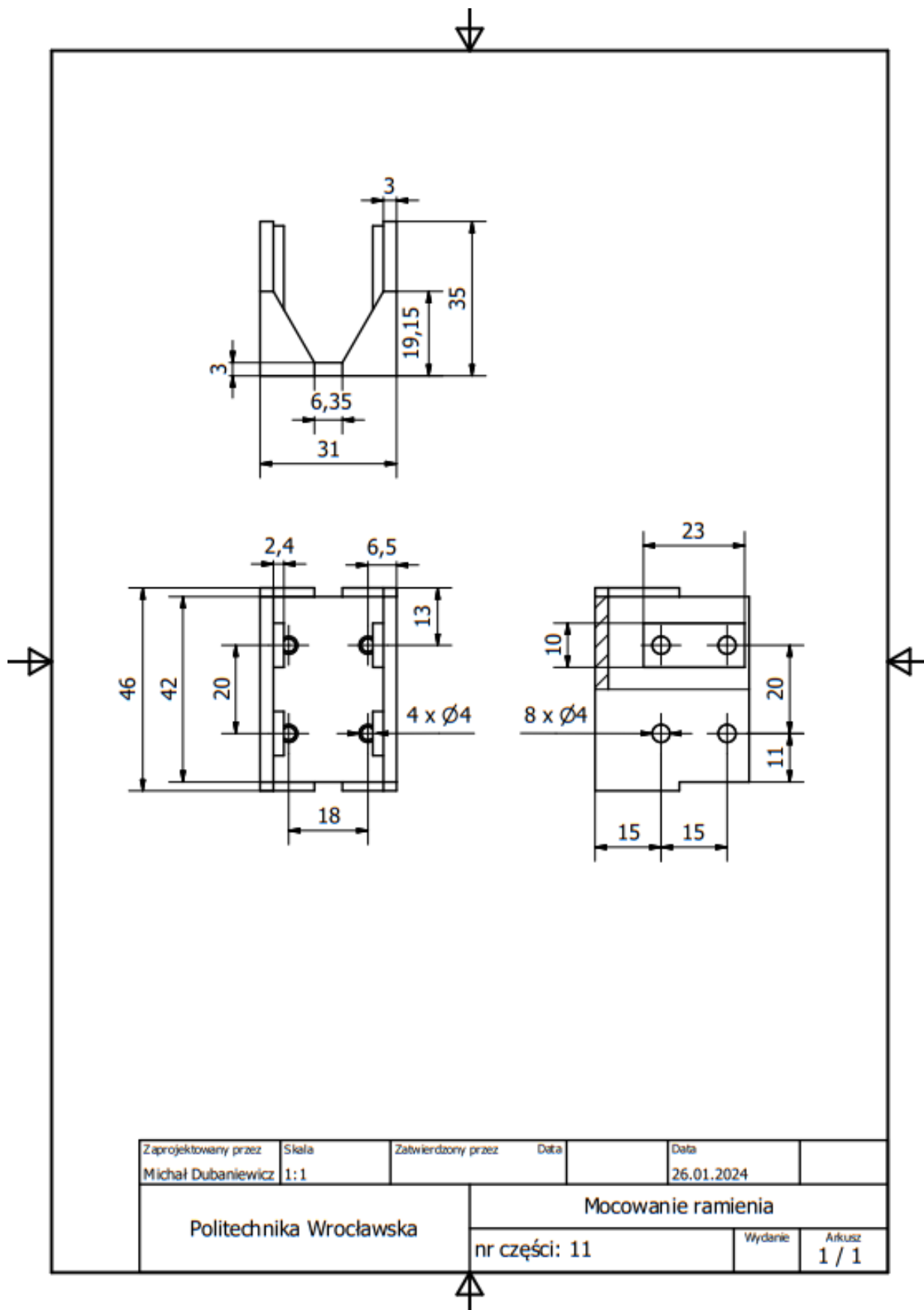


Rysunek 21 - Rysunek wykonawczy koła zębatego z tuleją łożyskową

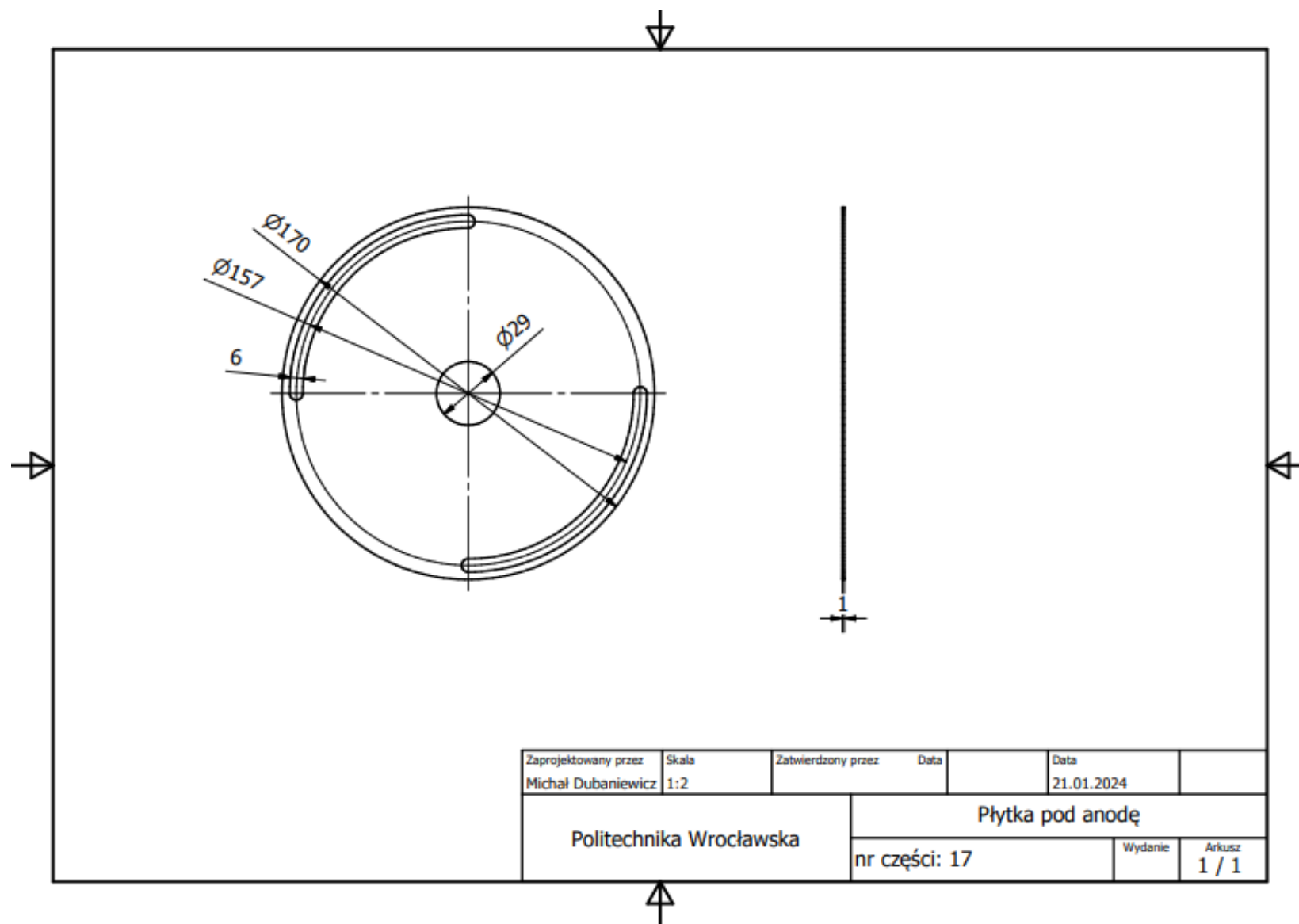




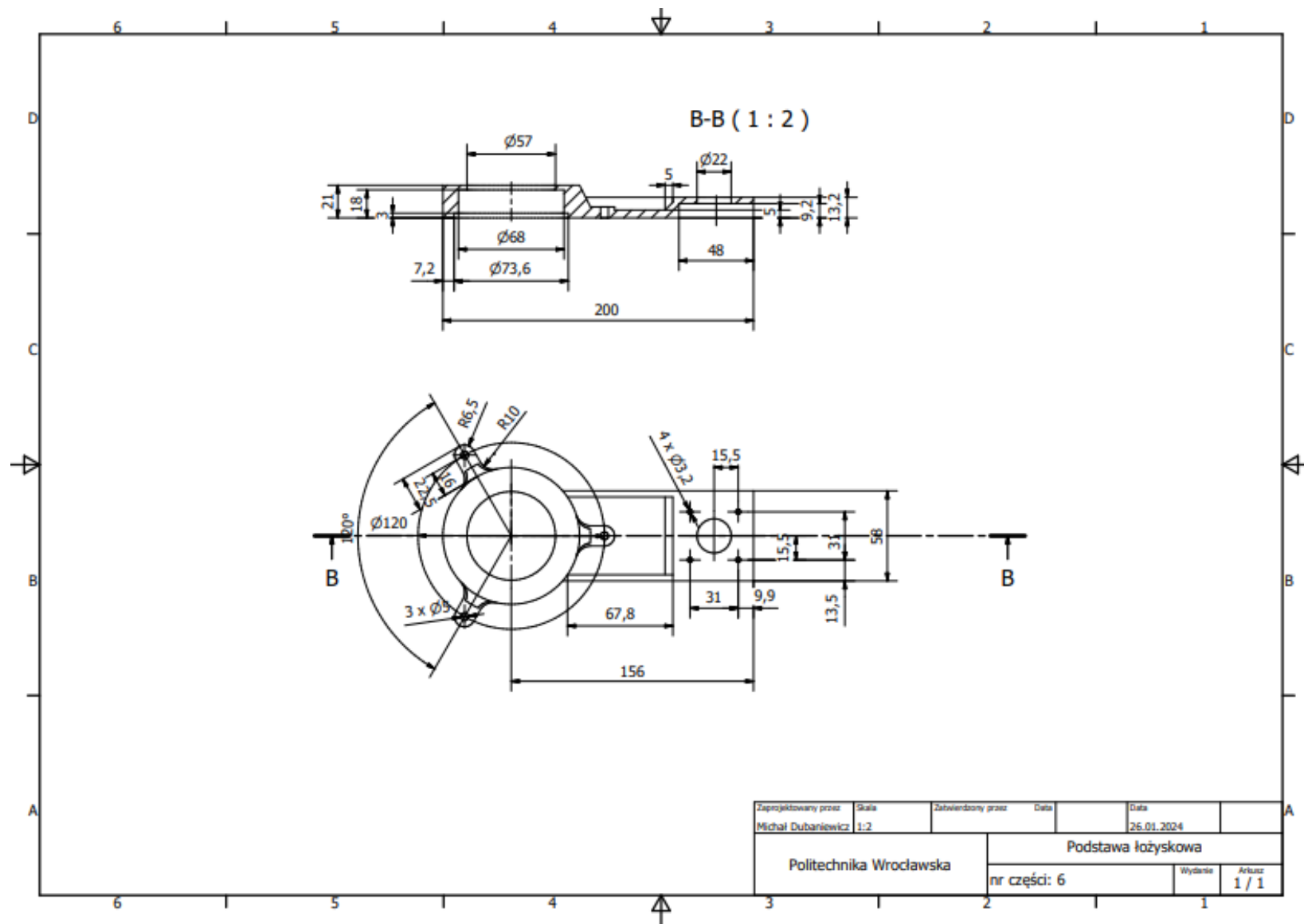
Rysunek 22 - Rysunek wykonawczy koła zębatego



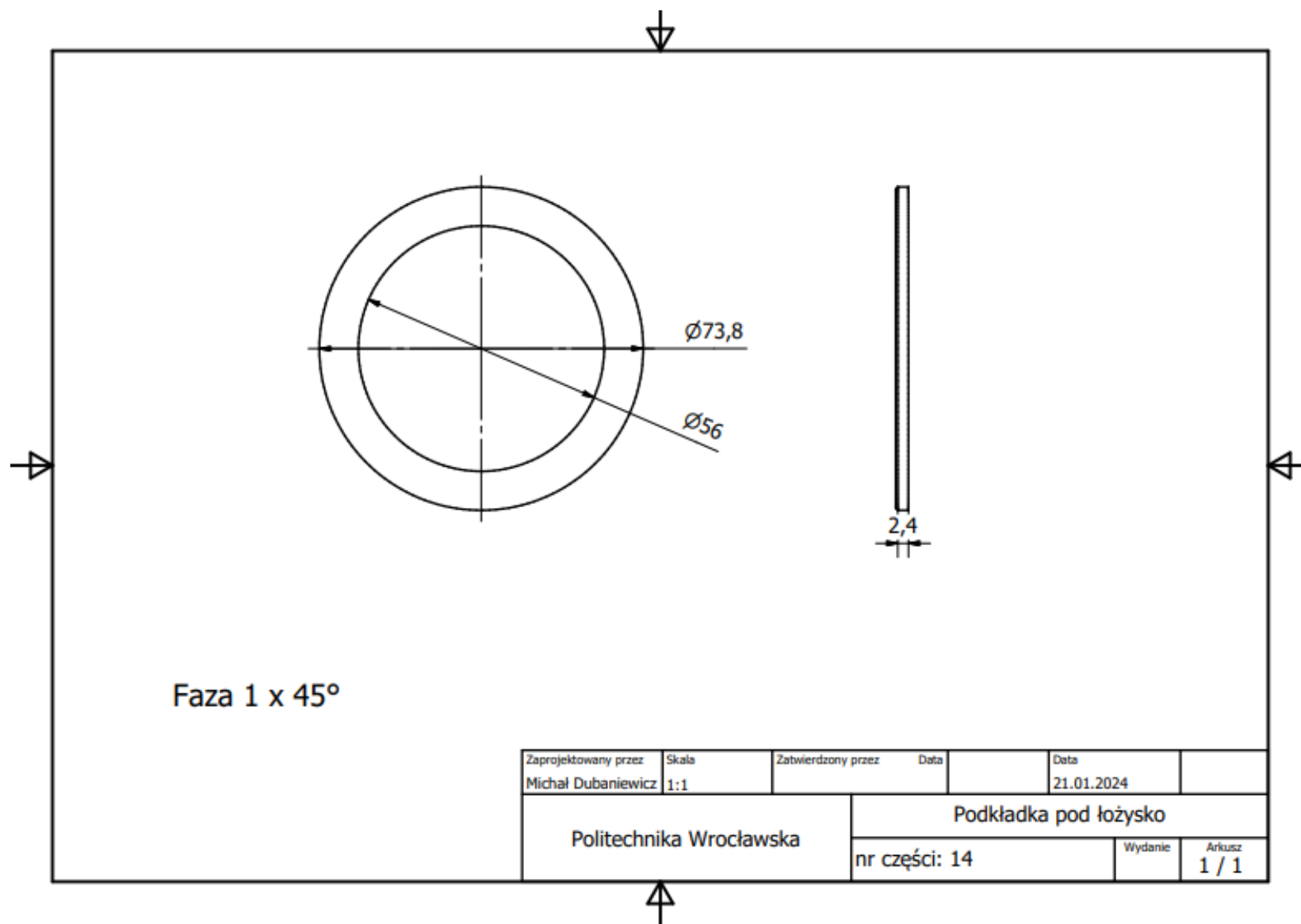
Rysunek 23 - Rysunek wykonawczy mocowania ramienia



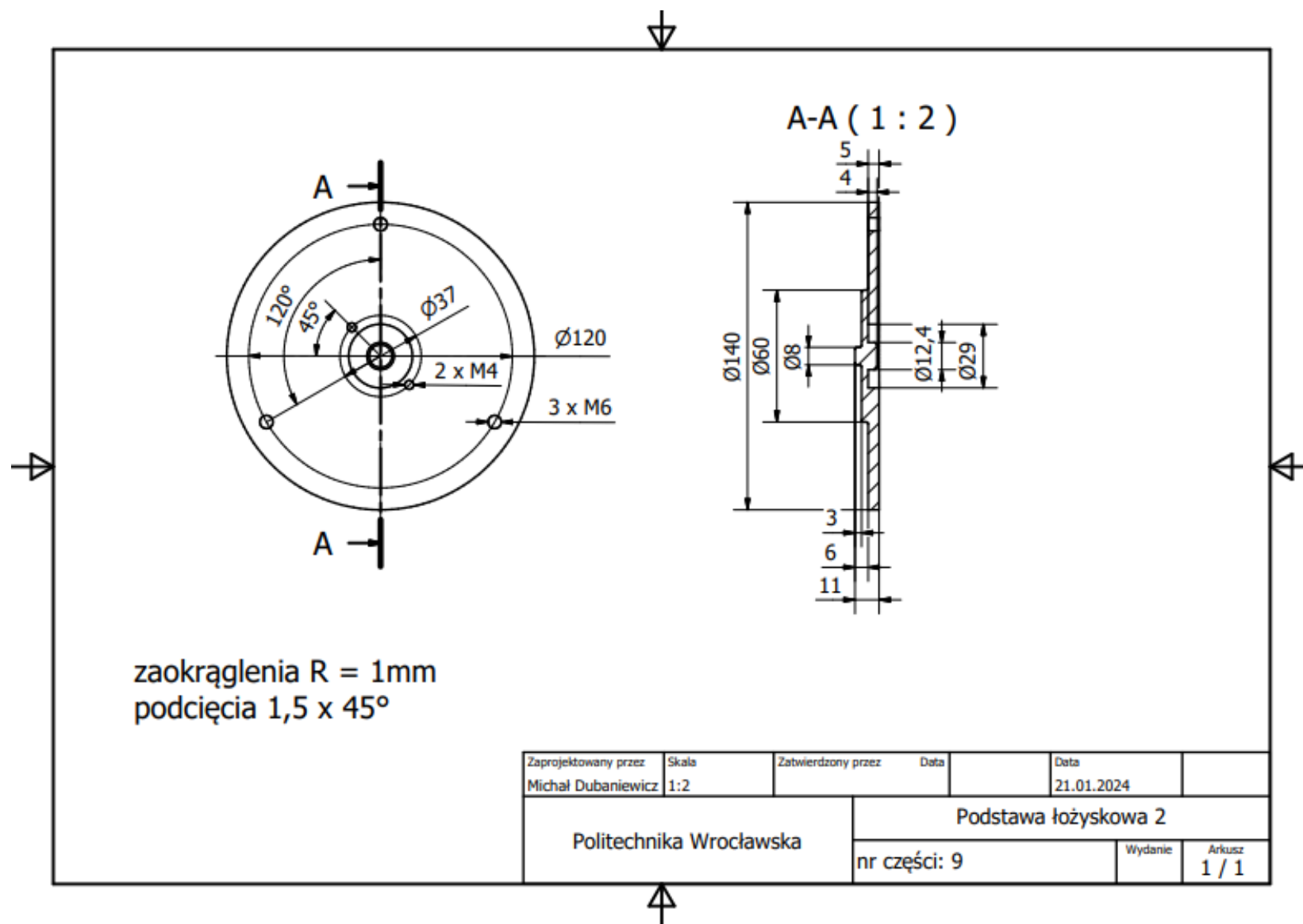
Rysunek 24 - Rysunek wykonawczy płytki montażowej do anody



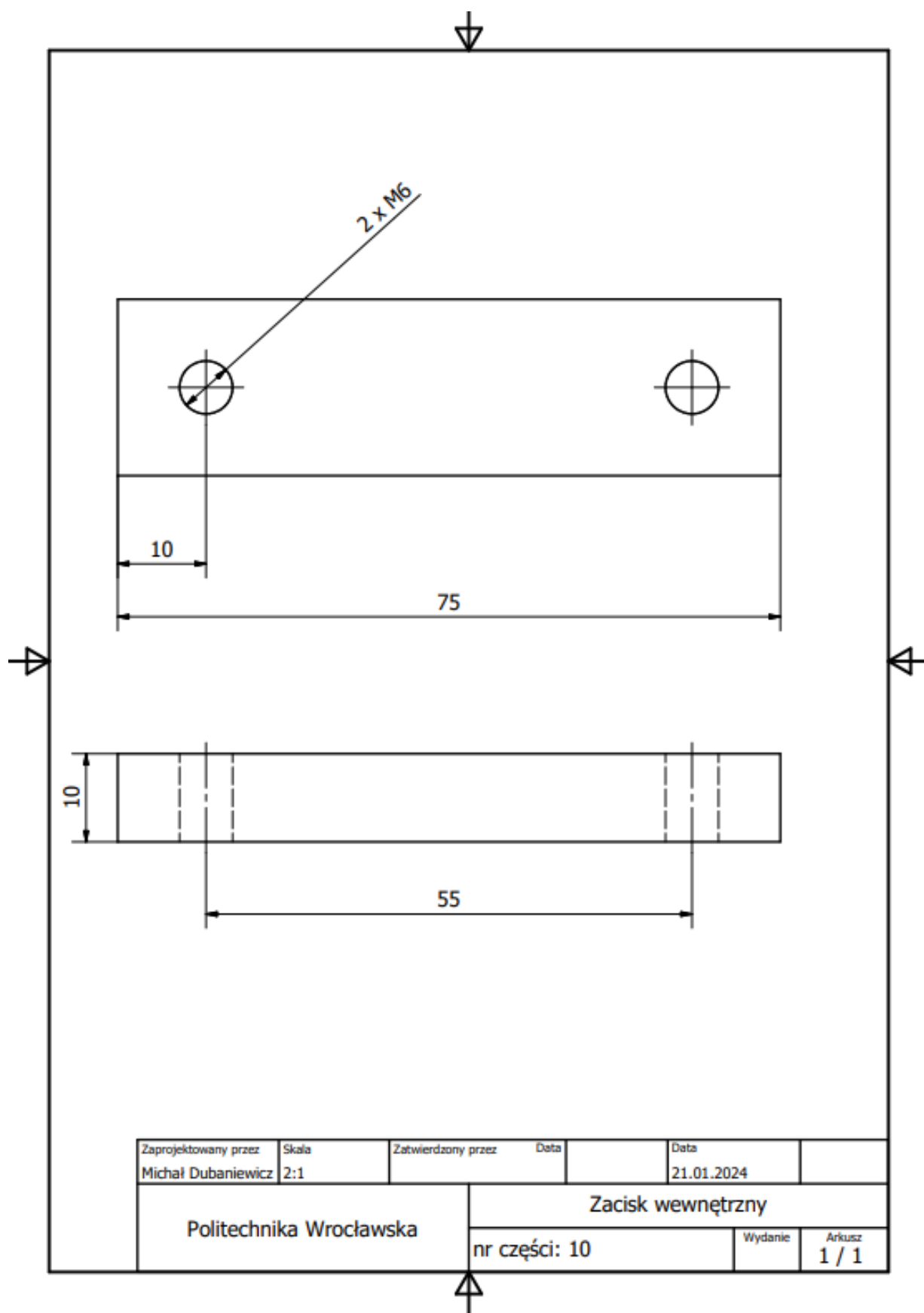
Rysunek 25 - Rysunek wykonawczy podstawy łożyskowej



Rysunek 26 - Rysunek wykonawczy podkładki pod łożysko

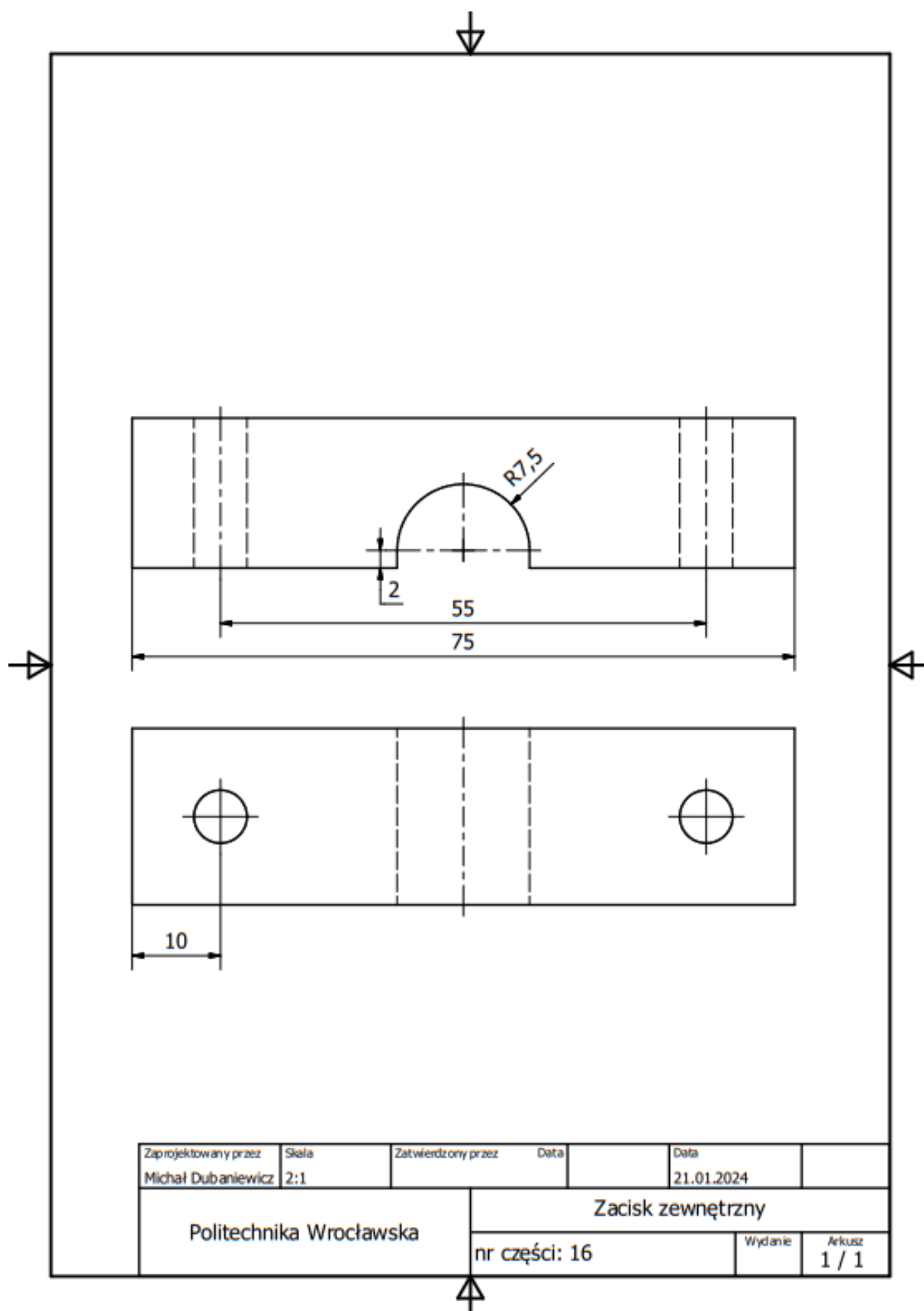


Rysunek 27 - Rysunek wykonawczy stolika



Rysunek 28 - Rysunek wykonawczy zacisku zewnętrznego elektrody



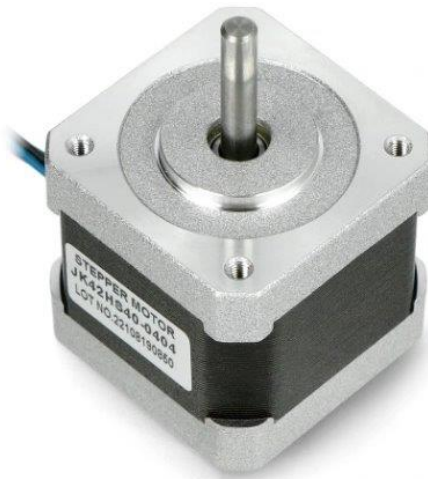


Rysunek 29 - Rysunek wykonawczy zacisku wewnętrznego elektrody

## 4. Sekcja Elektroniczna

### a. Układ elektroniczny

W celu realizacji pracy którą wykonywać ma urządzenie dobrano napędy z uwagi na niezbędne parametry mechaniczne. Dobrane zostały napędy krokowe JK42HS40-0404 (Rys. 30) które mają możliwość utrzymania momentu 0,25Nm i dokładność 200 kroków na obrót z możliwością uzyskania większego podziału za pomocą funkcji mikrostep. Napędy te zasilane są napięciem 12V i pobierając 0,4A na cewkę.



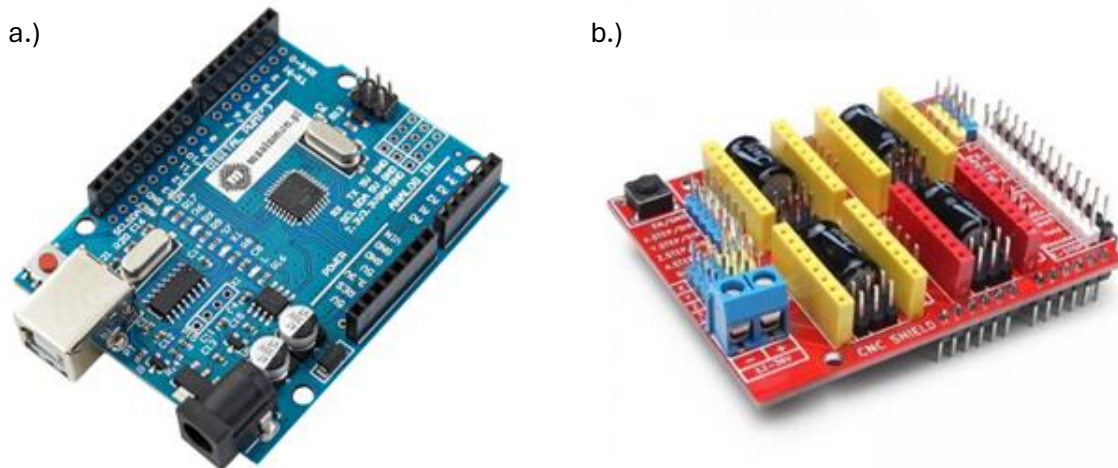
Rysunek 30 - silnik JK42HS40-0404

Bazując na niezbędnych wymaganiach z punktu widzenia mechaniki układu dobrano również czujnik nacisku. Wybrano czujnik CP15-151NS (Rys.31), jest to czujnik membranowy o zakresie pomiarowym na poziomie 0,2 – 100N, co sprawia że przy odpowiednim dobraniu zakresu pomiarowego możliwe będzie pomiar w nacisku przy sile ok. 2kg które ma być domyślnym naciskiem urządzenia.



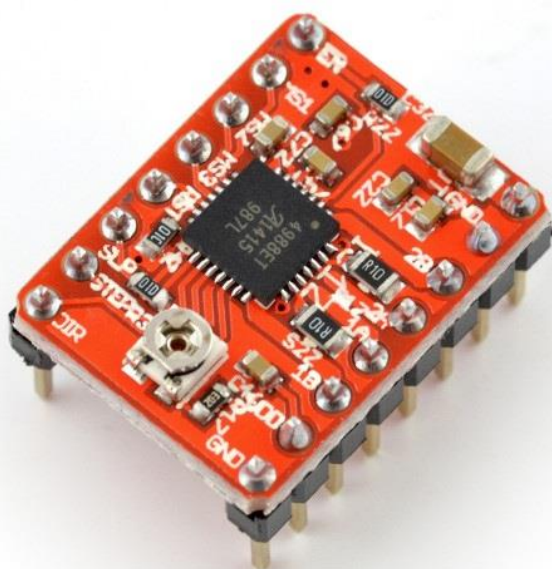
Rysunek 31 – Czujnik CP15-151NS

Pozostałe elementy dobrano kierując się głównie ceną i zgodnością parametrów elektrycznych. Do roli jednostki głównej wybrano Arduino UNO (Rys. 32) wraz z rozszerzeniem Arduino CNC Shield (Rys. 27) które jest wyspecjalizowanym, gotowym układem umożliwiającym sterowanie silnikami krokowymi, co oszczędza projektowania złożonych układów elektronicznych.



Rysunek 32 - a.) Arduino UNO b.) Arduino CNC Shield

Do pracy w CNC Shield konieczne są sterowniki silników krokowych. Zdecydowano się na wykorzystanie układów typu A4988 (Rys.33). Są to bardzo podstawowe konstrukcje, natomiast mają możliwość obsługi napędów pracujących na napięciach 8-35V i pobierających do 1A na cewkę oraz pozwalają na obsługę microstep do rozdzielczości 1/16 kroku. Jest to układ jak najbardziej wystarczający w tym przypadku.



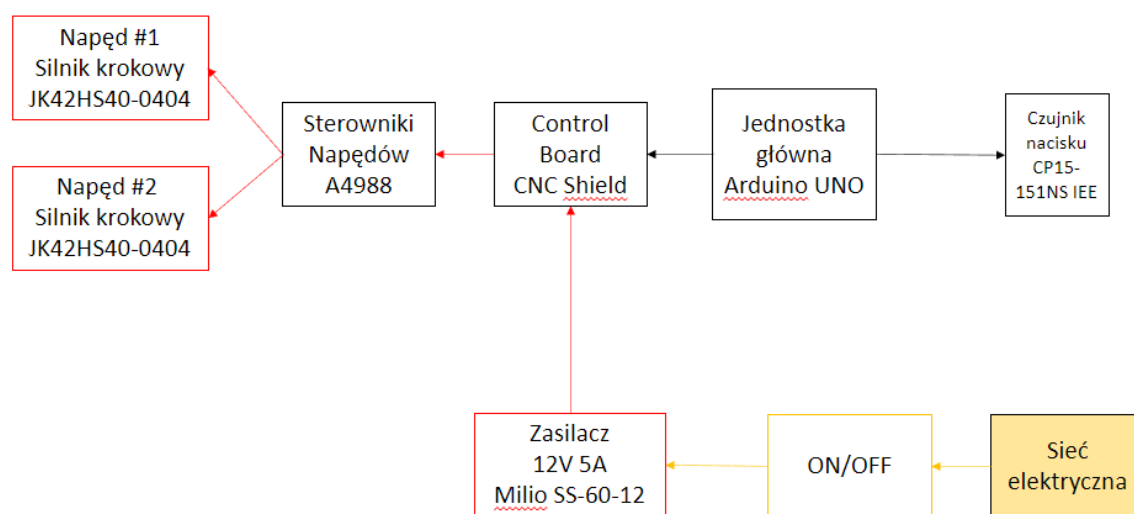
Rysunek 33 - Układ A4988

Dobry zasilacz musiał mieć możliwość zasilania napędów oraz posiadać odpowiedni zapas amperażu. Z tego powodu dobrano zasilacz Milio SS-60-12 (Rys. 34). Jest to prosta jednostka o mocy 60W oferująca na wyjściu prąd stały o napięciu 12VDC o natężeniu do 5A.



Rysunek 34 - Zasilacz Milio SS-60-12

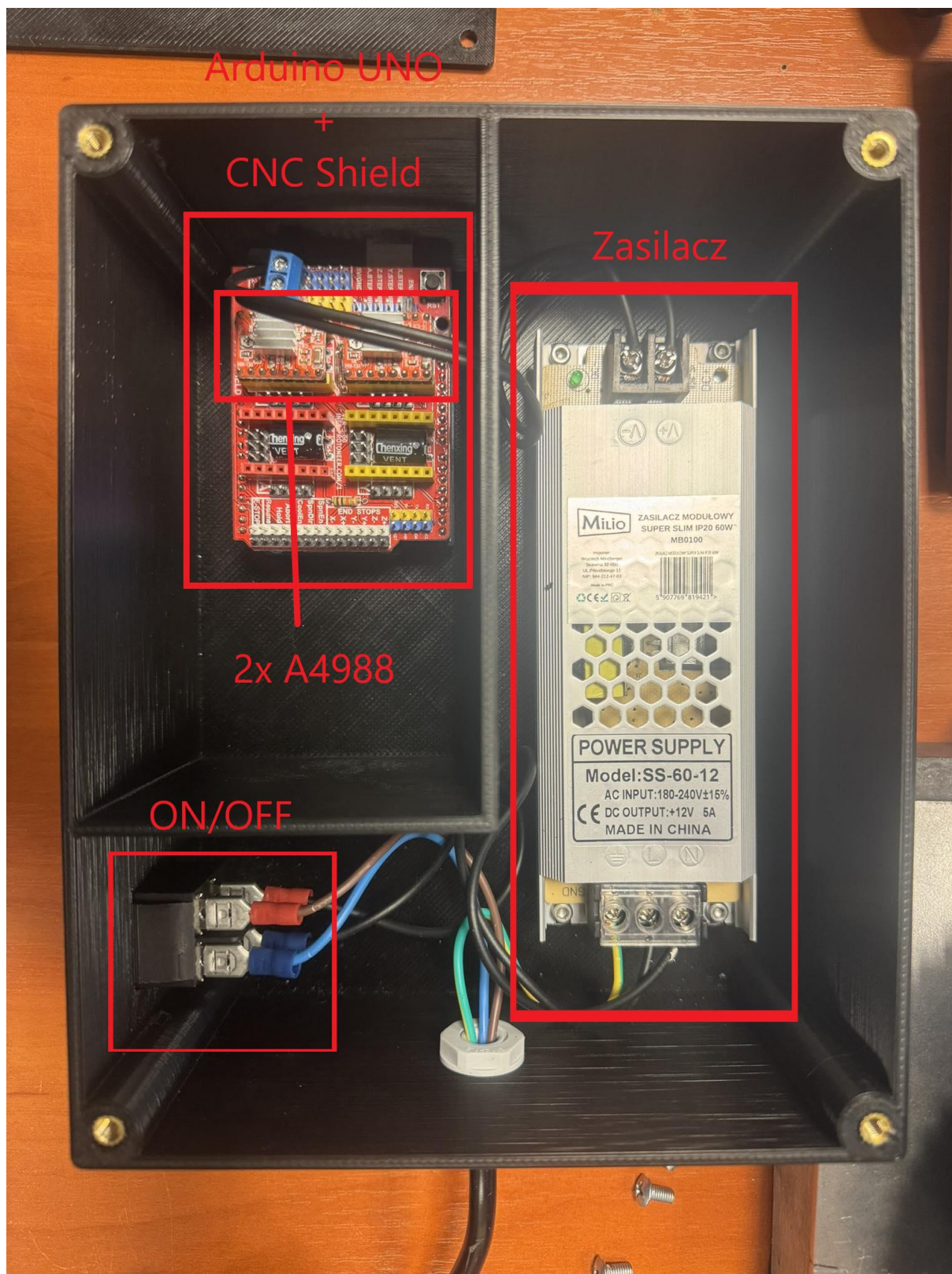
Całość została połączona zgodnie ze schematem na Rys 35. Jak można zobaczyć na schemacie, dodany został przycisk ON/OFF umożliwiający włączenie i wyłączenie układu. Należy zaznaczyć że ze względu na wybraną metodę obsługi (aplikacja okienkowa) zdecydowano się na nie podłączanie Arduino UNO do głównego zasilacza 12V. Będzie ono zasilane z potu USB, jako że podłączenie z komputerem jest wymagane.



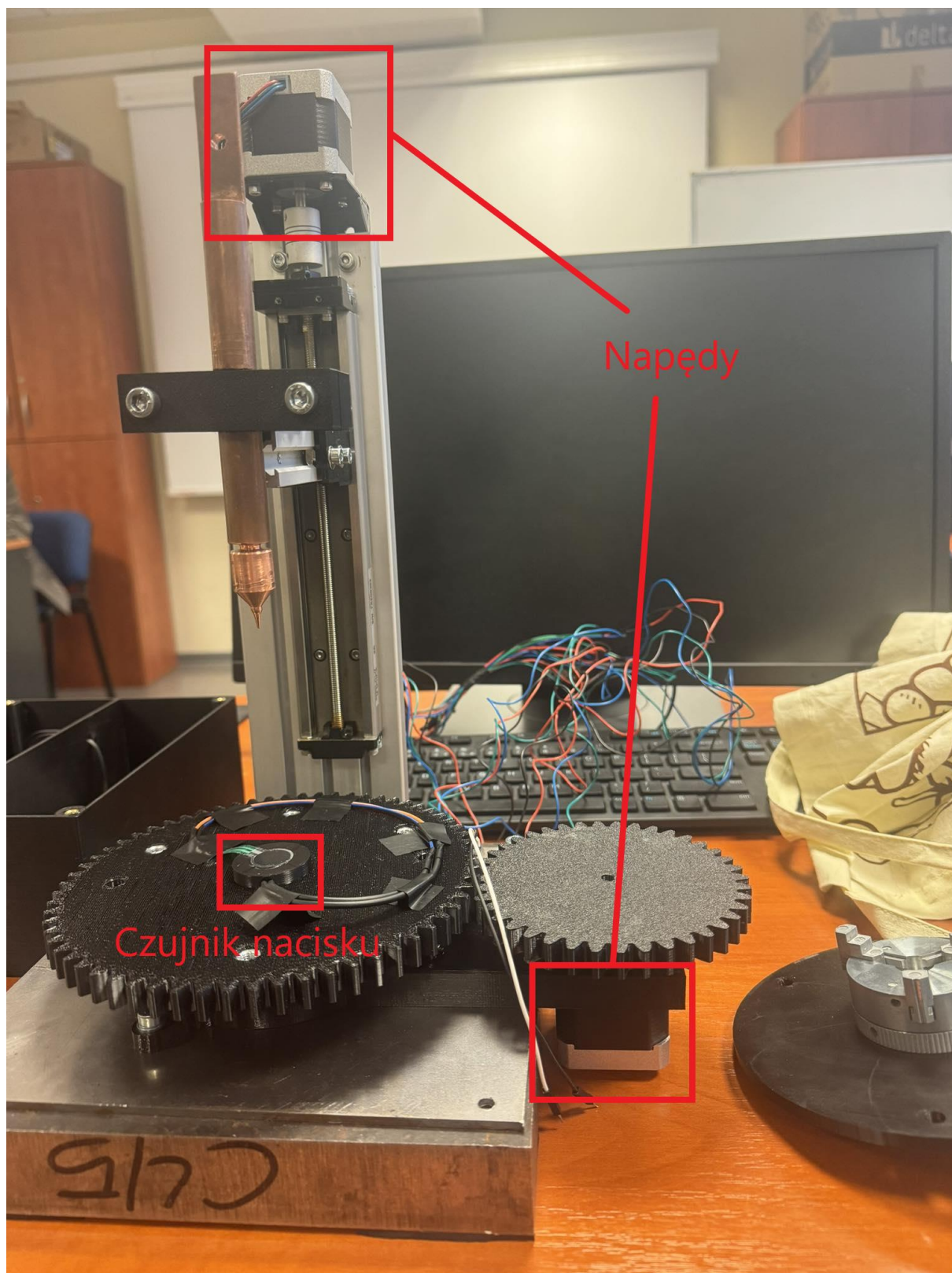
Rysunek 35 - Schemat ideowy układu elektronicznego



Na Rys. 36 i Rys. 37 przedstawiono położenie elementów na gotowej mechanice.



Rysunek 36 - Rozmieszczenie elementów wewnątrz obudowy



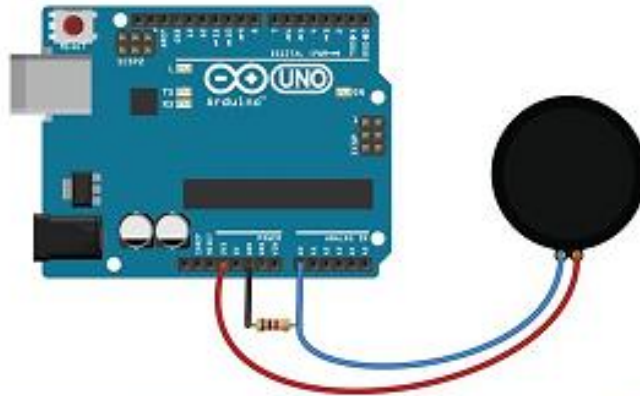
Rysunek 37 - elementy układu na ramie urządzenia



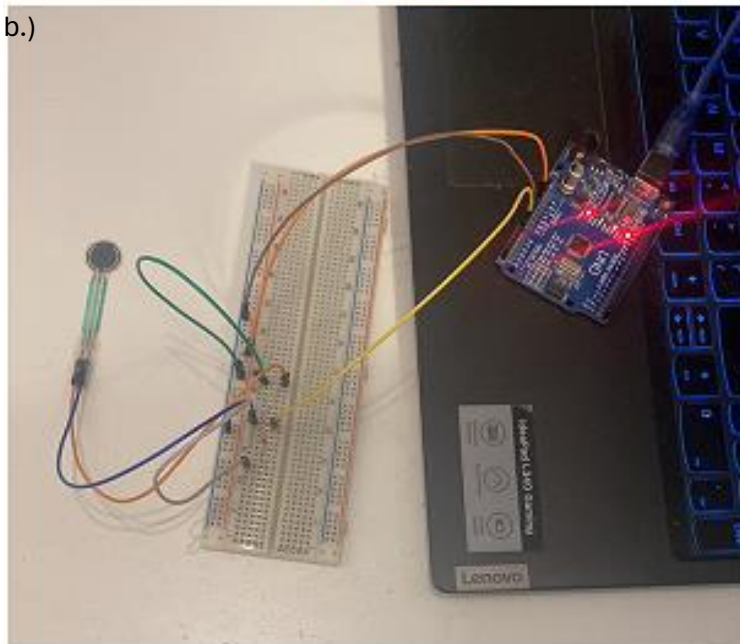
## b. Podłączenie czujnika nacisku

Podłączenie czujnika nacisku jest dość specyficznym zadaniem. Jest to czujnik analogowy, który opiera się na zmianie rezystancji membrany pod wpływem nacisku. Czujnik jest jednym z wyprowadzeń podpinany do zasilania, a drugim do wejścia analogowego. Dodatkowo konieczne jest poprowadzenie mostka między przewodem idącym na wejście analogowe, a masą przez rezystor. Zmiana wartości rezystancji tego rezystora wpływa na zakres pomiarowy, należy więc przeprowadzić kalibrację i dobrać ją odpowiednio. Poniżej na Rys. 38 przedstawiono podłączenie czujnika.

a.)



b.)



Rysunek 38 - a.) schemat podłączenia b.) rzeczywista realizacja podłączenia

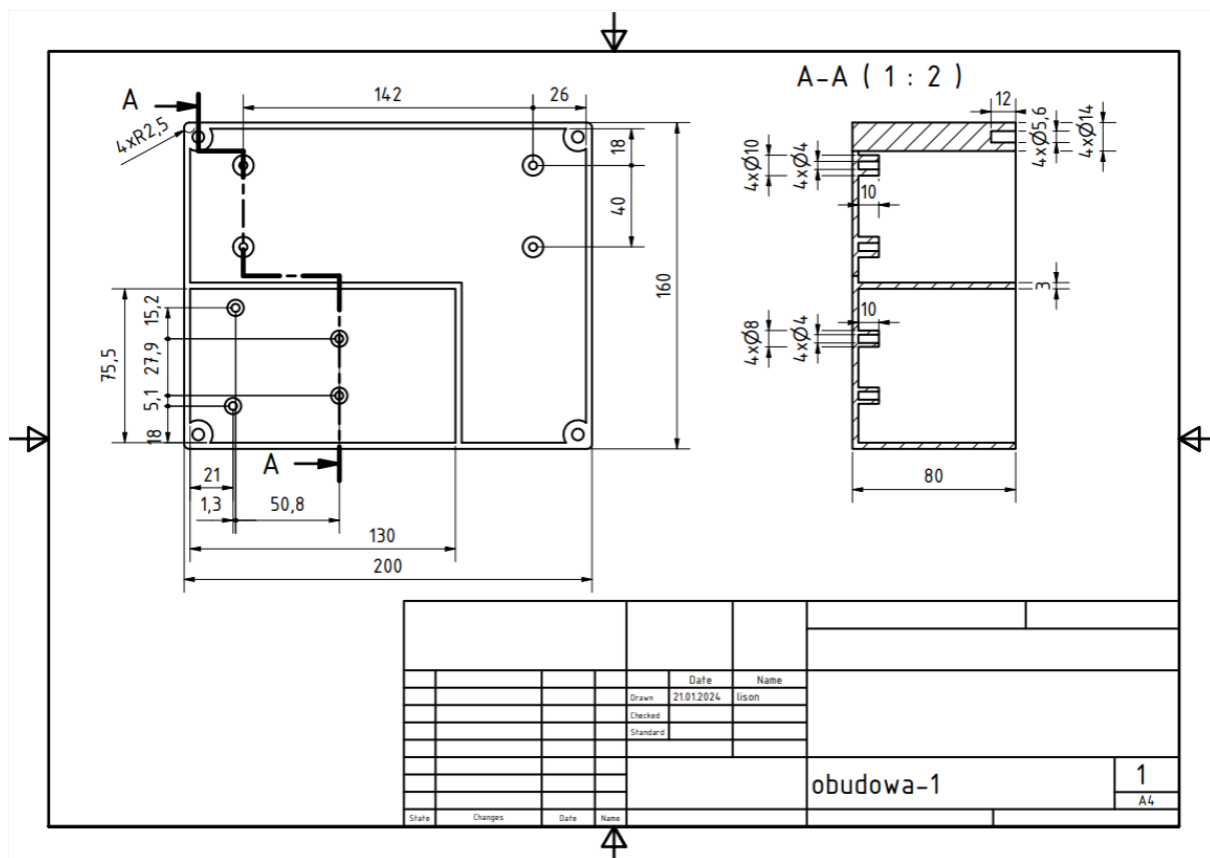


### c. Obudowa

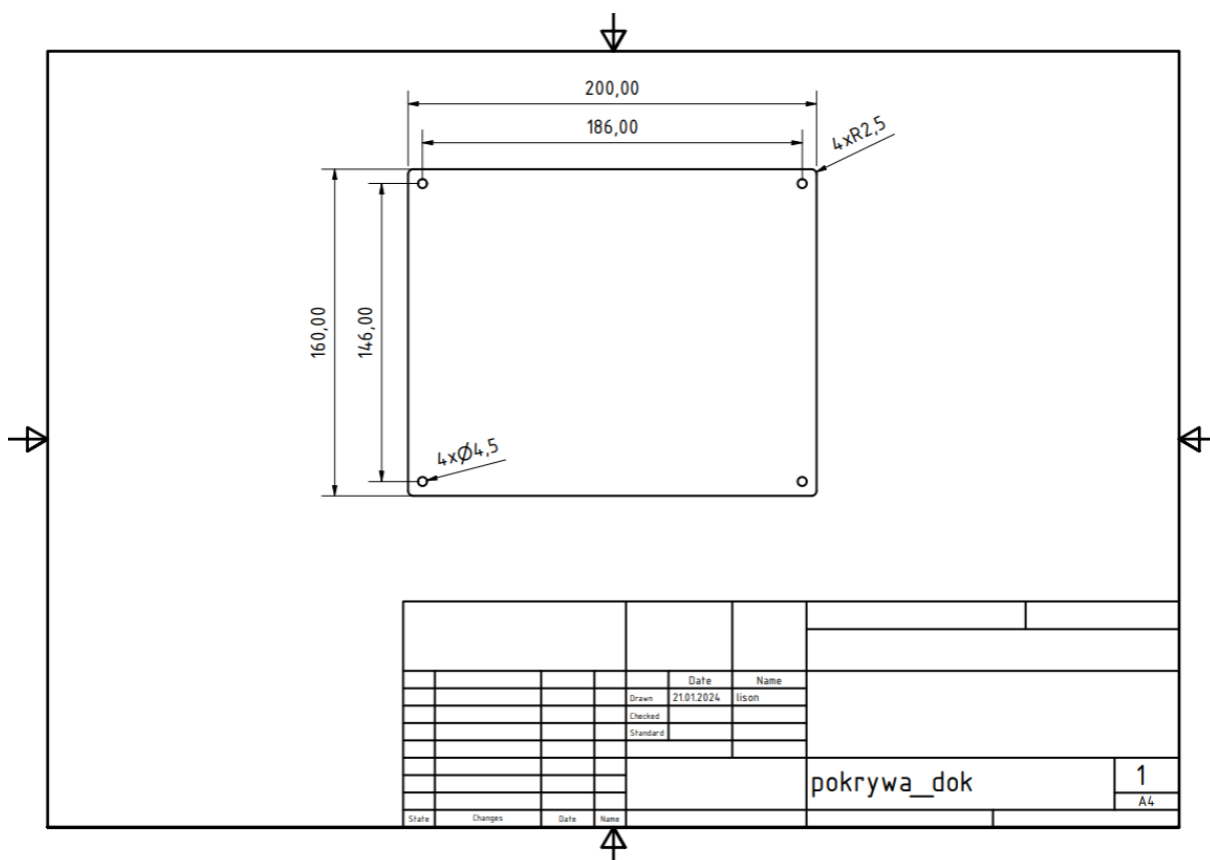
W ramach zabezpieczenia układu przed wpływem otoczenia oraz zapewnienia bezpieczeństwa użytkownikom zaprojektowano obudowę z tworzywa, wykonaną metodą druku 3D. Na Rys 39 przedstawiono finalny rezultat. Na Rys. 40 i 41 przedstawiono rysunki wykonawcze obudowy.



*Rysunek 39 – Obudowa*



Rysunek 40 - Rysunek wykonawczy komory



Rysunek 41 - Rysunek wykonawczy pokrywy

## 5. Sekcja programistyczna

### a. Konceptcje i zastosowane technologie

Początkowo założono 3 koncepcje realizacji sterowania procesem:

- zbudowanie fizycznego panelu obsługi z przyciskami, kontrolkami i małymi wyświetlaczami LCD.
- zakupienie dotykowego wyświetlacza LCD, który mógłby posłużyć jako panel sterowania.
- stworzenie aplikacji do obsługi procesu, którą można uruchomić na dowolnym komputerze.

Finalny wybór padł na koncepcję stworzenia aplikacji, co potwierdzają rozważania na temat następujących aspektów:

- aspekt finansów i dostępności – zamiast kupowania nowych komponentów wykorzystywane są komputery znajdujące się w laboratorium, bądź komputer prywatny osoby obsługującej urządzenie.
- aspekt możliwych modyfikacji – aplikację komputerową dużo łatwiej i szybciej można zmodyfikować, niż przebudowywać cały fizyczny panel sterowania.
- aspekt czasu – postawienie na rozwiązanie w pełni software'owe pozwala na zaoszczędzenie czasu związanym z fizycznym budowaniem panelu obsługi procesu.

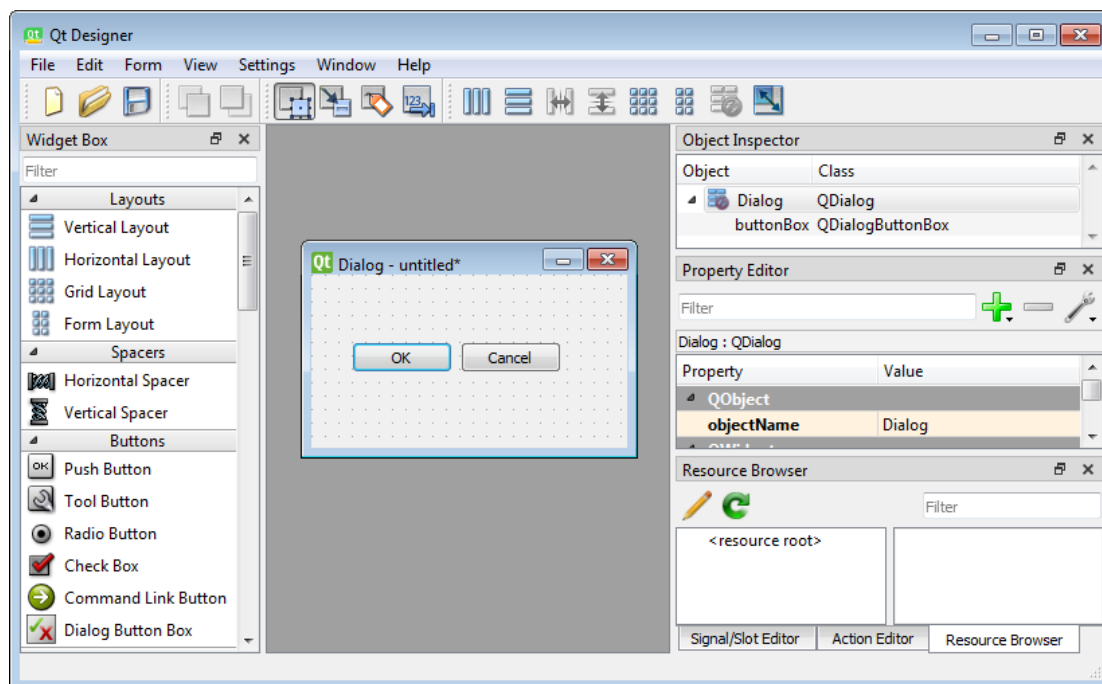
Do budowy aplikacji wykorzystano język programowania Python wraz z 3 bibliotekami rozszerzającymi funkcjonalności:

- -PyQt5 – biblioteka wykorzystana do stworzenia całego interfejsu graficznego aplikacji w formie okienek.
- pyserial – biblioteka wykorzystana do komunikacji z modułem Arduino (realizacja polecenia programu w zbudowanym urządzeniu).
- pyinstaller – biblioteka wykorzystana do konwersji kodu w języku Python do pliku wykonywalnego w formacie exe, który jesteśmy w stanie uruchomić na każdym komputerze z zainstalowanym systemem operacyjnym Windows.

### b. Interfejs użytkownika

Interfejs użytkownika wykonano w pełni z wykorzystaniem biblioteki PyQt5, rozszerzającej podstawowe funkcjonalności języka programowania Python. Użycie przytoczonej biblioteki pozwoliło na stworzenie przejrzystego interfejsu w formie okienek. Projektowanie okienek przeprowadzono w programie „Qt designer” (Rys. 42), gdzie zrealizowano całą warstwę graficzną aplikacji:

- modyfikacja kolorów i wielkości okienek oraz znajdujących się w nich napisów.
- modyfikacja kolorów, wielkości i oznaczeń przycisków.
- wstawienie pól do zmiany parametrów procesu.
- ograniczenia wyjścia z poszczególnych okienek.



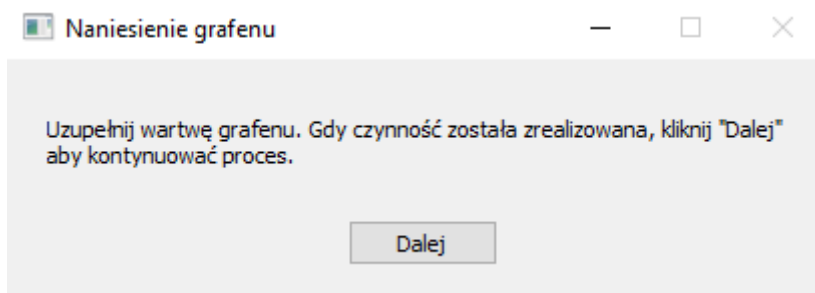
Rysunek 42 - interfejs programu "Qt designer" do tworzenia warstwy graficznej aplikacji

Głównym elementem interfejsu jest okno główne „MainWindow” (Rys. 43), gdzie znajdują się przyciski odpowiedzialne za wyłączenie programu i uruchomienia procesu anodowania. W oknie głównym znajdziemy również okienka umożliwiające wprowadzenie odpowiednich parametrów procesu, takich jak siła nacisku elektrody oraz ilość cykli obrotu.

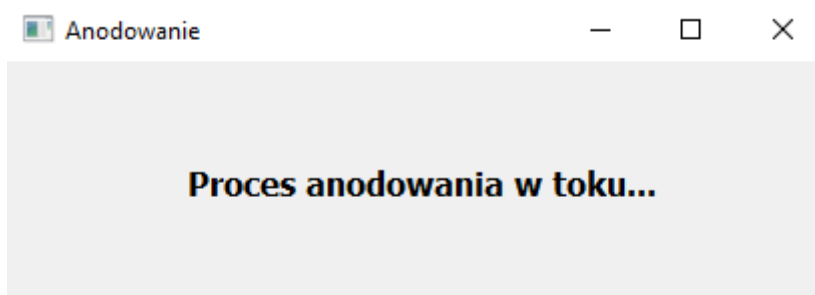


Rysunek 43 - Główne okno aplikacji "MainWindow".

Dodatkowo w trakcie obsługi aplikacji pojawiają się okienka informacyjne z przyciskami pozwalającymi na dalszą realizację programu (Rys. 44), bądź bez nich (Rys. 45)



Rysunek 44 - Przykładowe okno informacyjne z przyciskiem do kontynuowania realizacji programu



Rysunek 45 - Przykładowe okno informacyjne bez przycisku

## c. Obsługa programu

### i. Uruchomienie programu

Aby uruchomić program rozpakowujemy paczkę z folderem zawierającym program „IPZ\_proj.zip” na dysku komputera. W rozpakowanym folderze wyszukujemy pliku uruchomieniowego programu pod nazwą „init.exe” (Rys.46), którego uruchomienie pozwala nam na pracę z aplikacją.

| Nazwa                          | Data modyfikacji | Typ                 | Rozmiar   |
|--------------------------------|------------------|---------------------|-----------|
| .idea                          | 18.10.2023 01:44 | Folder plików       |           |
| build                          | 18.10.2023 01:25 | Folder plików       |           |
| dist                           | 18.10.2023 01:25 | Folder plików       |           |
| venv                           | 18.10.2023 01:21 | Folder plików       |           |
| <b>init</b>                    | 18.10.2023 01:25 | Aplikacja           | 35 242 KB |
| init                           | 18.10.2023 01:45 | JetBrains PyChar... | 5 KB      |
| init.spec                      | 18.10.2023 01:28 | Plik SPEC           | 1 KB      |
| logo PWr czarne poziom bez tła | 13.06.2016 08:28 | Plik PNG            | 27 KB     |
| mainwindow.ui                  | 17.10.2023 23:20 | Plik UI             | 11 KB     |
| window1.ui                     | 17.10.2023 20:07 | Plik UI             | 2 KB      |
| window2.ui                     | 17.10.2023 20:07 | Plik UI             | 2 KB      |
| window3.ui                     | 17.10.2023 20:08 | Plik UI             | 2 KB      |
| window4.ui                     | 17.10.2023 20:08 | Plik UI             | 2 KB      |
| window6.ui                     | 17.10.2023 20:08 | Plik UI             | 2 KB      |

Rysunek 46 - Widok folderu z podkreślonym plikiem uruchomieniowym aplikacji "init.exe"

### *ii. Wyłączenie programu*

Wyłączenie programu z poziomu aplikacji możliwe jest do zrealizowania tylko w momencie znajdowania się w oknie głównym aplikacji (Rys. 47) przyciskiem „Wyłącz program”, zlokalizowanym w prawym dolnym rogu głównego okna aplikacji.

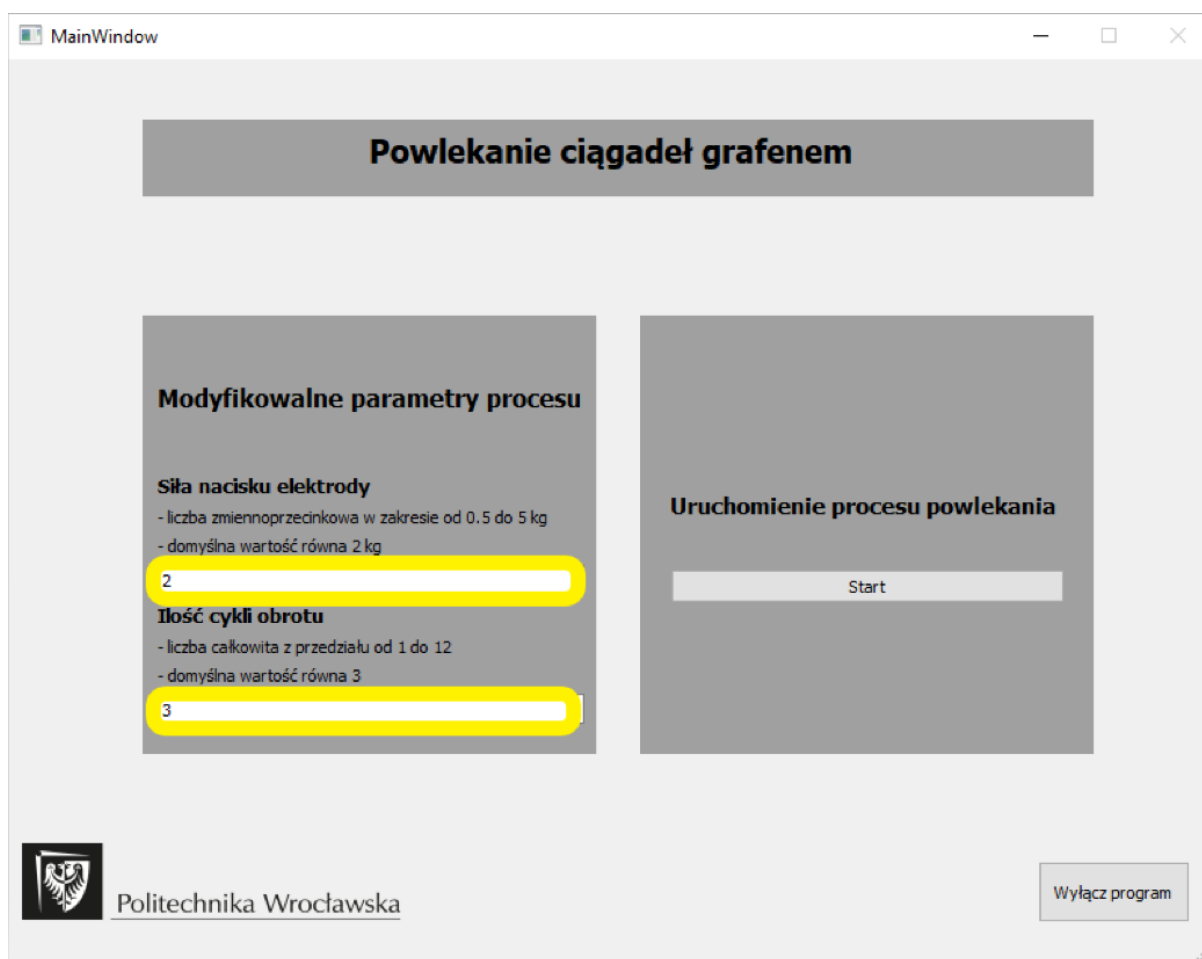


Rysunek 47 - Widok okna głównego z oznaczonym na czerwono przyciskiem „Wyłącz program” służącym do wyjścia z programu.

### *iii. Realizacja procesu*

Na samym początku realizacji procesu sprawdzamy określamy wartości modyfikowalnych parametrów za pomocą dwóch okienek umieszczonych po lewej stronie okna głównego (Rys. 48). Domyślnie owe parametry wynoszą odpowiednio:

- 2kg siły nacisku elektrody
- 3 cykle obrotu

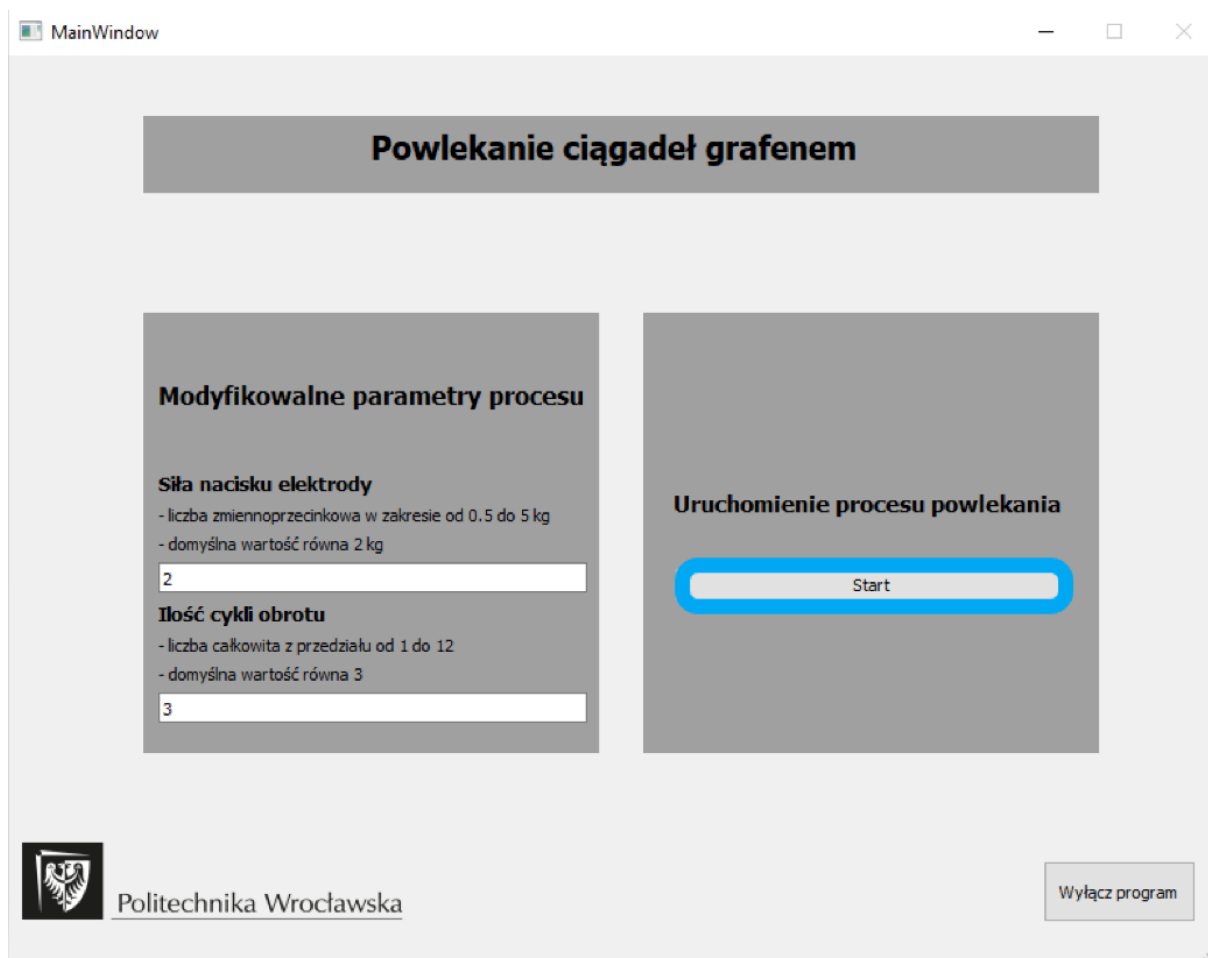


Rysunek 48 - Widok okna głównego z oznaczonymi na żółto okienkami służącymi do modyfikacji parametrów procesu.

Jeśli chcemy wykonać proces z wykorzystaniem innych wartości parametrów niż domyślne, wystarczy usunąć w okienku wartość domyślną i wpisać wartość własną (**należy pamiętać aby wartość własna była zgodna z opisem znajdującym się nad okienkiem konkretnego modyfikowalnego parametru!**).

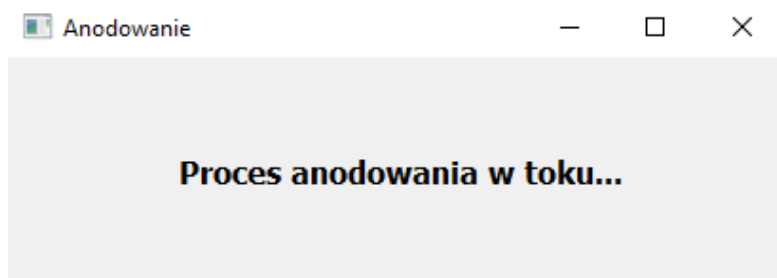
Przed uruchomieniem procesu powinniśmy upewnić się, że przedmiot do anodowania został poprawnie umieszczony w szczękach stołu maszyny. Uruchomienie procesu zaczynamy poprzez naciśnięcie przycisku „Start”, znajdującego się po prawej stronie okna głównego aplikacji (Rys. 49).





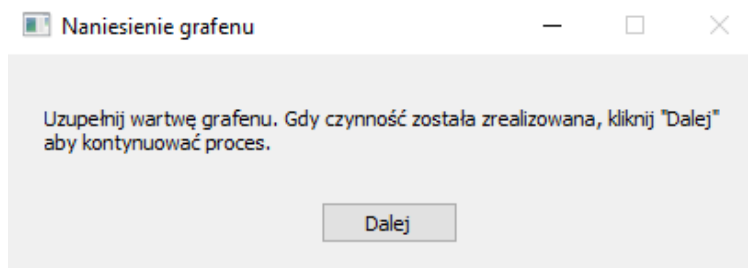
Rysunek 49 - Widok okna głównego z oznaczonym na niebiesko przyciskiem „Start”, służącym do uruchomienia procesu.

Obsługę procesu anodowania zapewniają wyświetlające się komunikaty. O trwaniu procesu użytkownika informuje poniższy komunikat (Rys. 50).



Rysunek 50 - Okno informacyjne o realizacji procesu anodowania.

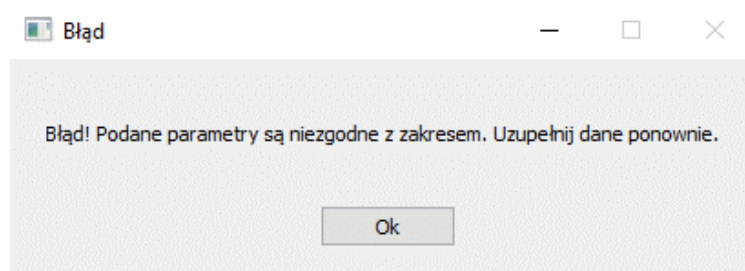
W trakcie realizacji procesu pojawiają się również komunikaty, które informują użytkownika o wymaganej fizycznej interakcji, takie jak uzupełnienie warstwy grafenu oraz uruchomienie elektrody (Rys. 51).



Rysunek 51 - Okno informacyjne o uzupełnieniu warstwy grafenu.

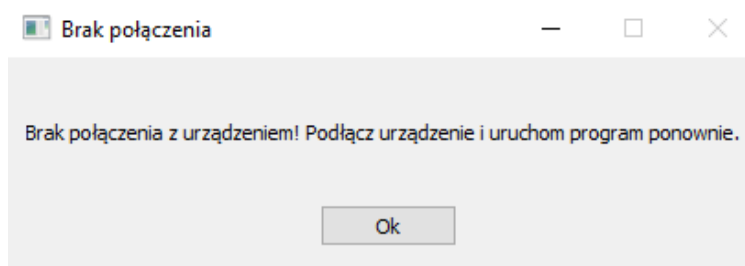
#### d. Zabezpieczenia oprogramowania

W przypadku wprowadzenia nieprawidłowych danych w okienkach siły nacisku elektrody oraz ilości cykli obrotu, następuje obsługa błędu w postaci resetu wartości w okienkach oraz wyświetlenia komunikatu poniżej (Rys. 52).



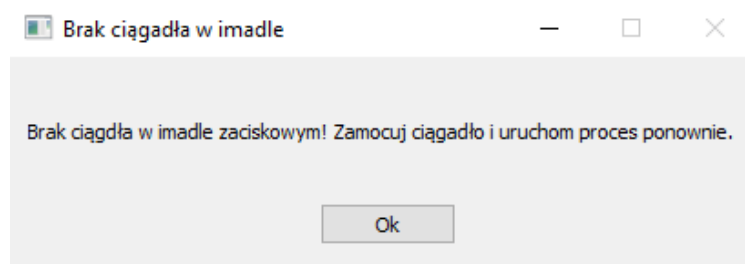
Rysunek 52 - Okno informacyjne z odniesieniem do wprowadzenia błędnych parametrów w programie.

W przypadku uruchomienia programu bez podłączenia urządzenia przez port USB do komputera, wyświetla się komunikat o braku połączenia urządzenia z komputerem (Rys. 53).



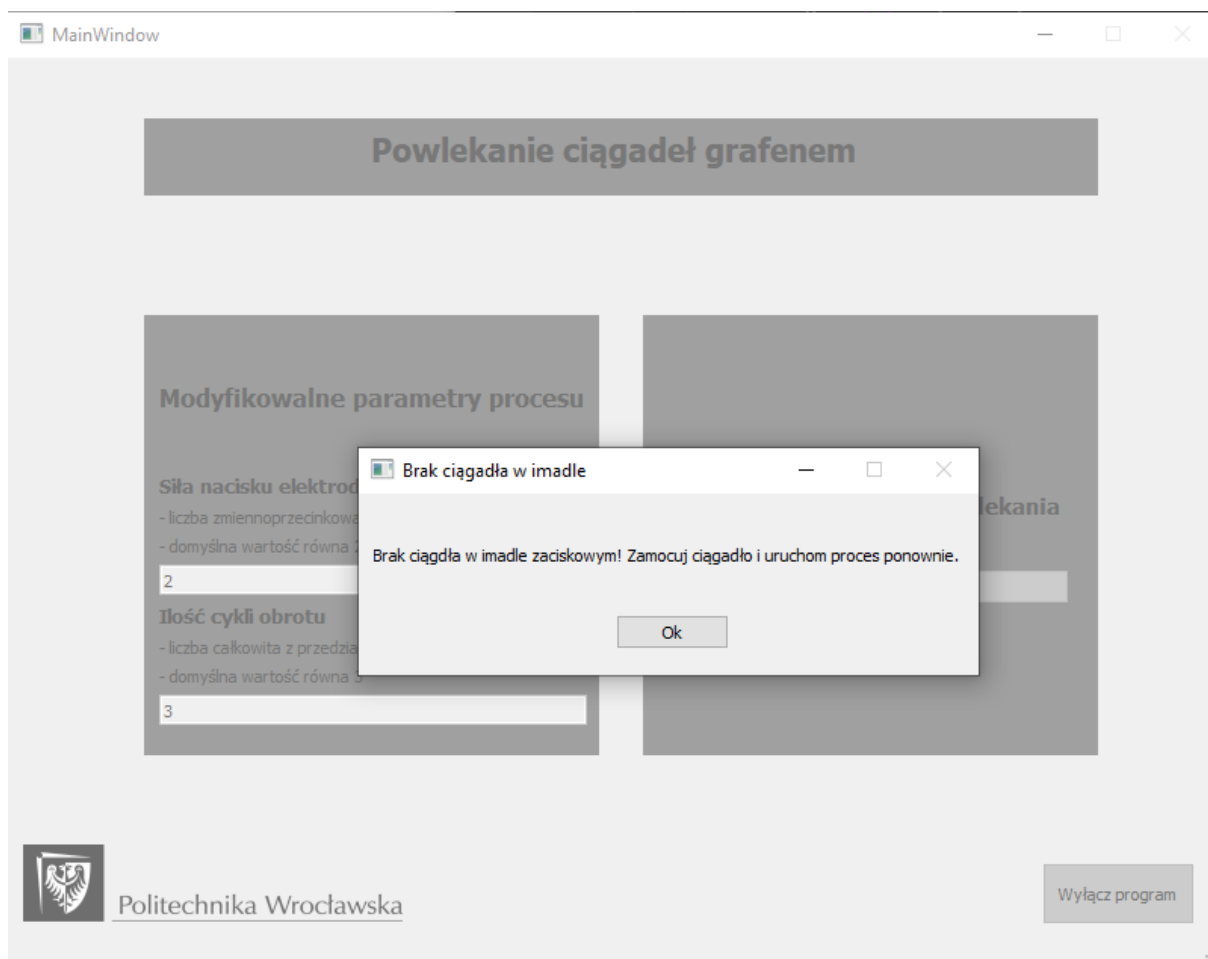
Rysunek 53 - Okno informacyjne z odniesieniem do braku połączenia urządzenia z komputerem.

W przypadku nie umieszczenia obiektu anodowanego na początku uruchomienia procesu anodowania następuje obsługa błędu w postaci zatrzymania procesu, wyświetlenia poniższego komunikatu oraz cofnięcia do okna głównego (Rys. 54).



Rysunek 54 - Okno informacyjne z odniesieniem do braku wykrycia obiektu do anodowania w imadle zaciskowym.

W przypadku uruchomienia procesu anodowania, okno „MainWindow” staje się zablokowane i dalsza komunikacja odbywa się wyłącznie poprzez wyświetlane komunikaty (Rys. 55).



Rysunek 55 - Widok okna głównego w trakcie zablokowania w trakcie realizacji procesu anodowania.

#### e. Kalibracja czujnika nacisku

Proces kalibracji rozpoczęto od organizacji odważników o konkretnych masach – 4 odważniki po 200g (Rys. 56) i 2 odważniki po 1250g (Rys. 57).



Rysunek 56 - Obiekt stanowiący odważnik 200g.



Rysunek 57 - Obiekt stanowiący odważnik 1250g.

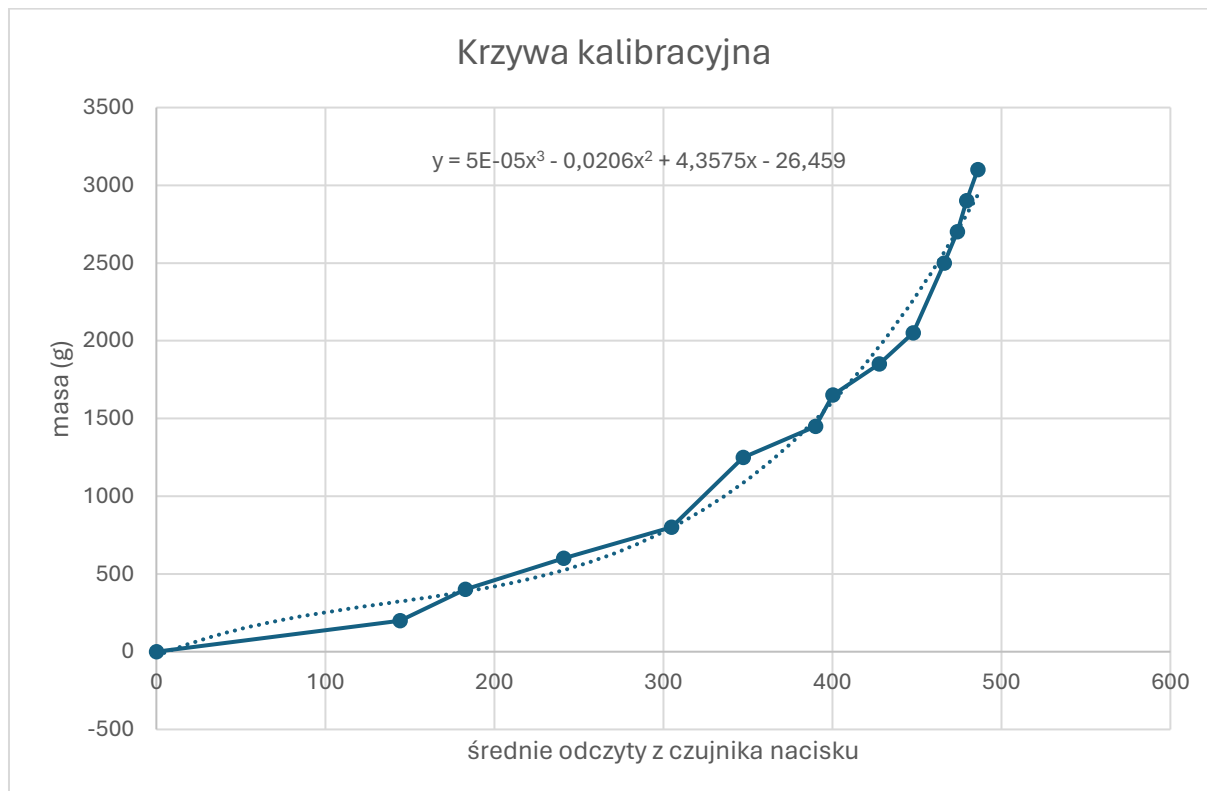
Dla znanych mas przeprowadzono 4 serie pomiarowe podczas których odczytywano wartości czujnika, odpowiadające odpowiedniej kombinacji mas kalibracyjnych. Po przeprowadzeniu pomiarów, z czterech serii pomiarowych wyznaczono wartości średnich arytmetycznych, które posłużyły następnie do wyznaczenia krzywej kalibracji. Wszystkie dane pomiarowe zebrane zostały w tabeli 1.

Tabela 1 Dane uzyskane w celu wyznaczenia krzywej kalibracji.

| masa(g) | pomiar 1 | pomiar 2 | pomiar 3 | pomiar 4 | średnia |
|---------|----------|----------|----------|----------|---------|
| 0       | 0        | 0        | 0        | 0        | 0       |
| 200     | 145      | 142      | 146      | 144      | 144,25  |
| 400     | 173      | 178      | 180      | 200      | 182,75  |
| 600     | 245      | 238      | 242      | 239      | 241     |
| 800     | 278      | 314      | 276      | 351      | 304,75  |
| 1250    | 367      | 342      | 339      | 341      | 347,25  |
| 1450    | 393      | 388      | 390      | 389      | 390     |
| 1650    | 400      | 398      | 402      | 401      | 400,25  |
| 1850    | 427      | 428      | 429      | 427      | 427,75  |
| 2050    | 446      | 442      | 449      | 454      | 447,75  |
| 2500    | 462      | 469      | 467      | 467      | 466,25  |
| 2700    | 475      | 473      | 473      | 475      | 474     |
| 2900    | 478      | 480      | 479      | 481      | 479,5   |
| 3100    | 489      | 485      | 486      | 484      | 486     |

Krzywa kalibracji została wyznaczona poprzez zdefiniowanie wartości osi X jako wartości odpowiadające kombinacjom mas kalibracyjnych (**żółta kolumna tabeli 1**) oraz wartości osi Y jako wartości średnich arytmetycznych odczytu czujnika (**zielona kolumna w tabeli 1**).

Do wyznaczonej krzywej dopasowano linię trendu, która jest najbardziej zbliżona kształtem do krzywej kalibracyjnej. Na wykresie wyświetlono również wzór funkcji dopasowanej linii trendu do danych. **Wzór funkcji dopasowanej linii trendu stanowi przelicznik do odczytu wartości rzeczywistych na podstawie odczytów czujnika w programie Arduino!** Postać krzywej kalibracyjnej oraz dobranej do niej linii trendu przedstawiono na rysunku poniżej (Rys. 58).



Rysunek 58 - Krzywa kalibracyjna z dobraną linią trendu i wzorem funkcji linii trendu.

Ostatecznie w wyniku przeprowadzonej kalibracji udaje odczytać się wartości masy rzeczywistej z błędem na poziomie od 5% do 15%.

Aby zmodyfikować odczyty na podstawie własnej kalibracji należy podmienić postać funkcji przelicznika w następujących miejscach programu, realizowanego na Arduino (**Podkreślone na rysunkach czerwoną linią**):

- warunek pojedynczego odczytu masy po kalkulacji (Rys. 59)
- warunek poruszania w dół z limitem czujnika nacisku (Rys. 60)
- warunek precyzyjnego poruszania w dół w celu uzyskania najdokładniejszego możliwego odczytu (Rys. 61)

```

57 if (firstChar == 's') { // Reading once value of calculated mass
58   val = analogRead(A5);
59   calibrationWeigth = 0.00005 * pow(val, 3) - 0.0206 * pow(val, 2) + 4.3575 * val - 26.459; //Change if own calibration function is needed
60   delay(100);
61   Serial.println(calibrationWeigth);
62 }

```

Rysunek 59 - Warunek pojedynczego odczytu masy po kalkulacji.

```

94  if (firstChar == 'd') { // Moving downwards (limited by pressure sensor)
95      val = analogRead(A5);
96      zero = 0.00005 * pow(val, 3) - 0.0206 * pow(val, 2) + 4.3575 * val - 26.459; // Change if own calibration function is needed
97      digitalWrite(DirZ, LOW);
98
99      while (true) {
100          for (int i = 0; i < 200; i++) { // Amount of steps on full revolut of motor Z axis
101              digitalWrite(StepZ, HIGH);
102              delayMicroseconds(1000); // Delay between steps
103              digitalWrite(StepZ, LOW);
104              delayMicroseconds(1000); // Delay between steps
105          }
106          val = analogRead(A5);
107          calibrationWeight = 0.00005 * pow(val, 3) - 0.0206 * pow(val, 2) + 4.3575 * val - 26.459; // Change if own calibration function is needed
108          reqWeight = calibrationWeight - zero;
109          if (reqWeight > 400) { // If read actual value is high enough - stop the motor Z axis
110              stepperZ.stop();
111              break;
112          }
113      }
114      Serial.println("Koniec");
115  }

```

*Rysunek 60 - Warunek poruszania w dół z limitem czujnika nacisku.*

```

117  if (firstChar == 'm') { // Precise downward movement to get the most accurate mass reading from sensor
118      KGforce = command.substring(1).toFloat();
119      Gforce = KGforce * 1000;
120
121      if (Gforce >= 500) { // Realise the function only when the value of Gforce was specified
122          digitalWrite(DirZ, HIGH);
123          stepperZ.move(500);
124          stepperZ.runToPosition();
125
126          val = analogRead(A5);
127          calibrationWeight = 0.00005 * pow(val, 3) - 0.0206 * pow(val, 2) + 4.3575 * val - 26.459; // Change if own calibration function is needed
128          reqWeight = calibrationWeight - zero;
129          delay(70);
130          digitalWrite(DirZ, LOW);
131          while (reqWeight < 0.95*Gforce) { // Loop until getting specified Gforce value
132              if (reqWeight > 0.7*Gforce) { // Move slower when weight value is getting closer to derired Gforce value
133                  stepperZ.move(-5);
134              }
135              else {
136                  stepperZ.move(-100);
137              }
138              stepperZ.runToPosition();
139              sum = 0;
140              reqWeight = 0;
141              for (int i = 0; i < 10; i++) { // Make 10 mearurements, than calculate the average value of read weight
142                  val = analogRead(A5);
143                  delay(40);
144                  calibrationWeight = 0.00005 * pow(val, 3) - 0.0206 * pow(val, 2) + 4.3575 * val - 26.459; // Change if own calibration function is needed
145                  reqWeight = calibrationWeight - zero;
146                  sum = sum + reqWeight;
147              }
148              reqWeight = sum/10;
149          }
150      }
151      Serial.println("Koniec");
152  }

```

*Rysunek 61 - Warunek precyzyjnego poruszania w dół w celu uzyskania najdokładniejszego możliwego odczytu.*

## 6. Podsumowanie

W ramach projektu skonstruowano urządzenie które pozwala na realizację zaproponowanej technologii nakładania powłok grafenowych. Udało się zrealizować wszystkie przyjęte założenia projektowe. Należy natomiast zaznaczyć że ze względu na ograniczenia budżetowe urządzenie jest tylko i wyłącznie akademickim dowodem na wykonalność urządzenia realizującego zaproponowany proces i nie nadaje się do aplikacji przemysłowych. W celu uzyskania urządzenia nadającego się do bardziej wymagających aplikacji konieczne byłyby modyfikacje i usprawnienia podnoszące wytrzymałość, bezpieczeństwo, powtarzalność i dokładność.

Przykładowe usprawnienia mogą dotyczyć podniesienia dokładności pomiaru nacisku. Możliwe realizacje można podzielić na 3 kategorie:

- Zastosowanie dokładniejszych czujników
- Zastosowanie większej ilości czujników i uśrednienia odczytów
- Modyfikacja mechanizmu naciskającego na czujnik tak aby pozwalał na zachowanie pełnej liniowości

Wszystkie z powyższych możliwości zostały odrzucone ze względów finansowych.

Z tych samych powodów w konstrukcji wykorzystano dużą ilość elementów drukowanych w technologii druku 3D. W celu zwiększenia wytrzymałości wykorzystanie elementów z bardziej trwałych materiałów np. stali pozwoliłoby zwiększyć wytrzymałość.

Jak widać, zaprojektowane urządzenie nie jest perfekcyjnym rozwiązaniem, natomiast dowodzi że urządzenie takie jest wykonalne, nawet przy poważnych ograniczeniach budżetowych