

<b>1. WSTĘP .....</b>	4
<b>1.1. Geneza.....</b>	4
<b>1.2. Cel i zakres pracy .....</b>	5
<b>2. PODSTAWY TEORETYCZNE .....</b>	6
<b>2.1. Serwonapędy .....</b>	6
2.1.1. Wprowadzenie .....	6
2.1.2. Zasada działania.....	6
2.1.3. Sterowanie serwonapędami .....	7
<b>2.2. Serwomechanizm.....</b>	8
2.2.1. Wprowadzenie .....	8
2.2.2. Sterowanie serwomechanizmu .....	9
2.2.3. Zasilanie serwomechanizmu.....	9
<b>3. BUDOWA STEROWNIKA.....</b>	10
<b>3.1. Wstęp .....</b>	10
<b>3.2. Wykorzystany sterownik oraz rozszerzenia.....</b>	10
3.2.1. Sterownik X20 CP 3586 oraz jego rozszerzenia .....	10
3.2.2. Bloki wejść/wyjść .....	11
3.2.3. Zasilacz 80PS080X3.10-01 .....	12
3.2.4. ACOPOSmicro 100D Servo .....	13
<b>4. PROTOTYP.....</b>	14
<b>4.1. Budowa .....</b>	14
<b>5. MAKIETA .....</b>	15
<b>5.1. Budowa .....</b>	15
5.1.1. Wstęp .....	15
5.1.2. Zasada działania.....	15
5.1.3. Mechanizm rewolwerowy.....	19
5.1.4. Mechanizm napełniania .....	23
5.1.5. Mechanizm zakręcania .....	23
5.1.6. Mechanizm podawania butelek .....	26
5.1.7. Mechanizm podawania nakrętek .....	29
5.1.8. Panel sterowniczy .....	32
5.1.9. Tor butelek .....	33
5.1.10. Połączenia elektryczne.....	35
<b>5.2. Połączenie Ethernet z PLC .....</b>	36
<b>5.3. Zmienne i konfiguracja portów.....</b>	41
<b>5.4. Kod programu.....</b>	44
<b>5.5. Zabezpieczenia .....</b>	48
<b>6. PODSUMOWANIE.....</b>	49
<b>6.1. Wnioski.....</b>	49
<b>7. BIBLIOGRAFIA.....</b>	51

# 1. WSTĘP

## 1.1. Geneza

Ze względu na szybki rozwój technologii w dzisiejszym świecie coraz więcej procesów zostaje zastąpionych przez fabryki, w których w wyniku zminimalizowanej pracy człowieka powstają gotowe produkty. Między XVII a XX wiekiem nastąpił największy skok technologiczny, w którym bardzo duża część procesów wymagających człowieka została zamieniona na pracę sterownika operującego danymi czynnościami. Początkowo były to proste układy pozwalające na automatyczną regulację, później wymyślono pracę układu ze sprzężeniem zwrotnym tworząc podwaliny do automatyki przemysłowej. Pod koniec XX wieku zaczęły powstawać komputery oraz pierwsze maszyny sterujące wieloma zadaniami jednocześnie, umożliwiając powstawanie dzisiaj znanych fabryk z taśmami produkcyjnymi [1].

Pomysł pracy inżynierskiej o tytule „System automatycznego napełniania butelek wykorzystujący sterownik X20 firmy B&R” powstał ze względu na chęć stworzenia pracy wykorzystującej wszystkie nauczone na uczelni umiejętności. Sterowanie za pomocą PLC, zautomatyzuje proces napełniania butelek w fabrykach. Nie jest to proces łatwy, gdyż istnieje wiele czynników, które mogą negatywnie wpływać na pracę całej linii produkcyjnej. Problemy, z którymi się spotkałem oraz musiałem rozwiązać zostały głównie opisane w rozdziale 4. Każda osoba tworząca takie rozwiązanie musi zmierzyć się z:

- trudnościami z napełnianiem – Różne maszyny do napełniania oferowane przez producentów automatycznych maszyn do napełniania butelek stosują różne zasady napełniania i typy dysz. Trzeba także pamiętać, że każda z cieczy posiada swoje parametry, które drastycznie wpływają na prędkość napełniania. Najczęstszym jednak problemem w takich rozwiązanych jest niewłaściwy sposób napełniania lub zużywanie się części maszyny, które wymagają wymiany.

- kłopotami z nakręcaniem – W makiecie była to najtrudniejsza część, gdyż nie dość, że składała się z wielu elementów ruchomych, to ich zgranie oraz odpowiednia praca zajęła najczęściej czasu w projekcie. Okazuje się, że w przypadku maszyn umieszczonych na liniach produkcyjnych jest to niemały problem, ponieważ w przypadku nieodpowiedniego zakręcenia butelki może okazać się, że ponowne zakręcenie butelki będzie niemożliwe ze względu na zniszczenie gwintu przez nieodpowiednie dobranie sił zakręcających, zakrętka może być za luźna lub zakręcona za mocno niszcząc korek. W przypadku producentów automatycznych maszyn do napełniania butelek użycie sprzętu do zamykania będzie się różnić w zależności od rodzaju nakrętki i uszczelki.

- wywracanie butelki/ spadanie z wyznaczonej trasy – Kolejnym znaczącym problemem jest takie dobranie zapór, ścianek oraz wielu innych części, aby butelka nie wypadała z trasy, bądź nie była wywracana przez samą maszynę. Znaczące jest wtedy odpowiednie pobieranie, a także pozycjonowanie butelki w zależności od czynności jaka ma zostać wykonana. Na przykład nieprawidłowo umiejscowiona butelka w mechanizmie napełniającym doprowadzić może do rozlania cieczy tracąc ją, dodatkowo istnieje możliwość, że zaleje części elektroniczne, co doprowadzić może do katastrofalnych skutków. Dlatego tak ważnym jest, aby jeśli w którejkolwiek maszynie lub miejscu można zidentyfikować przewrócenie się butelki lub rozlanie, należy sprawdzić maszynę i w razie potrzeby dokonać regulacji.

- problemy z komponentami maszyny i ustawieniami PLC – W większości przypadków istnieje wiele prostych oraz uniwersalnych rozwiązań radzenia sobie z błędami związanymi z maszyną i PLC, natomiast w rzadkich przypadkach, ze względu na budowanie maszyny na potrzeby konkretnego projektu, każda z maszyn jest lekko inna,

doprowadzając do potrzeby zatrudnienia, bądź przeszkolenia całego personelu tak, aby był w stanie radzić sobie z występującymi problemami [2].

Zauważamy zatem, że tworzenie linii produkcyjnych to skomplikowany i długий proces i wymaga wielu godzin poświęconych przemyśleniu wszystkich części, stworzeniu, testowaniu, a także wielu przygotowaniom i szkoleniom do odpowiedniej pracy całej fabryki.

Wszystkie powyższe problemy skłoniły mnie do wyboru takiej pracy, umożliwiając poszerzenie informacji w zakresie:

- wycinania elementów na obrabiarkę CNC,
- rozwiązywania problemów opisanych powyżej,
- stworzenie podstawowego obwodu elektronicznego,
- sterowanie ruchem napędów,
- programowanie sterownika PLC firmy B&R.

Niestety w dostępnych źródłach internetowych nie ma wiedzy w postaci artykułów bądź innego rodzaju technicznych danych, gdyż każda z informacji jest chroniona przez firmy. Musiałem zatem w dużym stopniu odnosić się do pojedynczych stron internetowych, w których były jakiekolwiek dostępne oraz potrzebne informacje.

## **1.2. Cel i zakres pracy**

Diplomowa praca inżynierska ma na celu opracowanie koncepcji i wykonanie modelu laboratoryjnego stanowiska do automatycznego zakręcania butelek, które będzie sterowane z wykorzystaniem sterownika PLC X20 i serwonapędu ACOPOS firmy B&R.

W zakres pracy wchodzi:

- zapoznanie się z budową i programowaniem sterowników X20 i serwo-napędów ACOPOS,
- opracowanie koncepcji modelu laboratoryjnego układu do automatycznego zakręcania nakrętek,
- opracowanie modelu elektrycznego,
- opracowanie programu sterującego i testy całego systemu,
- redakcja pracy.

## **2. PODSTAWY TEORETYCZNE**

### **2.1. Serwonapędy**

#### **2.1.1. Wprowadzenie**

W dynamicznie rozwijającym się świecie przemysłu, automatyzacji i technologii, serwonapędy odgrywają kluczową rolę w zapewnianiu precyzyji, stabilności oraz efektywności ruchu w różnorodnych aplikacjach. Serwonapędy stanowią integralną część systemów mechanicznych, umożliwiając niezwykle precyzyjne sterowanie ruchem maszyn, urządzeń i robotów. Pozwalają na precyzyjne przemieszczanie między różnymi położeniami w przypadku bardzo małych prędkości bądź tam, gdzie występuje duża dynamika ruchu.

Podstawową zasadą działania serwonapędów jest wykorzystanie sprzężenia zwrotnego, gdzie sygnały pomiarowe z czujników pozycji, prędkości lub momentu obrotowego są porównywane z sygnałem referencyjnym. Kontroler serwonapędu analizuje różnicę między tymi sygnałami i odpowiednio steruje silnikiem lub aktuatorem, aby zminimalizować tę różnicę. Dzięki temu serwonapędy są w stanie precyzyjnie utrzymywać określoną pozycję, osiągać żądaną prędkość oraz kontrolować moment obrotowy.

W tworzonym projekcie znalazły one idealnie zastosowanie w precyzyjnym sterowaniu mechanizmem zakręcania oraz rewolwerowym przesuwaniu butelek, gdyż każda z tych czynności musi zostać wykonana w jak najbardziej precyzyjnym stopniu [4].

#### **2.1.2. Zasada działania**

Zasada działania serwonapędów opiera się na zastosowaniu sprzężenia zwrotnego, co pozwala na precyzyjne kontrolowanie ruchu różnych urządzeń, maszyn czy robotów. Poniżej opisana została główna pętla działania napędu.

##### **a) Czujniki**

Serwonapędy są wyposażone w enkodery, potencjometry, rezolwery lub czujniki hallotronowe. Te czujniki pomiarowe monitorują rzeczywistą pozycję, prędkość lub moment obrotowy urządzenia. W zależności od zastosowania czujnika należy także wziąć pod uwagę ich rozdzielczości, a także wiele innych parametrów jakie posiadają.

##### **b) Sygnał referencyjny**

Sygnał referencyjny jest pożądaną wartością, którą chcemy osiągnąć. Może to być określona pozycja, prędkość czy moment obrotowy, który ma być utrzymany przez serwonapęd.

##### **c) Porównywanie sygnałów**

Sygnał zwrotny z czujników (mierzony pozycji, prędkości lub momencie obrotowym) jest porównywany z sygnałem referencyjnym. Różnica między sygnałem zwrotnym, a sygnałem referencyjnym jest używana do dostosowywania prędkości oraz korygowania pozycji.

##### **d) Kontroler**

Kontroler serwonapędu, często oparty na mikrokontrolerze lub dedykowanym układzie scalonym, analizuje ten błąd i generuje sygnał sterujący na podstawie tego porównania. Kontroler stosuje algorytmy sterowania, takie jak PID (regulator pro-

porcjonalno-integrująco-różniczkujący), do zoptymalizowania odpowiedzi serwonapędu.

e) Silnik lub aktuator

Na podstawie sygnału sterującego od kontrolera, silnik elektryczny, aktuator hydrauliczny, pneumatyczny lub inny rodzaj napędu jest odpowiednio sterowany. Silnik wykonuje ruch lub generuje moment obrotowy, który przesuwa urządzenie w kierunku żądanego położenia lub utrzymuje żądaną prędkość.

f) Sprzężenie zwrotne

Po wykonaniu ruchu, serwonapęd ponownie odczytuje sygnał zwrotny z czujników i porównuje go z sygnałem referencyjnym. Ten proces sprzężenia zwrotnego pozwala na ciągłe dostosowywanie ruchu urządzenia w celu utrzymania pożądanej pozycji, prędkości lub momentu obrotowego.

Dzięki tej złożonej pętli sprzężenia zwrotnego, serwonapędy są w stanie osiągać bardzo wysoką precyzję i stabilność w utrzymaniu położenia, prędkości lub momentu obrotowego. Są niezwykle przydatne w aplikacjach wymagających dokładnego sterowania, takich jak roboty przemysłowe, obrabiarki CNC, systemy pozycjonowania, a także w pojazdach autonomicznych i liniach produkcyjnych.

### 2.1.3. Sterowanie serwonapędami

Zasada sterowania serwonapędami obejmuje wykorzystanie odpowiednich algorytmów i technik w celu precyzyjnego kontrolowania ich ruchu. W przypadku serwonapędów istnieją różne techniki sterowania.

a) Sterowanie Pętlą Otwartą (ang. Open-Loop Control)

Sterowanie pętlą otwartą polega na bezpośrednim przekazywaniu sygnałów sterujących do serwonapędu bez uwzględniania sprzężenia zwrotnego z czujników. Jest to prosta metoda, ale może być niedokładna, ponieważ nie uwzględnia zmian w środowisku lub w obciążeniach, które mogą wpływać na ruch urządzenia.

b) Sterowanie Pętlą Zamkniętą (ang. Closed-Loop Control)

Sterowanie pętlą zamkniętą jest znacznie bardziej zaawansowane i precyzyjne. W tej metodzie serwonapęd jest wyposażony w czujniki (takie jak enkodery) do pomiaru rzeczywistej pozycji, prędkości lub momentu obrotowego. Sygnał zwrotny z tych czujników jest porównywany z sygnałem referencyjnym (pożdanym stanem) i na podstawie tej różnicy generowany jest sygnał sterujący. Popularnymi technikami sterowania pętlą zamkniętą są:

c) Sterowanie PID

Algorytm PID analizuje błąd między sygnałem zwrotnym a sygnałem referencyjnym i generuje sygnał sterujący, który składa się z trzech składników: proporcjonalnego, który jest proporcjonalny do błędu, całkowitego, który uwzględnia sumę błędów w czasie oraz różniczkującego, który stara się przewidzieć przyszłe zmiany błędu.

d) Sterowanie LQR (ang. Linear Quadratic Regulator)

LQR jest techniką sterowania opartą na równaniach matematycznych, które uwzględniają dynamikę systemu, celując w minimalizację funkcji kosztu. Jest stosowany w przypadku bardziej skomplikowanych systemów dynamicznych.

e) Sterowanie MPC (ang. Model Predictive Control)

MPC to technika sterowania, która wykorzystuje matematyczne modele systemu do przewidywania jego zachowania w przyszłości. Na podstawie tych przewidywań generowany jest sygnał sterujący, który minimalizuje różnicę między rzeczywistym zachowaniem systemu a pożądanym zachowaniem.

f) Sterowanie adaptacyjne

Sterowanie adaptacyjne wykorzystuje techniki sztucznej inteligencji, takie jak algorytmy uczenia maszynowego, aby automatycznie dostosować się do zmieniających się warunków środowiska lub obciążień. System może dostosowywać swoje parametry w czasie rzeczywistym na podstawie danych zebranych z czujników.

g) Sterowanie za pomocą logiki rozmytej (Fuzzy Logic):

Sterowanie oparte na logice rozmytej pozwala na uwzględnienie nieliniowych i rozmytych aspektów systemu, co jest przydatne w przypadku systemów, których zachowanie nie jest łatwo opisane za pomocą tradycyjnych równań matematycznych.

W projekcie wykorzystano specjalny moduł sterujący silnikami ACOPOS micro 100D Servo (80VD100PD.C00-01), który w bardzo prosty sposób pozwalał na regulację pracy serwonapędów za pomocą sterownika PLC. Stosuje on rozbudowaną metodą MPC, która dzięki pętli sprzężenia zwrotnego otrzymuje dane z precyzyjnego enkodera zamontowanego na silniku umożliwiając jego precyzyjną regulację, która w przypadku nakręcania nakrętek oraz obrotu do określonej pozycji odgrywa kluczową rolę [3].

## 2.2. Serwomechanizm

### 2.2.1. Wprowadzenie

Serwomechanizm jest elementem zamkniętym w małej obudowie składającym się z:

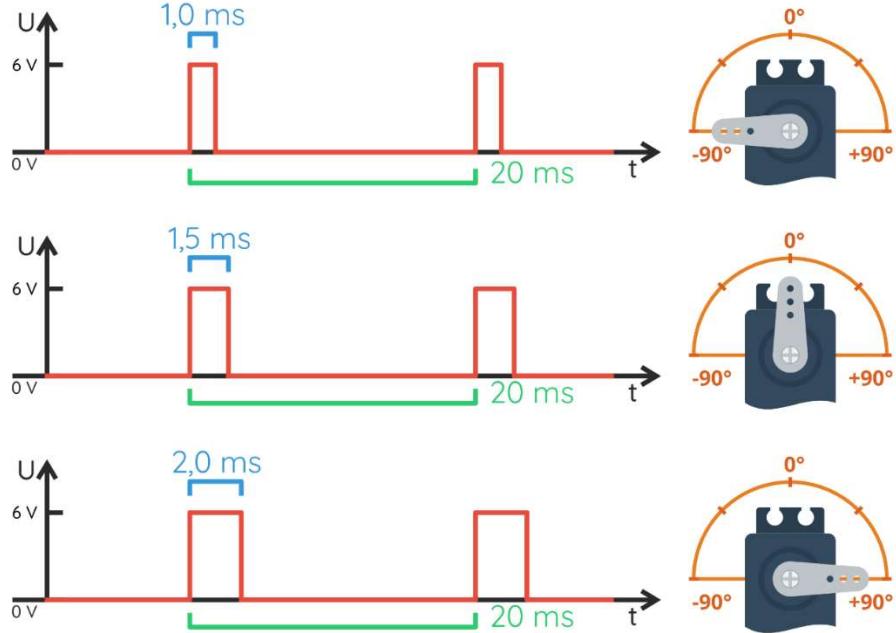
- przekładni,
- silnika DC,
- układu elektronicznego,
- potencjometru.

Dzięki tak skomplikowanej budowie umożliwia ruch po połowie okręgu w zakresie  $0^\circ$  do  $180^\circ$ . Stworzone zostały także serwa pełnoobrotowe, jednak w naszym przypadku będziemy stosować tylko te pierwsze. Są to względnie małe elementy posiadające duże wartości momentu w zależności od napięcia, które są kluczowe, gdyż będą one musiały podnieść mechanizm wraz z serwonapędem.

Głównym elementem oddzielającym serwomechanizmy od siebie, a także najczęściej wadliwym są zębatki, ponieważ rodzaj materiału z jakiego zostały wykonane, a także ich przekładnia, będą miały ogromne znaczenie w ich efektywności i niezawodności. Dlatego wielu producentów podaje jakiego one są rodzaju: przekładnie nylowe, karbonitowe czy metalowe.

## 2.2.2. Sterowanie serwomechanizmu

Sterowanie serwomechanizmu ma postać impulsów prostokątnych o czasie trwania 1-2ms, powtarzających się co 20ms. W projekcie za generację odpowiada rozszerzenie pozwalające na generację sygnałów PWM. Umożliwiając regulację obrotu serwonapędu w przypadku 1ms jest to  $-90^\circ$ , natomiast dla wypełnienia o długości 2ms jest to  $90^\circ$ . Poniżej graficznie zostało zaprezentowane działanie serwonapędu.



Rysunek 2.2.2.1 Rysunek przedstawiający zależność wypełnienia od położenia końcówki serwomechanizmu [5]

Jako że sterowanie z PLC odbywa się na 24V, należało użyć odpowiedniego przełożenia, aby napęd się nie spalił. Najodpowiedniejszym rozwiązaniem okazało się zastosowanie transoptorów, której pierwszą ze stron od strony PLC będą mieć połączoną z 24V, a drugą połączoną z elementem sterującym zasilonym pięcioma woltami.

## 2.2.3. Zasilanie serwomechanizmu

Użyty w projekcie serwonapęd posiada trzy przewody, jednym z nich jest pięciowoltowe zasilanie, oznaczać to będzie, że w przypadku zastosowania PLC, którego wyjścia posiadają 24V, będziemy potrzebować odpowiedniego rozszerzenia sterownika PWM, bądź użycia jednego z wyjść, którego napięcie będzie musiało zostać przetworzone przez przetwornik DC/DC z 24V na 5V.

### 3. BUDOWA STEROWNIKA

#### 3.1. Wstęp

Zastosowanie sterowników PLC w rozwoju automatyki przemysłowej znajduje coraz więcej zastosowań, ponieważ w ostatnich latach ich rozwój bardzo przyspieszył, umożliwiając w niektórych przypadkach na wymianę systemów DCS na sterowniki programowalne.

Dzięki zastosowaniu sterownika firmy B&R wraz z jego rozszerzeniami programowanie stało się o wiele prostsze. Dodatkowym plusem tego rozwiązania było zapoznanie się z nowym środowiskiem programowalnym oraz nauczeniem podstawowych funkcji poruszania silnikami serwo za pomocą biblioteki „MapMotion”.

#### 3.2. Wykorzystany sterownik oraz rozszerzenia

##### 3.2.1. Sterownik X20 CP 3586 oraz jego rozszerzenia



Rysunek 3.2.1.1 Zdjęcie sterownika wraz z jego trzema rozszerzeniami komunikacyjnymi

Sterownik użyty w pracy umożliwia bardzo szybkie przetwarzanie danych oraz posiada krótkie cykle pracy sięgające 0,2ms. Kontroler ten jest wyposażony w 3 miejsca dla modułów komunikacyjnych, zróżnicowane wejścia dla magistral komunikacyjnych i połączeń sieciowych. Programowanie PLC odbyć się może poprzez USB, Ethernet, POWERLINK oraz zmienną kartę pamięci [6] [7].

Cechy:

- Procesor Intel z dodatkowymi wejściami,
- Zintegrowane Ethernet, POWERLINK ze zgranymi połączonymi oraz USB,
- 3 miejsca dla modułów komunikacyjnych,
- Wejście na kartę pamięci oraz podręczna pamięć Flash,
- 512MB DDR2 SDRAM,
- Kontrolowana redundancja.

### 3.2.2.Bloki wejścia/wyjścia



Rysunek 3.2.2.1 Zdjęcie wejść oraz wyjść sterownika

- X20 DI9371 – rozszerzenie wejść cyfrowych
- X20 DO9322 – rozszerzenie wyjść cyfrowych
- X20 AI4632 – rozszerzenie wejść analogowych
- X20 AO4632 – rozszerzenie wyjść analogowych
- X20 DC1196 – rozszerzenie dla enkoderów inkrementalnych
- X20 DS1928 – rozszerzenie dla zaawansowanych enkoderów z interfejsem EnDat
- X20 PS4951 – rozszerzenie wejść potencjometrów
- X20 MM4456 – rozszerzenie generacji sygnału PWM [8]



Rysunek 3.2.2.2 Zdjęcie rozszerzenia PWM oraz modbus

- X20 BT9100 – rozszerzenie modułu komunikacyjnego modbus[8]

### 3.2.3.Zasilacz 80PS080X3.10-01



Rysunek 3.2.3.1 Zdjęcie zasilacza modułów oraz rozszerzeń sterownika

Moduł ten odpowiedzialny jest za zasilanie z wejściem 380-480 VAC i wyjściem 24-80 VDC 16.6A o maksymalnej mocy 1000W, wyjście 24VDC 2A, posiada także odbiornik mocy w przypadku hamowania silników [8].

### 3.2.4.ACOPOSmicro 100D Servo



Rysunek 3.2.4.1 Zdjęcie modułu odpowiedzialnego za sterowanie serwosilnikami

Moduł odpowiedzialny jest za sterowanie silnikami Serwo, posiada wejścia dla dwóch silników sterujących osiami. Wejście widoczne z przodu elementu z podłączonymi zielonymi przewodami na rysunku 3.2.4.1 są to enkodery silników, natomiast od dołu do modułu podłączone są wyjścia sterujące silnikami. Rozszerzenie to na podstawie bardzo prostych komend umożliwia precyzyjne sterowanie silnikami [8].

## 4. PROTOTYP

### 4.1. Budowa

Jak wiadomo w przypadku tworzenia jakichkolwiek maszyn na początku zaczyna się od stworzenia prototypu na podstawie dokonanych pomiarów bądź otrzymanych założeń. Umożliwia to sprawdzenie poprawności działania koncepcji, zwizualizowanie, aby ukazać wygląd, a także wielkość oraz sprawdzenie czy wszystkie parametry założeniowe zgadzają się ze stworzonym pomysłem.

Proces tworzenia prototypu jest jednym z ważniejszych elementów podczas tworzenia poprawnego rozwiązania. Czasami umożliwiającym, na jego podstawie, stworzenie kolejnych części makiety bądź ich poprawę.

Prototyp wykonany został z 3mm kartonu, z którego wykonano ścianki, podstawę oraz mechanizm obrotowy. Po stworzeniu makiety, sprawdzona została jakość wprowadzania butelek oraz ich wysuwania z mechanizmu.



Rysunek 4.1.1 Zdjęcie zawiera prototyp potrzebny do sprawdzenia poprawności działania makiety

## **5. MAKIETA**

### **5.1. Budowa**

#### **5.1.1. Wstęp**

Podczas budowy pojawiło się wiele problemów oraz nieprzewidzianych sytuacji opisanych w dalszych częściach podrozdziału.

Pierwszy projekt zainspirowany został filmem „Automatic Bottle Filling & Capping Machine Using PLC” stworzonym przez Nevon Project [9], w którym znalazłem wiele aspektów oraz podstawowych informacji potrzebnych do realizacji projektu. Zostały one w dalszej fazie zmodyfikowane bądź całkowicie zmienione. Niemożliwym było także użycie większej ilości silników aniżeli 2, gdyż moduł obsługujący serwomechanizmy posiada jedynie 2 wejścia. W znacznym stopniu utrudniło to stworzenie makiety, ponieważ niemożliwym było podawanie butelek oraz ich odbieranie za pomocą taśm. Problemem okazało się także jednolite i działające rozwiązańe nakładania nakrętek w taki sposób, aby nie spadały, nie były krzywo nakręcone. Opuszczanie mechanizmu nakręcania opierało się także na hydraulice, która wymagałaby dodatkowych silników, siłowników, zbiornika wyrównawczego i wiele więcej. W celu stworzenia pracy kompaktowej, gdyż o taką zostałem poproszony, niemożliwym było zastosowanie takiego rozwiązania. Głównym zadaniem projektu było nalewanie oraz zakręcanie butelek, przez co skłonienie się w stronę sprężyn pozwoliło na eliminację elektronicznych części, jednak wymagało większego przemyślenia związanego z mechaniką projektu. Projekt nie wymagał także pakowania wielu butelek jednocześnie, gdyż pokazywał jedynie podstawy związane z linią produkcyjną oraz problemami związanymi z jej tworzeniem.

#### **5.1.2. Zasada działania**

W pierwszej części pracy makiety włączane są wszystkie silniki oraz ustalane są ich pozycje „home”, które mogą zostać ustalone gdziekolwiek, gdyż nigdy nie będzie potrzebne odnoszenie się do nich, a jest to wymagane do obrotu silników.

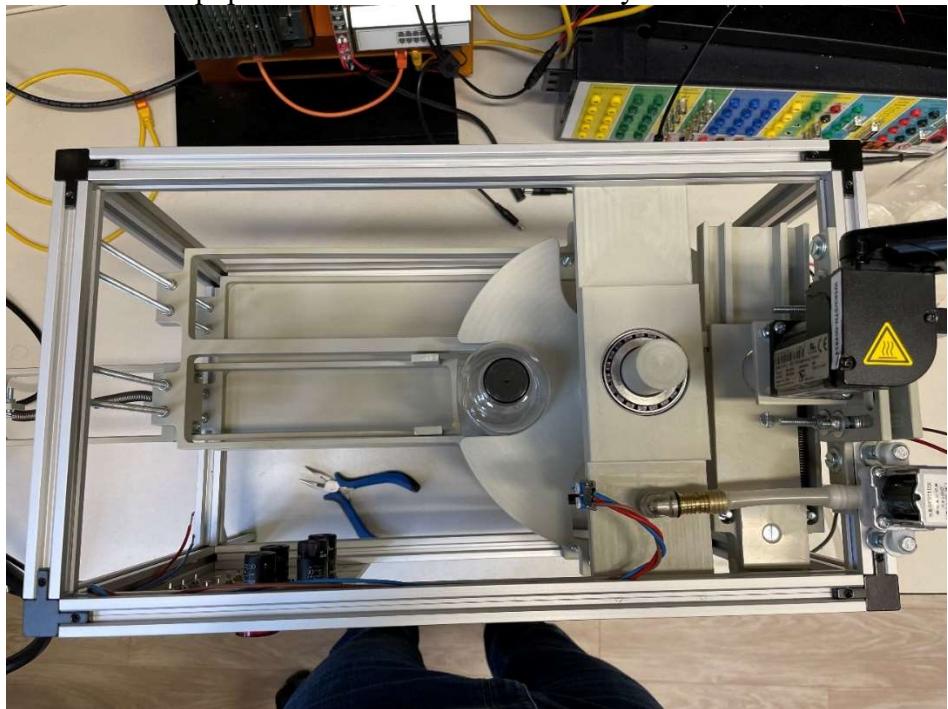
Poniżej opisane zostały kolejne kroki działania pracy inżynierskiej:

- Pozycja wyjściowa butelki umieszczonej w mechanizmie



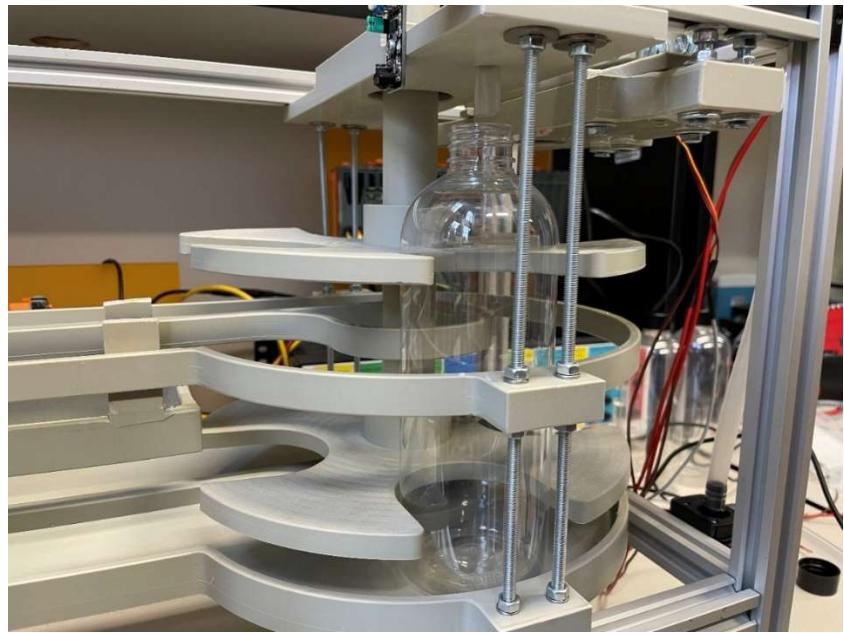
Rysunek 5.1.2.1. Zdjęcie przedstawia pozycję wejściową butelki umieszczonej w mechanizmie podawania

- Pobranie butelki poprzez mechanizm rewolwerowy



Rysunek 5.1.2.2. Zdjęcie ukazuje pracę po pobraniu butelki do mechanizmu rewolwerowego

- Czujnik widoczny na zdjęciu 5.1.2.3 umożliwia zweryfikowanie czy butelka została pobrana, jeżeli tak, to zostaje ona napełniona, w przypadku braku butelki następuje ciągły ruch o stałej prędkości
- W pozycji drugiej (napełniania), uruchamiana jest pompa oraz otwierany zostaje elektrozawór, po jej napełnieniu następuje wyłączenie pompy oraz zamknięcie zaworu



Rysunek 5.1.2.3 Zdjęcie przedstawia butelkę w pozycji drugiej, czyli napełniania.  
W górnej części zdjęcia widać czujnik podczerwieni

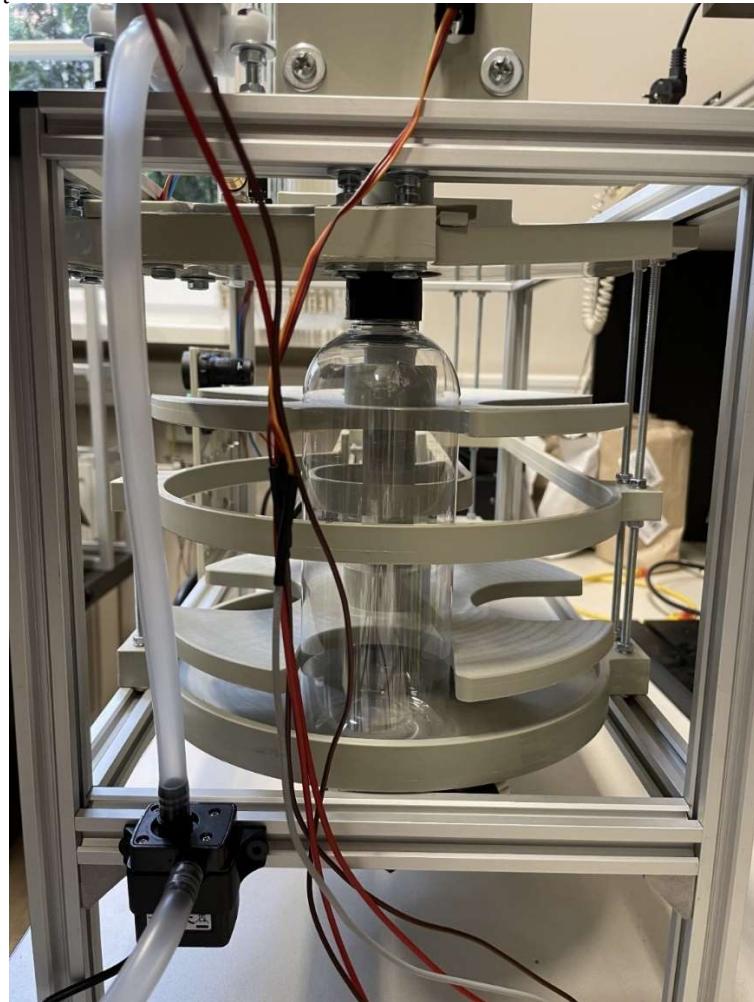
- W kolejnym kroku butelka przemieszczana jest do pozycji 3 zaprezentowanej na poniższym rysunku, jest to miejsce nakręcania butelki



Rysunek 5.1.2.4 Zdjęcie przedstawia prawą stronę makiety, w której znajduje się mechanizm nakręcania. W górnej części zdjęcia widać mechanizm podawania nakrętek

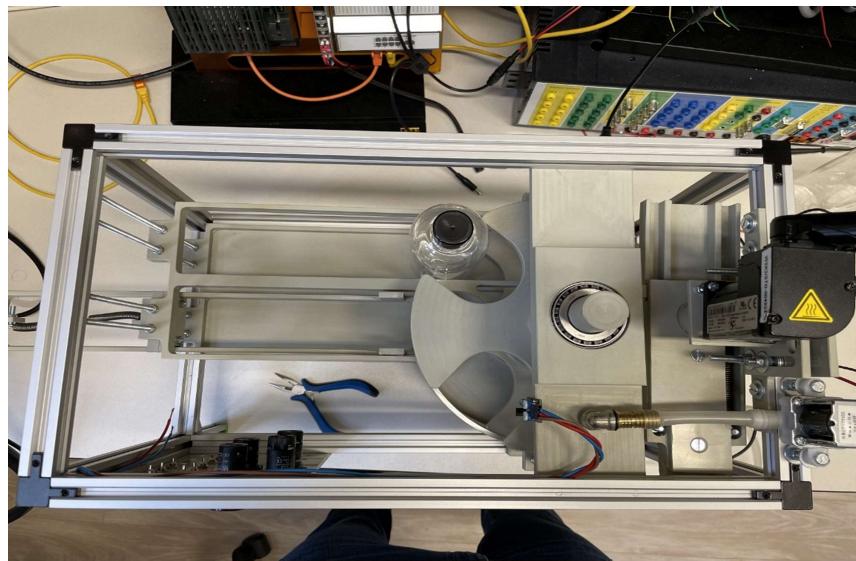
- W górnej części zdjęcia 5.1.2.4 widać element, w którym znajdują się nakrętki. Dzięki obniżeniu serwonapędu zostają nakręcone na szyjkę butelki,

a w miejsce nakręconej po podniesieniu serwa pojawią się kolejna nakrętka.  
Na rysunku 5.1.2.5 widzimy jak wygląda moment, w którym nakręcana jest  
nakrętka



Rysunek 5.1.2.5 Zdjęcie przedstawia moment nakręcenia nakrętki na butelkę

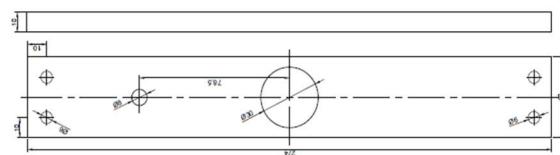
- Ostatecznie zakręcona oraz napełniona butelka przesuwana jest na tor poboczny, na którym może zostać odebrana, bądź pozostawiona jednak mieścią się w nim tylko trzy butelki ze względu na zmniejszoną skalę projektu



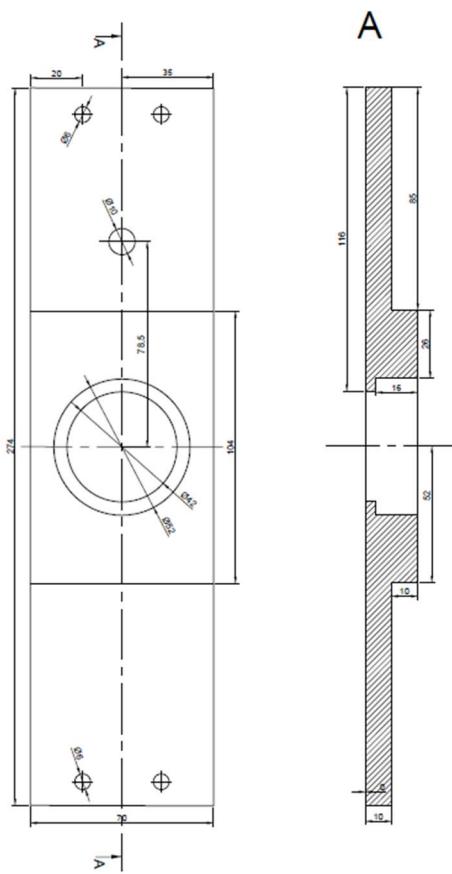
Rysunek 5.1.2.6 Zdjęcie przedstawia moment opuszczenia mechanizmu rewolwerowego przez butelkę

### 5.1.3.Mechanizm rewolwerowy

4



Rysunek 5.1.3.1 Pierwszy schemat elementu konstrukcyjnego podtrzymującego górę mechanizmu rewolwerowego

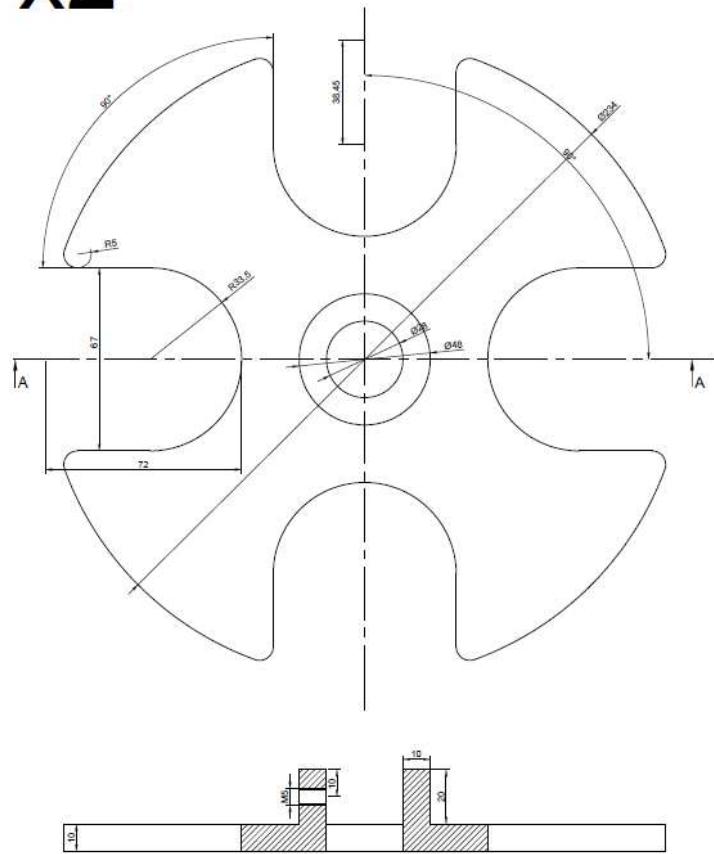


RYS. NR.  
**4**

Rysunek 5.1.3.2 Drugi poprawiony schemat elementu konstrukcyjnego podtrzymującego górę mechanizmu rewolwerowego

Rysunki 5.1.3.1 oraz 5.1.3.2 przedstawiają górny element podtrzymujący główny trzpień mechanizmu rewolwerowego. Porównując rysunki można zauważyc zmianę szerokości oraz grubości elementu, gdyż w środkowy otwór dodano łożysko pozwalające na stabilniejszą oraz precyzyjnieszą pracę serwa zachowując przy tym o wiele lepsze parametry środka obrotu.

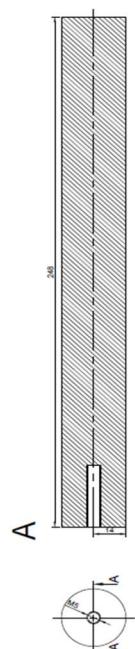
**x2**



RYS. NR.  
**1**

Rysunek 5.1.3.3 Schemat głównego elementu mechanizmu rewolwerowego

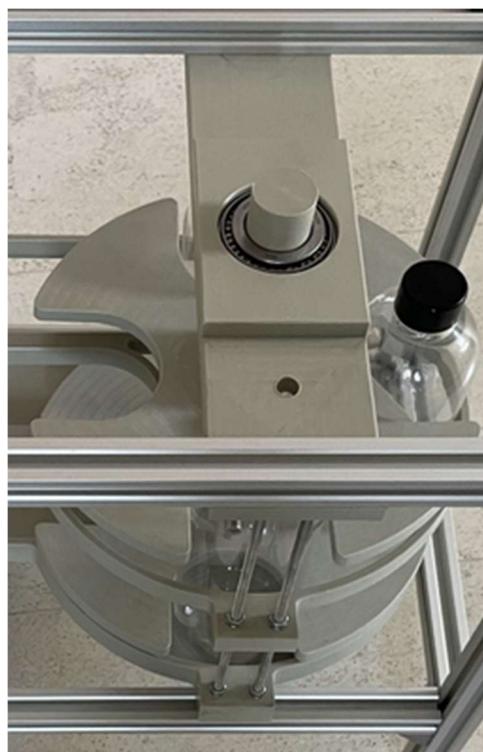
Na rysunku przedstawiony został główny element mechanizmu rewolwerowego, który od pierwszego jego zaprojektowania wymagał dodatkowego przemyślenia, bo wiem nie zakładał dodatkowej regulacji wysokości za pomocą śruby umieszczonej w dolnej części rysunku na pionowej ściance. Dodatkowo projekt nie zakładał dwóch takich elementów, dodanie go pozwoliło na polepszenie obrotu butelki ze względu na zwiększenie punktów w jakich stykał się z butelkami.



RYS. NR.  
5

Rysunek 5.1.3.4 Schemat trzpienia, na którym trzyma się element z rysunku 5.1.3.3

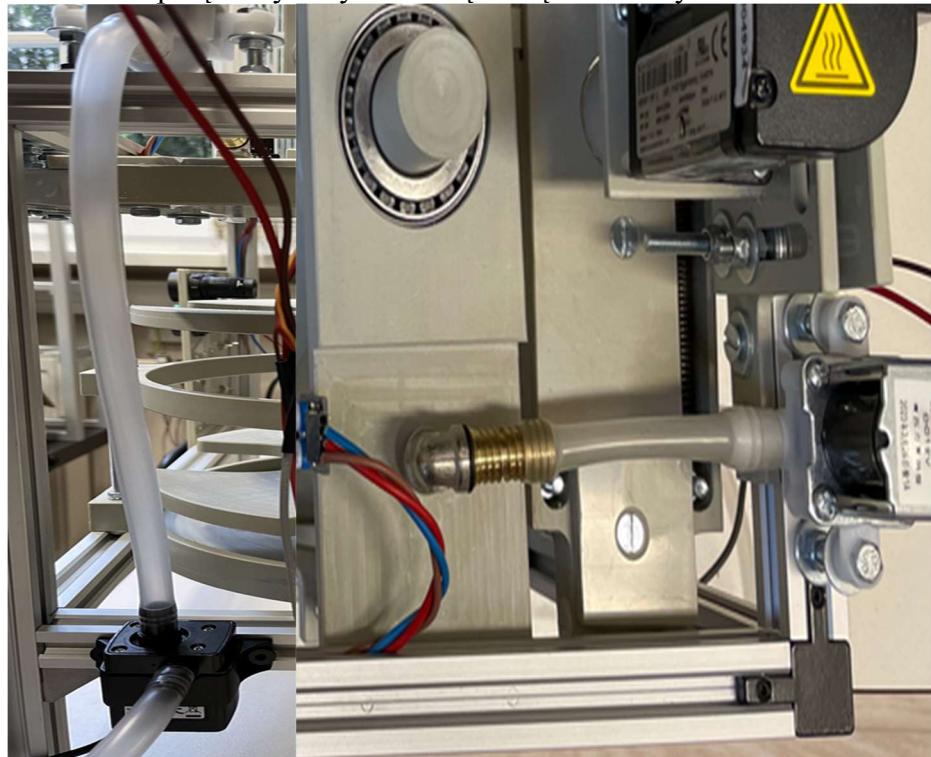
Schemat z rysunku 5.1.3.4 obrazuje schemat głównego trzpienia mechanizmu obrotowego, który po jego pierwszym zamodelowaniu okazał się być idealnym rozwiąza niem.



Rysunek 5.1.3.5 Zdjęcie skrręconego całego mechanizmu

#### 5.1.4.Mechanizm napełniania

Jest to jedna z prostszych części zaprojektowanych w makiecie, nie wymagała żadnych elementów tworzonych w AutoCAD i wycinania na maszynie CNC. Składa się z dwóch elementów połączonych sylikonową rurką o średnicy 8mm/12mm.

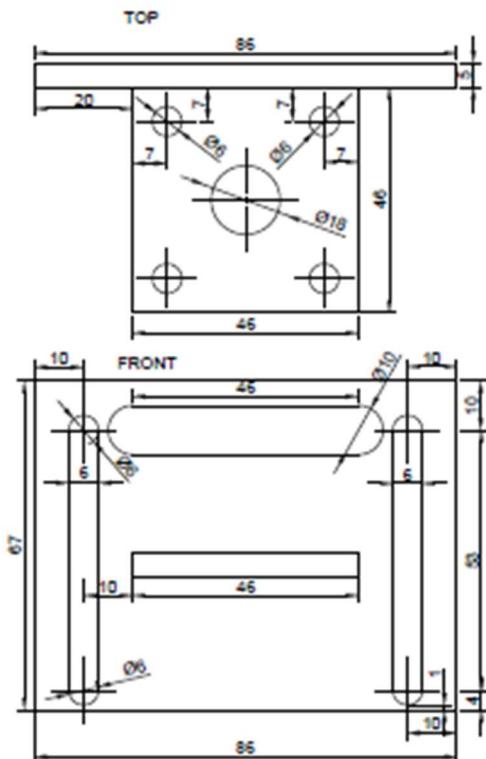


Rysunek 5.1.4.1 Zdjęcie po lewej oraz po prawej przedstawiają cały mechanizm napełniania

Na rysunku 5.1.4.1 po lewej stronie od dołu znajduje się wąż wyjściowy do zewnętrznego zbiornika z cieczą. Zaprojektowane zostało to w taki sposób, aby uniemożliwić przypadkowe rozlanie cieczy w pobliżu projektu. Poprzez pompkę przy mocowaną do ramy przepompowywany do góry jest płyn, który następnie trafia przez zawór (widoczny u góry lewego zdjęcia oraz po prawej stronie prawnego), do butelki umiejscowionej pod metalowym kolankiem. Końcówka odpowiedzialna za wycentrowanie nalewania jest nacięta, aby przed wylotem cieczy dotlenić ją ograniczając jej duże rozlewanie się na boki.

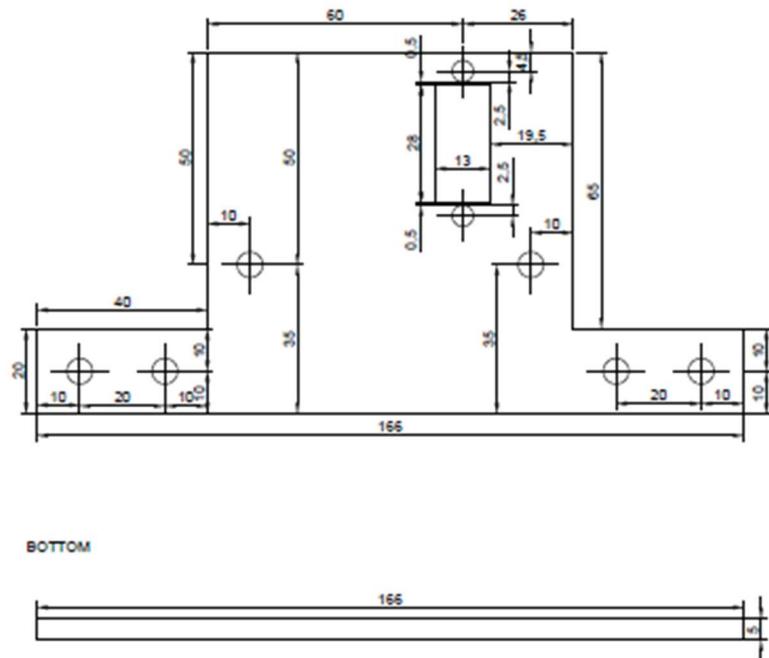
#### 5.1.5.Mechanizm zakręcania

Mechanizm ten jest najbardziej skomplikowanym układem zastosowanym w makiecie. Składa się on z trzech elementów przedstawionych na poniższych rysunkach. Pierwszy z nich pozwala na utrzymanie serwa w pozycji idealnie nad nakrętką oraz ruch liniowy pionowy za pomocą serwonapędu, którego ramie umiejscowione jest w górnym poziomym wycięciu widocznym na dolnej części rysunku 5.1.5.1.



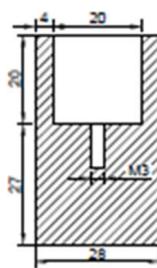
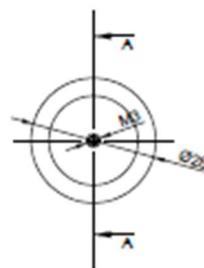
Rysunek 5.1.5.1 Schemat ruchomej części mechanizmu zakręcającego podtrzymującego serwonapęd

Kolejny element widoczny na rysunku 5.1.5.2 odpowiada za zamocowanie całego mechanizmu zakręcania do ramy pracy. Kwadratowy otwór widoczny w górnej części elementu odpowiedzialny jest za trzymanie serwonapędu w miejscu oraz za dopasowanie jego ramienia, w taki sposób, aby umożliwiał ruch pionowy części widocznej na rysunku 5.1.5.1. Element ten okazał się minimalnie wadliwy, jednak poprzez dodanie dwóch dodatkowych otworów na wysokości 20mm od dolnej oraz 10mm od prawej i lewej krawędzi obiektu pozwoliło na stabilniejszą pracę ruchu pionowego.



Rysunek 5.1.5.2 Schemat przedstawiający element trzymający ramę i mechanizm nakręcania

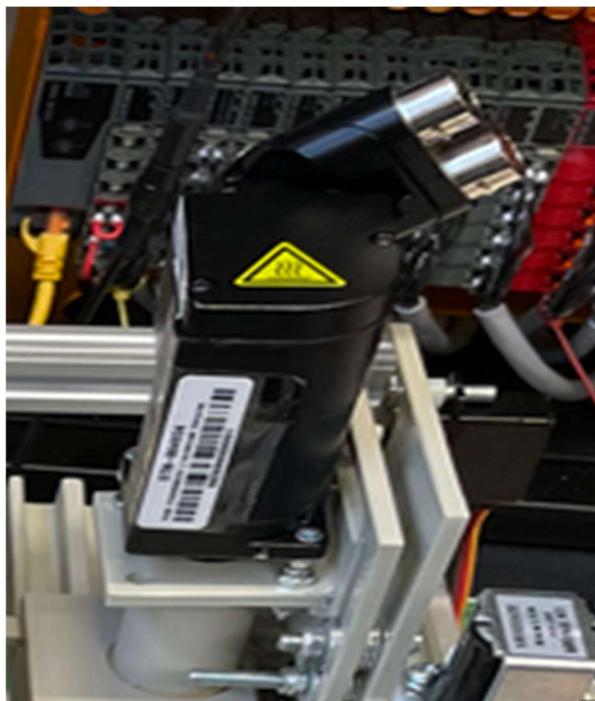
Ostatnią częścią mechanizmu nakręcania jest element naciskający na nakrętki umieszczone w mechanizmie podawania. Ścianki elementu zostały dobrane w taki sposób, aby po podniesieniu serwomechanizmu nie następowało dodatkowe blokowanie nakrętek o końcówkę. W miejscu stykania się obiektu z nakrętką dodatkowo zastosowano podkładki antypoślizgowe, które wyeliminowały ślizganie się nakrętki oraz jej efektywniejsze i jednolite nakręcanie.



Rysunek 5.1.5.3 Schemat przedstawiający nakładkę na końcówkę serwonapędu służącą do nakręcania nakrętek



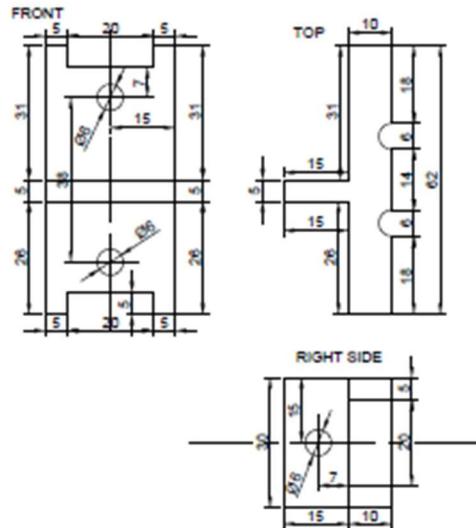
Rysunek 5.1.5.3 Zdjęcie przedstawiające skręcony cały mechanizm podnoszenia i opuszczania



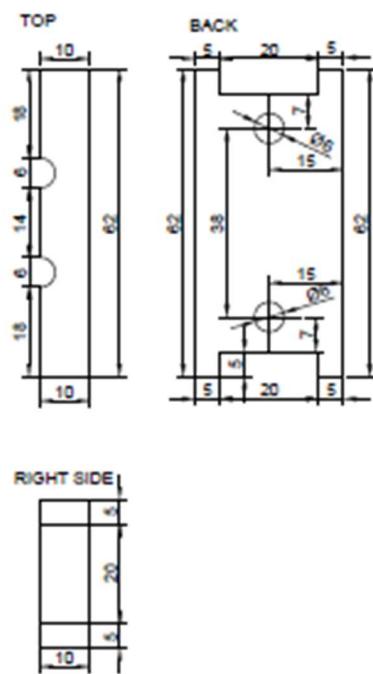
Rysunek 5.1.5.4 Zdjęcie złożonego oraz zamontowanego całego mechanizmu nakręcania wraz z końcówką

### 5.1.6. Mechanizm podawania butelek

Dwa pierwsze rysunki (5.1.6.1 oraz 5.1.6.2) obrazują łącznik między prętami obudowy oraz główną częścią popychającą butelki. Składa się on z dwóch części, gdyż jego montowanie odbyło się, gdy większość pracy była już skręcona.

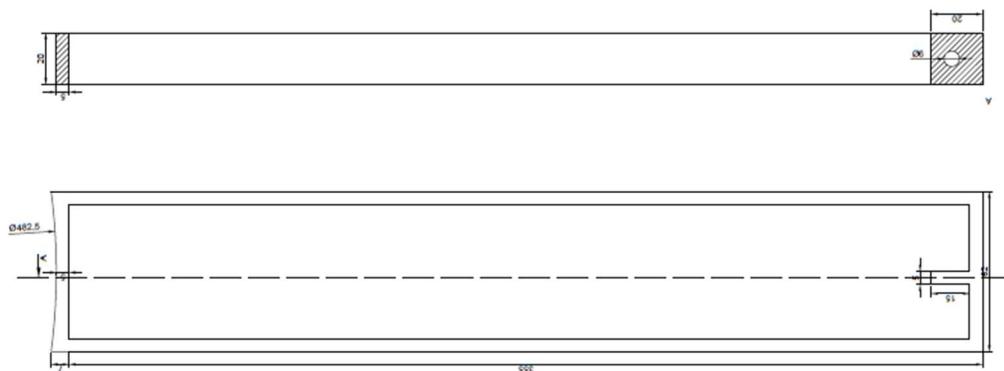


Rysunek 5.1.6.1 Schemat obrazujący pierwszą z części do podtrzymywania mechanizmu podawania butelek



Rysunek 5.1.6.2 Schemat obrazujący drugą z części do podtrzymywania mechanizmu podawania butelek

Poniżej widoczny jest element popychający butelki. Był on także lekko zmodyfikowany ze względu na jego nie zawsze poprawną pracę. Zostały przyspawane do jego górnej części (rysunek 5.1.6.3) dwa prostokątne kawałki umożliwiające poprawę toru ruchu elementu, tym samym niwelując jego występujące czasami blokowanie.



Rysunek 5.1.6.3 Schemat obrazujący główny element popychający butelki do mechanizmu rewolwerowego

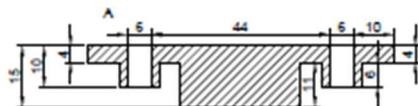
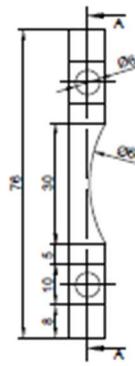
Cały układ działa na podstawie zastosowania dwóch sprężyn umożliwiających ruch butelek w stronę pobierania butelek. Może on pomieścić do 3 butelek jednocześnie, jednak w takim przypadku występuje duży huk, ze względu na siłę naciągu sprężyn. Na rysunku widać także ich mocowanie, które jest po przeciwej stronie, po to, by zniwelować siłę obrotową, by ta z kolei nie doprowadziła do niepoprawnego podawania butelek poprzez wypadnięcie mechanizmu z mocowań.



Rysunek 5.1.6.3 Zdjęcie złożonego oraz zmodyfikowanego mechanizmu podawania butelek

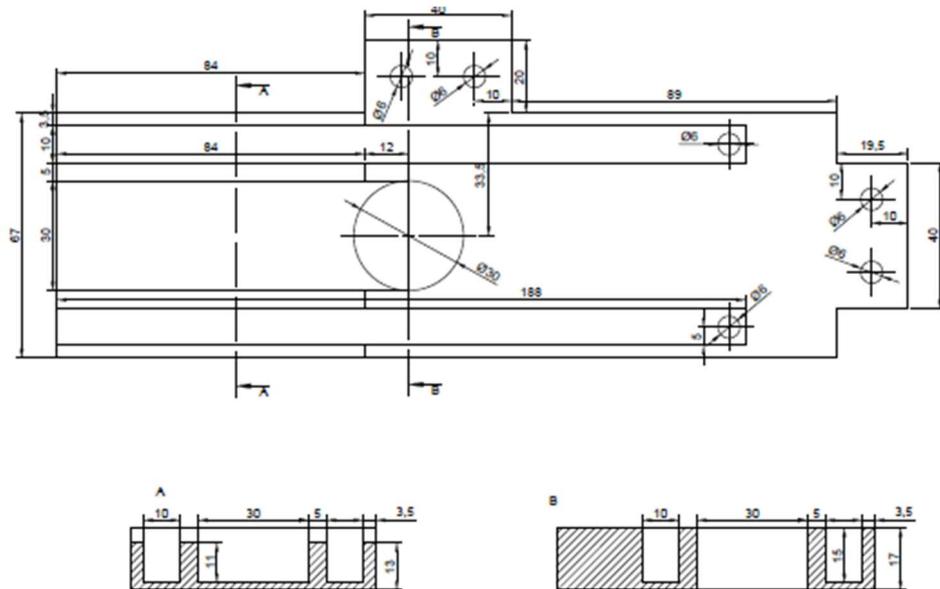
### 5.1.7.Mechanizm podawania nakrętek

Podawanie nakrętek było bardzo skomplikowanym procesem. Pojawiło się wiele trudności związanych z ich wyskakiwaniem, nieidealnym wpasowaniem elementów (po ich przetworzeniu na maszynie CNC) bądź z dopasowaniem sprężyn, które trzeba było rozciągnąć zmniejszając ich siłę naprężenia. Rozwiążanie takie pozwoliło jednak na jednolite i idealnie pionowe nakręcanie butelek eliminując wiele problemów takich jak: krzywe nakręcanie, spadanie podczas przemieszczania się do miejsca nakręcania.



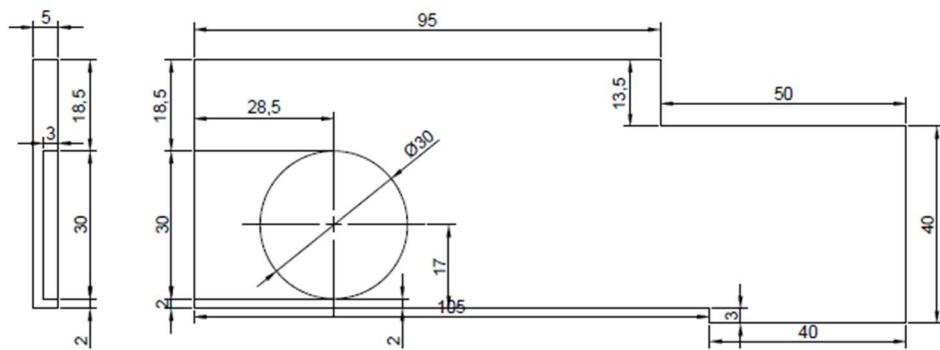
Rysunek 5.1.7.1 Schemat elementu popychającego nakrętki.

Część przedstawiona na rysunku 5.1.7.1 wymagała dodatkowego zeskrobania ścian oraz stworzenia stożkowatych wejść na śruby. Umożliwiło to ich schowanie do elementu dodatkowo umożliwiając dalsze poprawy działania opisane poniżej.



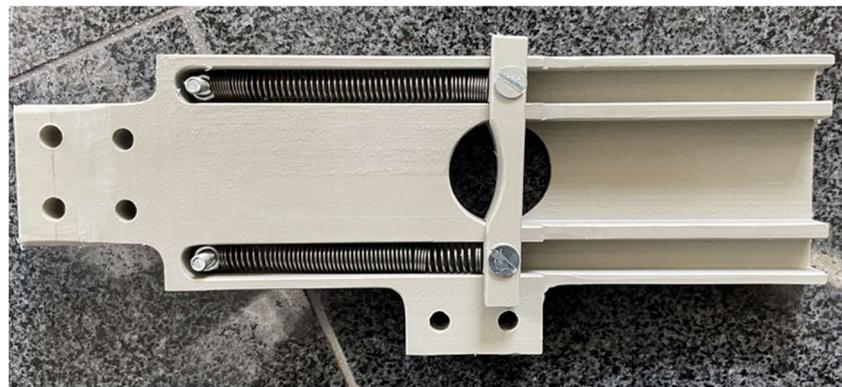
Rysunek 5.1.7.2 Schemat elementu magazynu nakrętek oraz połączenia wszystkich części z ramą.

Rysunek 5.1.7.2 jest schematem zawierającym główny element przechowujący nakrętki oraz cały mechanizm ich posuwu. Umocowane sprężyny w rowkach pozwalają na bardzo łatwy i kompaktowy sposób działania całego elementu. Po jego pierwotnym zamodelowaniu okazał się być wystarczającym rozwiązaniem.



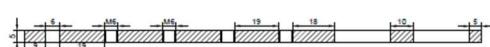
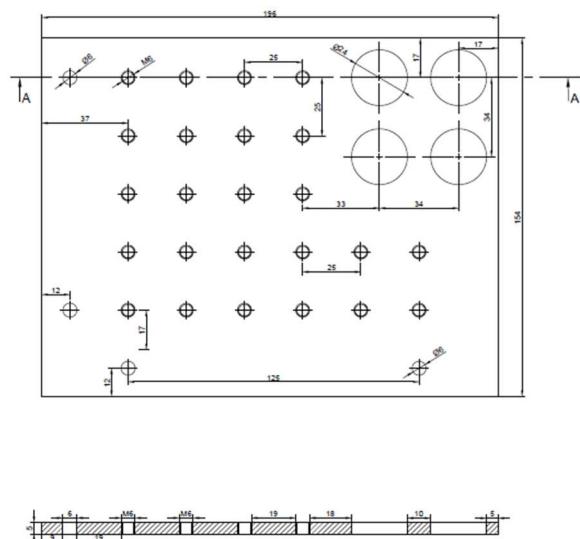
Rysunek 5.1.7.3 Schemat pokrywy na element z rysunku 5.1.7.2, widocznej z dołu

Element widoczny na rysunku 5.1.7.3 pozwala na wyeliminowanie głównego problemu związanego z wyskakiwaniem nakrętek podczas unoszenia się mechanizmu zakręcania. Odgrywa kluczową rolę w trakcie działania mechanizmu. Wymagał on lekkiej poprawy, aby część widoczna na rysunku 5.1.7.1 mogła się pod niego wsunąć bez żadnego problemu, spiłowany został jego lewy kawałek od ścianki aż po środek okręgu.



Rysunek 5.1.7.4 Zdjęcie przedstawiające mechanizm bez nakładki z rysunku 5.1.7.3

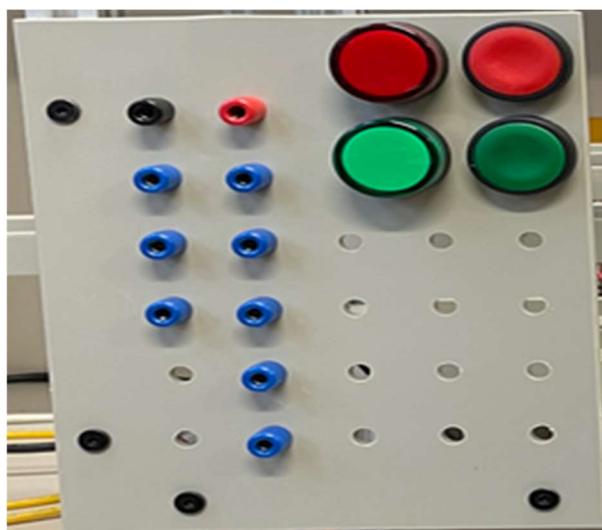
### 5.1.8.Panel sterowniczy



RYS. NR.  
**6**

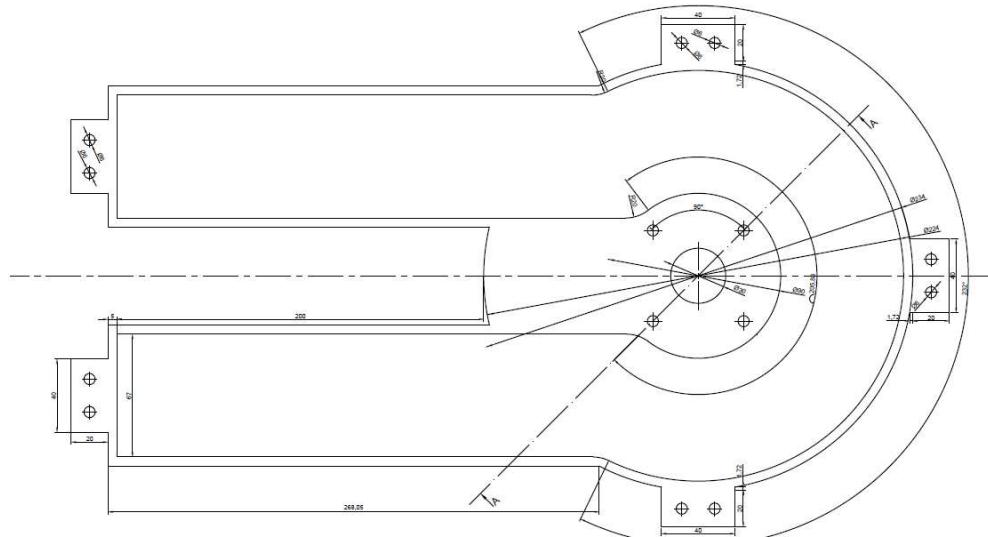
Rysunek 5.1.8.1 Schemat panelu sterowniczego z otworami na przyciski oraz światła

Element znajdujący się na rysunku 5.1.8.1 jest panelem zawierającym wszystkie wejścia/wyjścia służącymi do sterowania i zasilania układu. Posiada on 8 wejść sterujących (kolor niebieski), wejście zasilające 24V (kolor czerwony), masę 24V (kolor czarny), przyciski (zielony – start, czerwony - stop) oraz lampy sygnalizujące aktualny stan pracy.

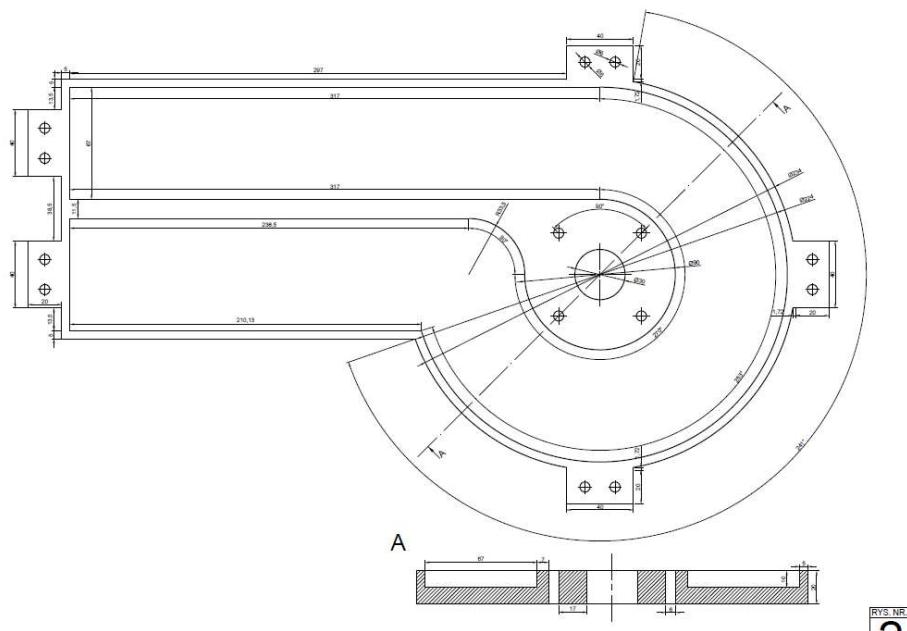


Rysunek 5.1.8.2 Zdjęcie panelu sterowniczego

### 5.1.9.Tor butelek

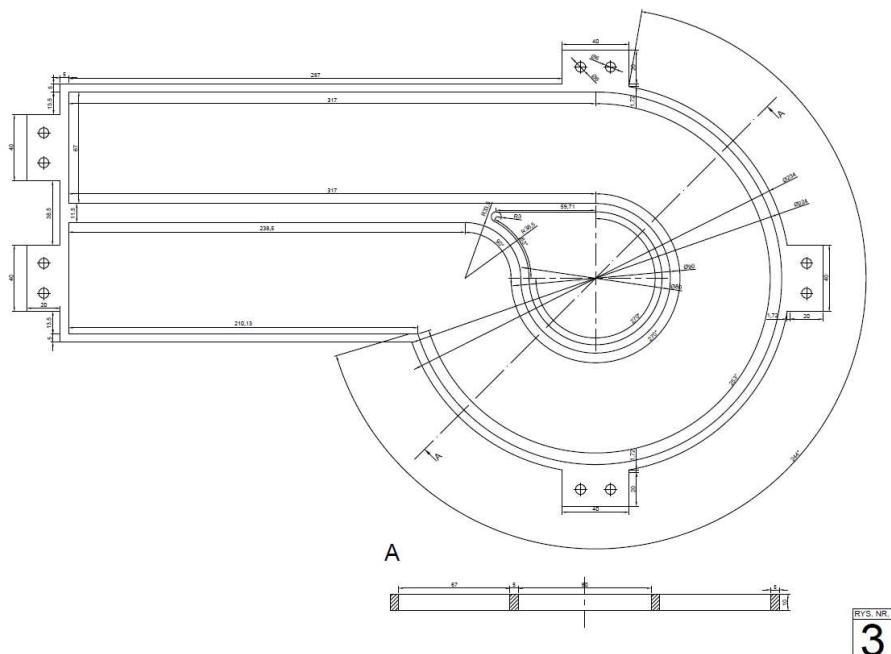


Rysunek 5.1.9.1 Schemat pierwszego toru butelek, który okazał się nieoptymalny



Rysunek 5.1.9.2 Schemat drugiego toru butelek, najoptymalniejsze rozwiązanie

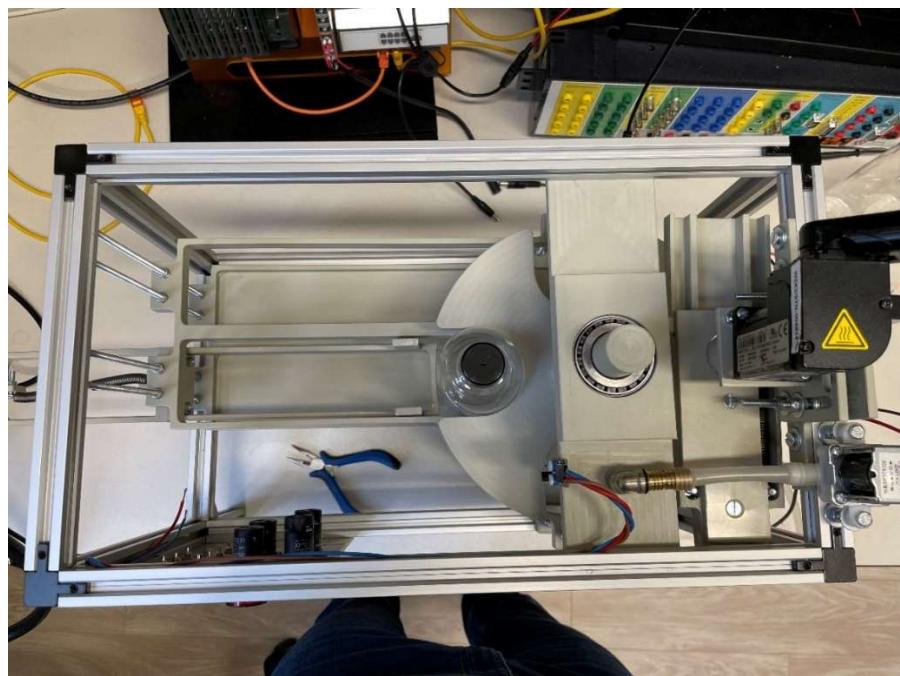
Oba powyższe rysunki są tym samym elementem, jednak pierwszy z nich obrazuje jego wcześniejszą wersję, która wymagała poprawek odnośnie dostarczania oraz wydawania butelek na linię poboczną. Jak można zauważyć, rysunek 5.1.9.2 pozwala pobrać butelki idealnie na wprost, skutkując optymalnym dopasowaniem mechanizmu podającego oraz obracającego, tak aby wejście nie wymagało dalszych zmian. Dodatkowo wyjście zasugerowane w drugim modelu pozwala na automatyczne wysunięcie butelki za każdym razem minimalizując przy tym wszystkie napięcia na mechanizmie, gdyż w przypadku pierwszej jego wersji okazało się, że butelka nie zawsze była oddawana, dodatkowo w każdym przypadku wytwarzaly się bardzo duże siły mechanizmu obrotowego na ścianki pracy.



Rysunek 5.1.9.3 Schemat ścianek toru butelek umieszczony w połowie wysokości butelek, w taki sposób, aby zwiększyć ilość miejsc jej podparcia.

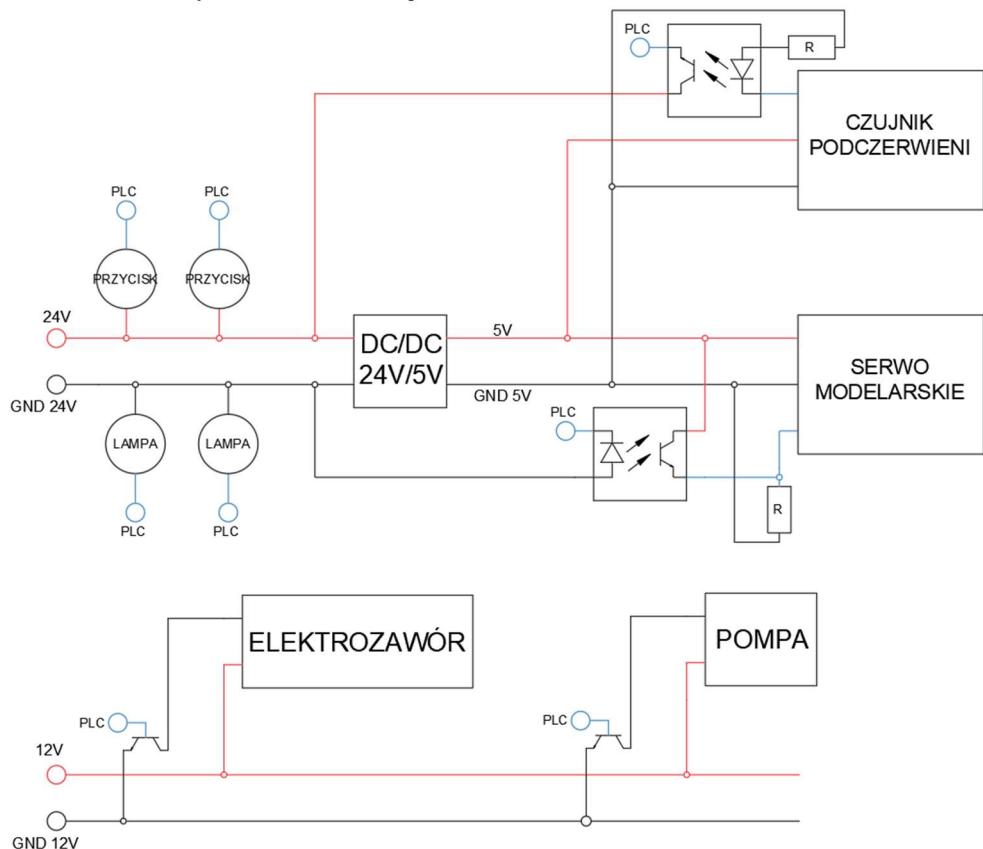
3

Rysunek 5.1.9.3 przedstawia schemat ścianek umieszczonych w połowie wysokości pracy pozwalających na lepsze prowadzenie butelek po torze (został on tak samo przerobiony jak wcześniej opisany element na rysunku 5.1.9.2).



Rysunek 5.1.9.4 Zdjęcie całego toru wraz z połączonymi pozostałymi elementami

### 5.1.10. Połączenia elektryczne



Rysunek 5.1.10.1 Schemat elektryczny układu znajdującego się na makiecie

Przedstawiony na rysunku 5.1.10.1 schemat układu elektrycznego makiety nie prezentuje połączeń silników, gdyż zostały one bezpośrednio podpięte do specjalnego modułu sterownika ACOPOSmicro. Schemat zawiera jedynie połączenia elektryczne zamieszone na makiecie. Na rysunku przedstawione zostały trzy główne obwody, jeden 24V, łączący lampy, przyciski oraz transistory, tak aby sygnał z 5V, mógł być odbierany przez sterownik, gdyż same wejścia są zgodne z ogólnodostępną normą pracy na 24V. Przeskalowanie umożliwia na najoptymalniejsze wykorzystanie układu. Sam obwód 5V uzyskujemy dzięki zastosowaniu przetwornika DC/DC, zasilając wszystkie elementy oraz skalując sygnał sterujący dla czujnika podczerwieni oraz serwonapędu. Elektrozawór wraz z pompą, ze względu na zapotrzebowanie większej ilości amperów, wymagają zastosowania tranzystorów, gdyż w przypadku zastosowania transistory niemożliwym jest uzyskanie większych mocy, służą one głównie do przekazywania sygnałów sterujących.

W projekcie znajdują się takie elementy jak:

- Przetwornik DC/DC LM2596 konwerter mocy Step-Down 3A regulator napięcia – Poprzez użycie tego przetwornika możliwe było precyzyjne doregulowanie napięcia do potrzebnej wartości, gdyż ze względu na mechanizm podnoszenia silnika zakręcającego niewiadome było jakie napięcie powinno być podane ze względu na masę oraz straty podczas podnoszenia. W przypadku podania 5V serwomechanizm osiągał wartość 2kg/cm, natomiast dla 6V miał on już wartość 2.7kg/cm [10].
- Fototranzystor MCT210 pozwalał na separację układu 24V oraz 5V, jednak najważniejszą jego cechą była możliwość sterowania za pomocą 24V elek-

mentów pracujących na 5V. Działanie elementu opiera się na wprowadzeniu na anodę diody albo sygnału sterującego, bądź sygnału z czujnika, a na stronę z emiterem tranzystora w przypadku sygnału sterującego, będzie to wejście na obiekt sterujący, natomiast dla sygnału z czujnika będzie to wejście na sterownik PLC [11].

- Wysoko prądowy tranzystor BC639 jest idealnym rozwiązaniem w przypadku sterowania zaworem oraz pompą, które potrzebują wyższe wartości prądu. Jego kolektor podłącza się najpierw do wyjścia obiektu, a następnie wejście obciążenia do zasilania. Baza sterowana jest za pomocą PLC, gdyż jest ona w stanie wytrzymać do 80V oraz 1A. Emiter podłącza się do uziemienia. Układ taki działa poprzez podanie na bazę sygnału sterującego, dzięki któremu obwód z zasilania zamyka się, załączając obiekt, którym sterujemy [12].

## 5.2. Połaczenie Ethernet z PLC

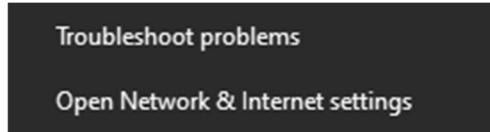
Aby ustawić połaczenie z PLC należy:

- Po prawej stronie paska zadań kliknąć prawym przyciskiem myszy w ikonkę Wi-Fi



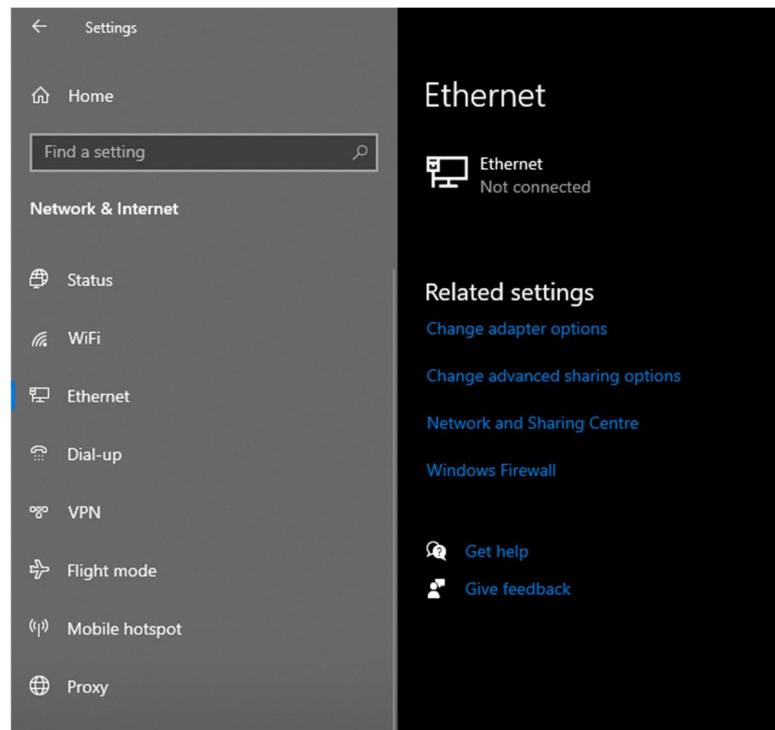
Rysunek 5.2.1 Rysunek przedstawiający ikonę Wi-Fi

- Po kliknięciu na ikonę należy kliknąć opcję „Open Network & Internet settings”.



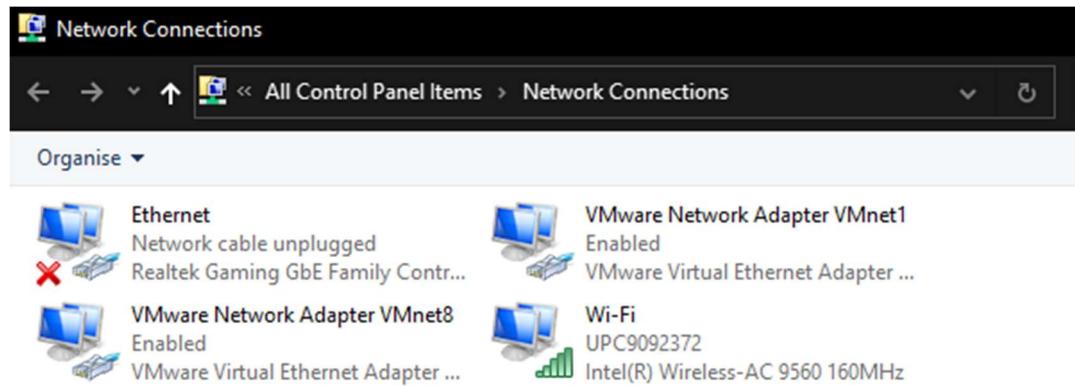
Rysunek 5.2.2 Opcje wyboru po kliknięciu PPM w ikonę Wi-Fi

- W wyświetlonych opcjach należy kliknąć w zakładkę „Ethernet”, następnie kliknąć w opcje „Change Adapter” options.



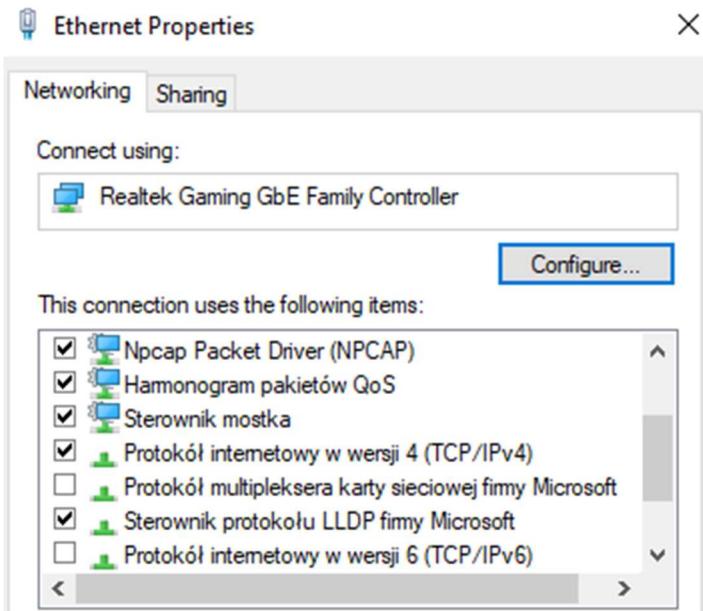
Rysunek 5.2.3 Otworzone okno opcji

- Kolejno na rysunku 5.2.4 w oknie programu klikamy opcje Ethernet.



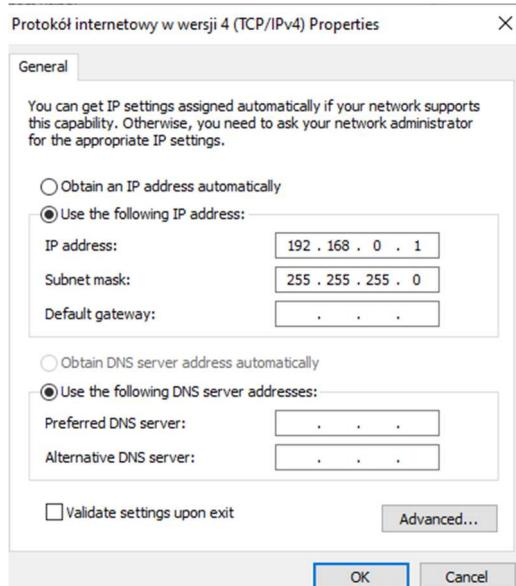
Rysunek 5.2.4 Okno opcji adaptera

- Po kliknięciu ukazuje nam się okno właściwości, w którym należy znaleźć opcję Protokół internetowy w wersji 4 (TCP/IPv4) i kliknąć w niego dwukrotnie.



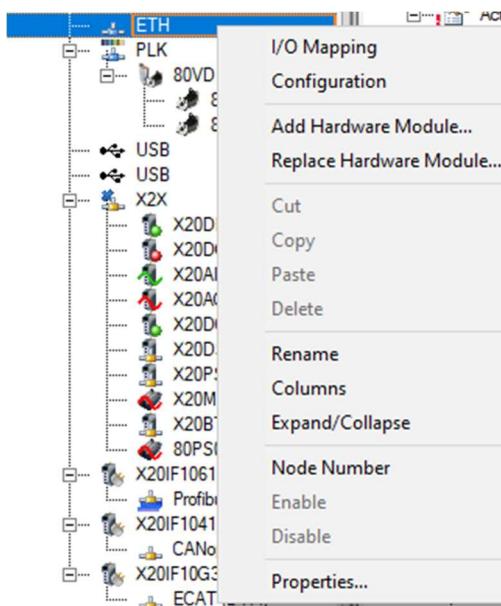
Rysunek 5.2.5 Okno właściwości z wyszukaną opcją protokołu

- Jako ostatnia czynność w komputerze należy ustawić adres IP, ważnym parametrem jest to, aby trzy pierwsze jego części ( 192.168.0.XXX ) odpowiadały kodowi nadanemu potem w programie Automation Studio zgodnie z maską podaną poniżej.



Rysunek 5.2.6 Okno protokołu internetowego

- Po dokonaniu czynności opisanych powyżej należy przejść do konfiguracji w programie. Po jego wyłączeniu, włączamy Physical View oraz klikamy prawym przyciskiem na element ETH – Ethernet i wybieramy opcję „Configuration”.



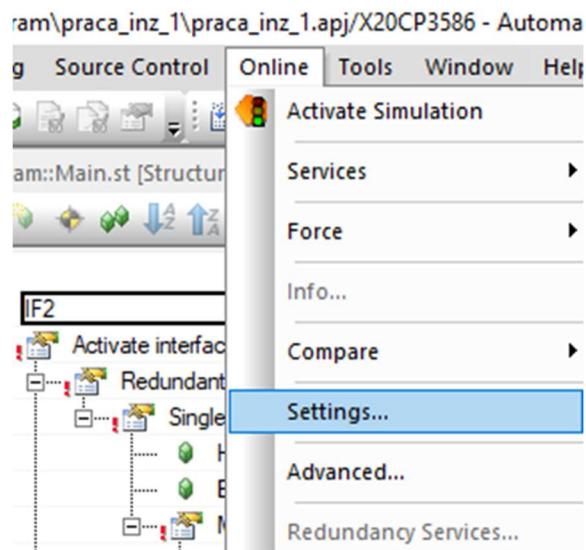
Rysunek 5.2.7 Physical view z klikniętym prawym przyciskiem myszy

- W wyświetlonym oknie należy dokładnie wykonać opisane czynności: po pierwsze ustawić IP adres na wartość o jeden większą w ostatniej części adresu zachowując pierwsze jego trzy składowe ( 192.168.0.X ) tak jak było to w przypadku ustawień w komputerze. Kolejno ustawić Subnet Mask dokładnie taki jaki pojawił się w ustawieniach komputera. Jako trzecią opcję należy wybrać Activate SNMP jako activated, ma to na celu pokazanie PLC naszego komputera, gdyż z opcją odznaczoną my widzimy sterownik, a on nie widzi PC.

Name	Value
IF2	
Activate interface	on
Redundant parameter	Single CPU Project
Single CPU parameters	
Host name	br-automation
Baud rate	auto
Mode	enter IP address manually
IP address	192.168.0.2
Subnet Mask	255.255.255.0
INA parameters	
Activate online communicati...	on
Port number	11159
Redundant INA configuration	Single CPU
ANSL parameters	
Activate online communicati...	on
VNC Servers	
VNC Server 1	
VC Mapping	
VC object name	
Key mapping file	
Passwords	
Authentication	off
Port number	5900
Max. connections	1
Refresh rate	1000
Connection timeout	4
Title	
SNMP parameters	
Activate SNMP	activated
openSAFETY-over-UDP Tunnel	
Activate openSAFETY-over...	off
Modbus parameters	
Activate Modbus communic...	off
Bonding	
Activate bonding	off

Rysunek 5.2.8 Konfiguracja Ethernetu w programie Automation Studio

- Na pasku MENU wchodzimy w opcję Online i klikamy w ustawienia



Rysunek 5.2.9 Okno pojawiające się po najechaniu na pasku MENU

- W wyświetlonym oknie jako ostatnią czynność tworzymy kolejne połączenie TCP/IP, wprowadzamy IP ustawione w konfiguracji sterownika.

Ethernet	Serial	Modem	Remote
Connection name	Use in active config	Source INA node number	Destination IP address
<input type="button" value="New TCPIPConnection"/>	<input checked="" type="checkbox"/>	1	192.168.0.2

Rysunek 5.2.10 przykładowe połączenie TCP/IP

- Po dokonaniu wszystkich zmian komplujemy cały program, a następnie przesyłamy go na kartę pamięci. Po wykonaniu tej czynności włączamy sterownik PLC. Po pewnym czasie powinien się włączyć, a w programie na komputerze w prawym dolnym rogu zaświeci się numer sterownika, adres IP połączenia oraz zielone „RUN” oznaczające prawne połączenie się do sterownika.

### 5.3. Zmienne i konfiguracja portów

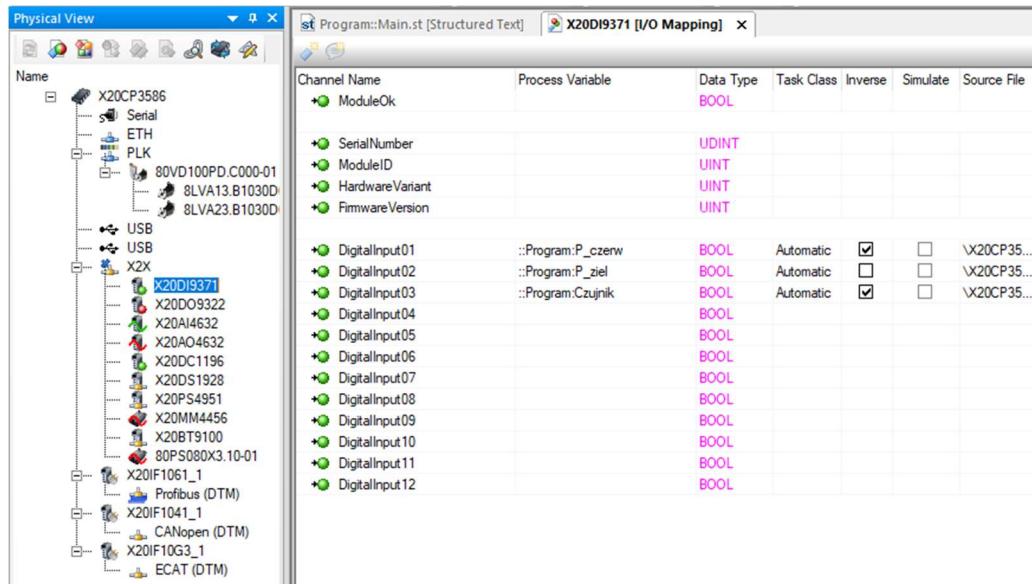
W całym programie znajdują się tylko zmienne lokalne przedstawione na poniższym rysunku 5.3.1, które oznaczają:

- CyclicStep - przełączanie pomiędzy maszyną stanów
- P\_czerw, P\_ziel - zmienne przycisków
- L\_ziel, L\_czerw - zmienne lamp
- Zawor - zmienna włączająca i wyłączająca elektrozawór
- Pompa - zmienna włączająca i wyłączającą pompę
- Czujnik – stan czujnika podczerwieni
- WidthPWM - długość stanu wysokiego PWM
- Wait - zmienna potrzebna do liczenia czasu
- MpAxisBasic\_1, BasicParameters\_1 - zmienne ustawień oraz działania silnika pierwszego
- MpAxisBasic\_1, BasicParameters\_1 - zmienne ustawień oraz działania silnika drugiego
- PeriodPWM\_ms - podanie długości jednego cyklu PWM w ms

Name	Type	& Reference	Constant	Retain	Replacable
Zawor	BOOL	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>
P_ziel	BOOL	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>
P_czerw	BOOL	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>
Pompa	BOOL	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>
L_ziel	BOOL	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>
L_czerw	BOOL	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>
Czujnik	BOOL	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>
WidthPWM	INT	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>
Wait	INT	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>
MpAxisBasic_2	MpAxisBasic	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>
MpAxisBasic_1	MpAxisBasic	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>
BasicParameters_2	MpAxisBasicParT...	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>
BasicParameters_1	MpAxisBasicParT...	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>
PeriodPWM_ms	UINT	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>
CyclicStep	USINT	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>

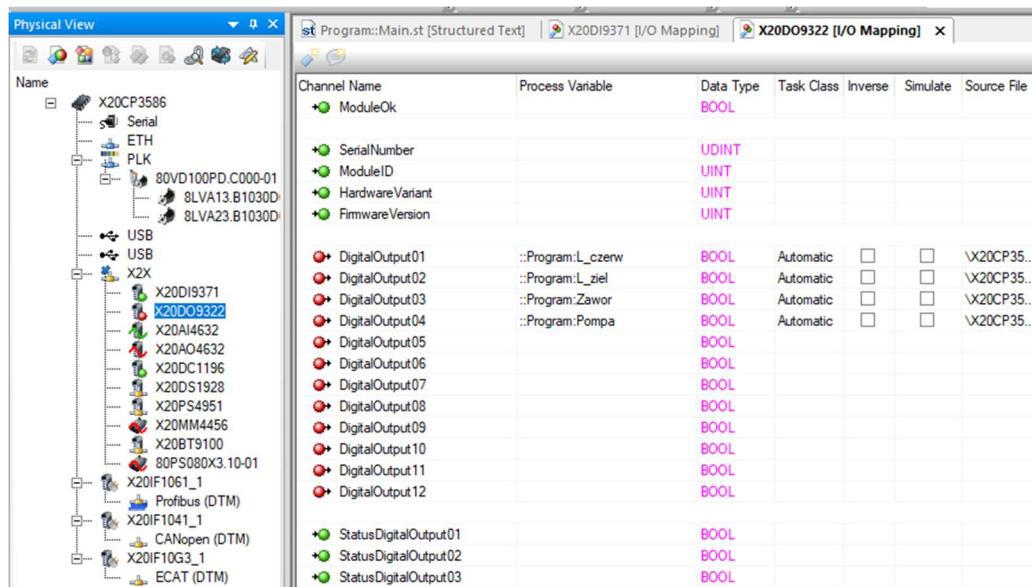
Rysunek 5.3.1 Zmienne lokalne programu

Wejścia cyfrowe widoczne na rysunku 5.3.2 operują jedynie przyciskami oraz czujnikiem, przy czym wyjście czujnika jest 0, gdy zostanie wykryty obiekt w pobliżu, dlatego wejście PLC zostało odwrócone. Przycisk czerwony poprzez jego przypadkowe zakupienie jest NC, oznaczać to będzie, że nieprzyciśnięty podaje wartość 1, w momencie jego kliknięcia 0.



Rysunek 5.3.2 Zmienne modułu wejść cyfrowych

Wyjścia cyfrowe, jak możemy zobaczyć na rysunku 5.1.10.1, sygnały sterujące podawane są albo na lampy, albo na tyrystory/fototranzystory. Dzięki zastosowaniu układów przekazujących sygnał sterujący 24V skalowany jest na mniejsze wartości tak, aby nie zepsuć obiektu sterowanego.



Rysunek 5.3.3 Zmienne modułu wyjść cyfrowych

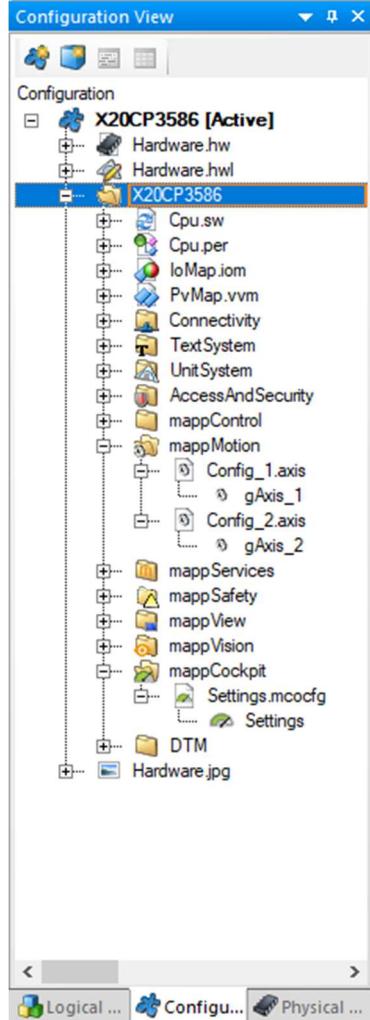
Ustawienia modułu PWM X20MM4456 składają się na główne dwa elementy, określenie jak długo ma trwać jeden cykl w milisekundach ( PeriodDurationPWM ) oraz jak długo ma trwać wypełnienie ( PulseWidtgCurrentPWM ). W przypadku obli-

czania wypełnienia należy wziąć pod uwagę przeskalowanie np.:  $\frac{2}{20} \cdot 32761$  oznaczać będzie, że 2 ms z całego cyklu w tym przypadku ustawionego na 20 ms, będą w stanie wysokim.

Channel Name	Process Variable	Data Type	Task Class	Inverse	Simulate	Source File	Description [1]
<b>ModuleOk</b>							
SerialNumber		UINT					Serial number
ModuleID		UINT					Module ID
HardwareVariant		UINT					Hardware variant
FirmwareVersion		UINT					Firmware version
PeriodDurationPWM	::Program:PeriodPWM_ms	UINT	Automatic	<input type="checkbox"/>	<input type="checkbox"/>	\X20CP35...	Period duration for outputs 01-04
UnderVoltageError		BOOL					Error module supply < 18V
VoltageWarning		BOOL					Warning module supply > 60V
OverVoltageError		BOOL					Error module supply > 80V
OvertemperatureError		BOOL					Error overtemperature
PulseWidthCurrentPWM01	::Program:WidthPWM	INT	Automatic	<input type="checkbox"/>	<input type="checkbox"/>	\X20CP35...	Depending on the module configuration, the current or the PWM pulse width is indicated

Rysunek 5.3.4 Ustawienia I/O modułu PWM MM4456

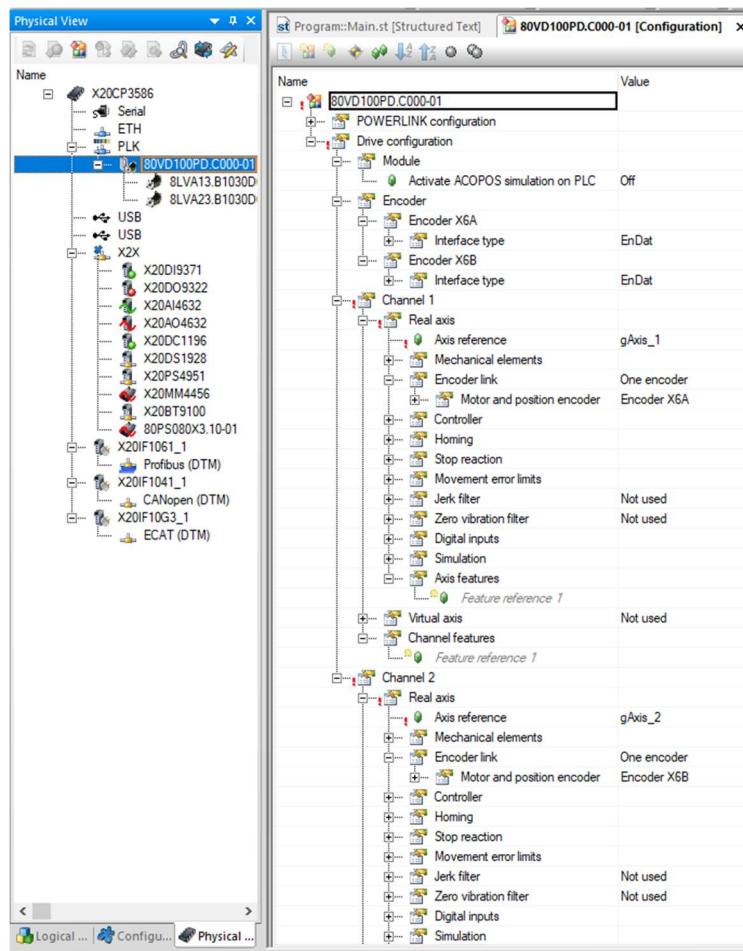
Konfiguracja modułu sterującego silnikami opiera się na stworzeniu dwóch zmiennych gAxis\_1 oraz gAxis\_2 w przypadku sterowania dwoma silnikami. Widocznymi na rysunku 5.3.5 poprzez przeciągnięcie na folder mappMotion pliku Axis z menu wyboru po prawej stronie aplikacji. Jest to plik posiadający konfigurację silnika wraz z jego potrzebnymi komendami.



Rysunek 5.3.5 Okno ukazujące dodane konfiguracje dla dwóch osi

Po stworzeniu plików należy przejść do konfiguracji naszego ACOPOSmicro klikając prawym przyciskiem na moduł i wchodząc w zakładkę Configuration. W otrzyma-

nym oknie należy zmienić Encoder X&A oraz X&B na EnDat, Axis reference dla silnika pierwszego na gAxis\_1 oraz dla silnika drugiego gAxis\_2.



Rysunek 5.3.6 Okno konfiguracji modułu ACOPOS

#### 5.4. Kod programu

Wiedza potrzebna do napisanie programu znaleziona w źródłach internetowych, pomocy wsparcia B&R oraz książki odnośnik [13].

```

PROGRAM _INIT
    //inicjalizacja zmiennych programu
    P_czerw := 1;
    P_ziel := 0;
    Czujnik := 0;
    L_czerw := 1;
    L_ziel := 0;
    Zawor := 0;
    Pompa := 0;
    CyclicStep := 0;
    Wait := 0;

    //inicjalizacja PWM
    PeriodPWM_ms := 20;
    //Przeskalowanie długości 20 ms jest na 0 -> 32761
    //dlatego dodajemy 2/20 aby silnik początkowo posiadał
    //wypełnienie 2 ms
    WidthPWM := 2/20*32761;

    //inicjalizacja bloków silników
    MpAxisBasic_1(MpLink := ADR(gAxis_1), //ustawienie adresu pliku zmienianego
                  Enable := TRUE, //włączenie działania silników

```

```

Parameters := ADR(BasicParameters_1)); //zaadresowanie
                                         //pliku z parametrami silnika
MpAxisBasic_2(MpLink := ADR(gAxis_2),Enable := TRUE,Parameters :=
ADR(BasicParameters_2));

//inicjalizacja parametrów bloków silników
BasicParameters_1.Velocity := 10;
BasicParameters_1.Acceleration := 50;
BasicParameters_1.Deceleration := 50;
BasicParameters_1.Direction := 0;
BasicParameters_1.Distance := 250;

BasicParameters_2.Velocity := 30;
BasicParameters_2.Acceleration := 50;
BasicParameters_2.Deceleration := 50;
BasicParameters_2.Direction := 0;
BasicParameters_2.Distance := 1500;

END_PROGRAM

PROGRAM _CYCLIC

//przełącznaie logiki jeżeli przycisk czerwony
IF P_czerw = 1 THEN //logika normalna bo przycisk odwrócony,
                     //ale sygnał na wejściu tez
    L_czerw := 1;
    L_ziel := 0;
END_IF

//przełączanie logiki jeżeli przycisk zielony
IF P_ziel = 1 THEN
    L_ziel := 1;
    L_czerw := 0;
END_IF

//jeżeli przycisk zielony włączony wtedy wykonujemy program
IF L_ziel = TRUE THEN
    //maszyna stanów
    //stany 0,1,2 są wymagane do włączenia silników oraz ich konfiguracji
    CASE CyclicStep OF
        0:
            //załączanie automatycznego strojenia silników
            MpAxisBasic_1.AutoTune := TRUE;
            //ustawienie parametrów potrzebnych do strojenia
            BasicParameters_1.AutoTune.MaxPositionError := 1;
            BasicParameters_1.AutoTune.MaxCurrentPercent := 20;

            MpAxisBasic_2.AutoTune := TRUE;
            BasicParameters_2.AutoTune.MaxPositionError := 1;
            BasicParameters_2.AutoTune.MaxCurrentPercent := 20;

            //jeżeli oba silniki się dostroją przechodzimy do stanu 1
            IF MpAxisBasic_1.Info.AutoTuneDone AND
                MpAxisBasic_2.Info.AutoTuneDone THEN
                CyclicStep := 1;
            END_IF
        1:
            // załączenie zasilania ruchu silników
            MpAxisBasic_1.Power := TRUE;
            MpAxisBasic_2.Power := TRUE;

            //jeżeli oba silniki są włączone przechodzimy do stanu 2
            IF MpAxisBasic_1.PowerOn AND MpAxisBasic_2.PowerOn THEN
                CyclicStep := 2;
            END_IF
        2:
            //ustawianie home dla obu silników, jest to wymagane
            //ustawienie aby nimi ruszyć
            MpAxisBasic_1.Home := TRUE;
            MpAxisBasic_2.Home := TRUE;

            //jeżeli home są ustawione wtedy przechodzimy do stanu 3
            IF MpAxisBasic_1.IsHomed AND MpAxisBasic_2.IsHomed THEN
                CyclicStep := 3;
            END_IF
    END_CASE
END_PROGRAM

```

```

3:
(*Stan początkowy, pobieranie butelki, jeżeli czujnik
wykryje butelkę wtedy przechodzi do kolejnego stanu
    - silnik obracający rewolwer ON
    - silnik zakrętki OFF
    - zawór OFF
    - pompa OFF*)
//włączenie obracania silnika mechanizmu rewolwerowego
IF MpAxisBasic_1.MoveVelocity = FALSE THEN
    MpAxisBasic_1.MoveVelocity := TRUE;
END_IF

//w przypadku wykrycia butelki przechodzimy do stanu 4
IF Czujnik THEN
    CyclicStep := 4;
END_IF

4:
(*Stan butelki pod nalewaniem, brak butelki pod zakrętką
    - silnik obracający rewolwer OFF
    - silnik zakrętki OFF
    - zawór ON
    - pompa ON*)
//wyłączenie mechanizmu rewolwerowego
IF MpAxisBasic_1.MoveVelocity = TRUE THEN
    MpAxisBasic_1.MoveVelocity := FALSE;
END_IF

//włączenie pompy oraz zaworu, każdy cykl zajmuje 100ms,
//po mniej więcej 50 cykli czyli 5s (napełnieniu butelki)
//zerujemy czas i przechodzimy do cyku 5
Zawor := 1;
Pompa := 1;
Wait := Wait + 1;
IF Wait > 50 THEN
    Wait := 0;
    CyclicStep := 5;
END_IF

5:
(*Stan obrotu o 90 stopni w przód
    - silnik obracający rewolwer ON
    - silnik zakrętki OFF
    - zawór OFF
    - pompa OFF*)
//wyłączenie zaworu oraz pompy z cyku 4
Zawor := 0;
Pompa := 0;

//włączenie obrotu Addytywnego czyli dodanie
//do aktualnego stanu obrotu dodatkowych 250 jednostek
//będących obrotem o 90 stopni
IF MpAxisBasic_1.MoveAdditive = FALSE THEN
    MpAxisBasic_1.MoveAdditive := TRUE;
END_IF

//jeżeli czujnik po obrocie o 90 stopni to
IF MpAxisBasic_1.InPosition THEN
    //jeżeli czujnik wykryje kolejną butelkę
    //przechodzimy do cyku 6, jeżeli nie
    //wykryje przechodzimy do cyku 7
    IF Czujnik THEN
        CyclicStep := 6;
    ELSE
        CyclicStep := 7;
    END_IF
END_IF

6:
(*Stan, dwie butelki w slotach, pod nalewakiem i pod
nakrętką
    - silnik obracający rewolwer OFF
    - silnik zakrętki ON
    - zawór ON
    - pompa ON*)
//wyłączenie ruchu addytywnego

```

```

        IF MpAxisBasic_1.MoveAdditive = TRUE THEN
            MpAxisBasic_1.MoveAdditive := FALSE;
        END_IF

        //obniżenie serwa
        IF WidthPWM = (2/20*32761) THEN
            WidthPWM := 1/20 * 32761;
        END_IF

        //włączenie silnika zkręcającego
        IF MpAxisBasic_2.MoveAdditive = FALSE THEN
            MpAxisBasic_2.MoveAdditive := TRUE;
        END_IF

        //włączenie pompy oraz zaworu po 5s wyłączenie
        Zawor := 1;
        Pompa := 1;
        Wait := Wait + 1;
        IF Wait > 50 THEN
            Zawor := 0;
            Pompa := 0;
        END_IF

        //jeżeli zkręcanie się skończy to
        IF MpAxisBasic_2.InPosition THEN
            //wyłączamy ruch addytywny
            MpAxisBasic_2.MoveAdditive := FALSE;
            //podnosimy ramię serwa modelarskiego
            WidthPWM := 2/20 * 32761;
        END_IF

        //jeżeli wszystkie czynności się skończą to zerujemy
        // licznik czasu i przechodzimy do cyklu 5
        IF MpAxisBasic_2.InPosition AND Wait > 50 THEN
            Wait := 0;
            CyclicStep := 5;
        END_IF

7:
        (*Stan obrotu o 90 stopni w przód
         - silnik obracający rewolwer OFF
         - silnik zkrętki ON
         - zawór OFF
         - pompa OFF*)

        //wyłączamy ruch silnika obracającego mechanizm rewolwerowy
        IF MpAxisBasic_1.MoveVelocity = TRUE THEN
            MpAxisBasic_1.MoveVelocity := FALSE;
        END_IF

        //obniżamy ramię serwa modelarskiego
        IF WidthPWM = (2/20 * 32761) THEN
            WidthPWM := 1/20 * 32761;
        END_IF

        //włączamy silnik zkręcający
        IF MpAxisBasic_2.MoveAdditive = FALSE THEN
            MpAxisBasic_2.MoveAdditive := TRUE;
        END_IF

        //po skończeniu zkręcania
        IF MpAxisBasic_2.MoveDone THEN
            //przejście do cyku 4
            CyclicStep := 4;
            //wyłączenie silnika i podniesienie ramienia serwa
            //modelarskiego
            MpAxisBasic_2.MoveAdditive := FALSE;
            WidthPWM := 2/20 * 32761;
        END_IF

    END_CASE

ELSE
    (*w przypadku przyciśniętego przycisku czerwonego wszystko się wyłącza*)
    Pompa := 0;
    Zawor := 0;

```

```

    IF MpAxisBasic_1.MoveVelocity = TRUE THEN
        MpAxisBasic_1.MoveVelocity := FALSE;
    END_IF

    IF MpAxisBasic_1.MoveAdditive = TRUE THEN
        MpAxisBasic_1.MoveAdditive := FALSE;
    END_IF

    IF      WidthPWM = (1/20 * 32761) THEN
        WidthPWM := 2/20 * 32761;
    END_IF

    IF MpAxisBasic_2.MoveAdditive = TRUE THEN
        MpAxisBasic_2.MoveAdditive := FALSE;
    END_IF
END_IF

(*załączanie pracy silników*)
MpAxisBasic_1();
MpAxisBasic_2();

END_PROGRAM

PROGRAM _EXIT

END_PROGRAM

```

## 5.5. Zabezpieczenia

Układ przedstawiony na rysunku 5.1.10.1 poprzez zastosowanie transceptorów posiada separację galwaniczną strony 24V od 5V. Dodatkowo rezystancja wejść PLC jest bardzo duża pozwalając na ochronę układu w przypadku wystąpienia zwarcia. Dlatego w przypadku wystąpienia uszkodzenia, będzie on w stanie zapobiec negatywnym skutkom, gdyż sam posiada wbudowane zabezpieczenia, które w tym przypadku będą wystarczające. Dodatkowym rozwiązaniem, które musi pojawić się na stanowisku pracy jest tzw. „grzybek”, poprzez odpowiednie połączenie do PLC, a dokładniej do jego specjalnego wejścia pozwala na bezpośrednie odcięcie zasilania makiety. Jest to przyムsowe rozwiązanie, gdyż w przypadku podłączenia go do normalnych wejść PLC oraz zawieszenia sterownika mogłoby okazać się, że maszyna nie zakończyła swojej pracy doprowadzając do katastrofalnych strat, bądź utraty życia.

## 6. PODSUMOWANIE

### 6.1. Wnioski

Ostateczny model makiety przeszedł wiele modyfikacji od założonego planu, był to proces długi, pozwalający na nauczenie się rozwiązywania problemów związanych z mechaniką, elektroniką, a także programowaniem, co widoczne było w sprawozdaniu. Na zdjęciu 6.1.1 widoczny jest pełen podłączony do sterownika PLC model oraz układ elektroniczny.



Rysunek 6.1.1 Zdjęcie ostatecznego skończonego projektu wraz z jego połączeniem

Sama makieta pozwala na zobrazowanie prostego rozwiązania linii przemysłowej w małej skali, pozwalając studentom na przybliżenie procesu automatyki. Dodatkowym atutem pracy jest możliwość jej łatwej zmiany, bądź ulepszeń, gdyż każdy z jej elementów został przemyślany w taki sposób, aby wszystko dało się doregulować, umożliwiając napełnianie oraz zakręcanie butelek oryginalnych przeznaczonych do pracy, butelek szklanych 0,33l, butelek szklanych 0,5l, butelek plastikowych i ze zmianą mechanizmu zakręcania nawet puszek. Wprowadziła to bardzo dużą gammę obiektów obrabianych, co poszukiwane jest na rynku pracy.

Cel i zakres pracy zostały zrealizowane do końca, makieta może w przyszłości zostać przerobiona lub zaprogramowana w inny sposób. Umożliwia ona zapoznanie się z:

- Biblioteką mappMotion
- Konfiguracją modułów oraz rozszerzeń sterownika X20
- Mapowaniem wejść I/O
- Modułem sygnału PWM
- Diagnostyka: Watch, Logger, SDM

Należy także wspomnieć o pomocy ze strony firmy B&R, która posiada wiele materiałów dydaktycznych online, a także osobą z suportu, które rozwiązały wiele problemów z konfiguracją sterownika.

## **7. BIBLIOGRAFIA**

- [1]<https://hovlerengineering.pl/od-pomyslowego-greka-do-maszyny-ktora-sama-liczy/>
- [2]<https://www.linkedin.com/pulse/common-problems-automatic-bottle-filling-machine-well-solomon-oyedo>
- [3]<https://www.br-automation.com/pl/produkty/motion-control/acoposmicro/inverter-modules/80vd100pdc000-01/>
- [4] <https://www.br-automation.com/pl/search/?q=serwonap%C4%99d>
- [5] <https://forbot.pl/blog/leksykon/serwomechanizm> [17]  
<https://www.youtube.com/watch?v=JdXzMI1PXcs>
- [6] <https://www.br-automation.com/pl/produkty/plc-systems/x20-system/x20-plc/x20cp3586/>
- [7] B&R "X20 System User's Manual" 2023
- [8] <https://www.br-automation.com/pl/produkty/>
- [9] <https://www.youtube.com/watch?v=JdXzMI1PXcs>
- [10] Texas Instrument, „LM2596 SIMPLE SWITCHER Power Converter 150-kHz 3-A Step-Down Voltage Regulator”, 2023
- [11] ISOCOM COMPONENT, „MCT210 optically coupled isolator phototransistor output”
- [12] ON Semiconductor, „BC639 High Current Transistor”, 2005
- [13] Book on Demand, “PLC Controls with Structured Text (ST)” część 3, 23.07.2020, 9788743018551

Wszystkie hiperłącza działają na dzień 15.12.2023.