**Temat: Synteza cyfrowego układu stabilizacji poziomu cieczy**

**1. Podstawowe założenia:**

* Platforma wykonawcza algorytmów: Sterownik PLC X20CP1486
* Graficzny interfejs użytkownika: Komputer, tablet, smartfon
* IDE: Automotion Studio
* Cel pracy: Implementacja i praktyczna weryfikacja oprogramowania sterującego poziomu cieczy.
* Efekty końcowe: Dokumentacja wykonawcza prototypu urządzenia, obejmująca:
  + rysunki poglądowe oraz techniczne,
  + wykaz elementów gotowych,
  + syntezę algorytmu sterowania,
  + stanowisko badawcze,
  + oprogramowanie sterujące,
  + instrukcja (schemat funkcjonalny, schemat elektryczny, wykaz elementów, wykaz punktów (items) oraz tagów, opis działania w postaci słownej i schematu blokowego, wytyczne uruchomieniowe i serwisowe).
* Nakład pracy: 50h / osoba
* Zarządzanie projektem: Trello
* Podział zadań:
  + Koordynator – Łukasz Krzyżanowski
  + Programiści – Łukasz Krzyżanowski, Dominik Smutek, Paweł Wywijas
  + Konstruktorzy – Wojciech Pawluć, Mariusz Augustynek, Wojciech Humienik
  + Projektanci GUI – Patryk Janik, Dominik Smutek
  + Archiwiści – Alicja Kapiszka
  + Wizualizacja 3D – Jakub Izbicki

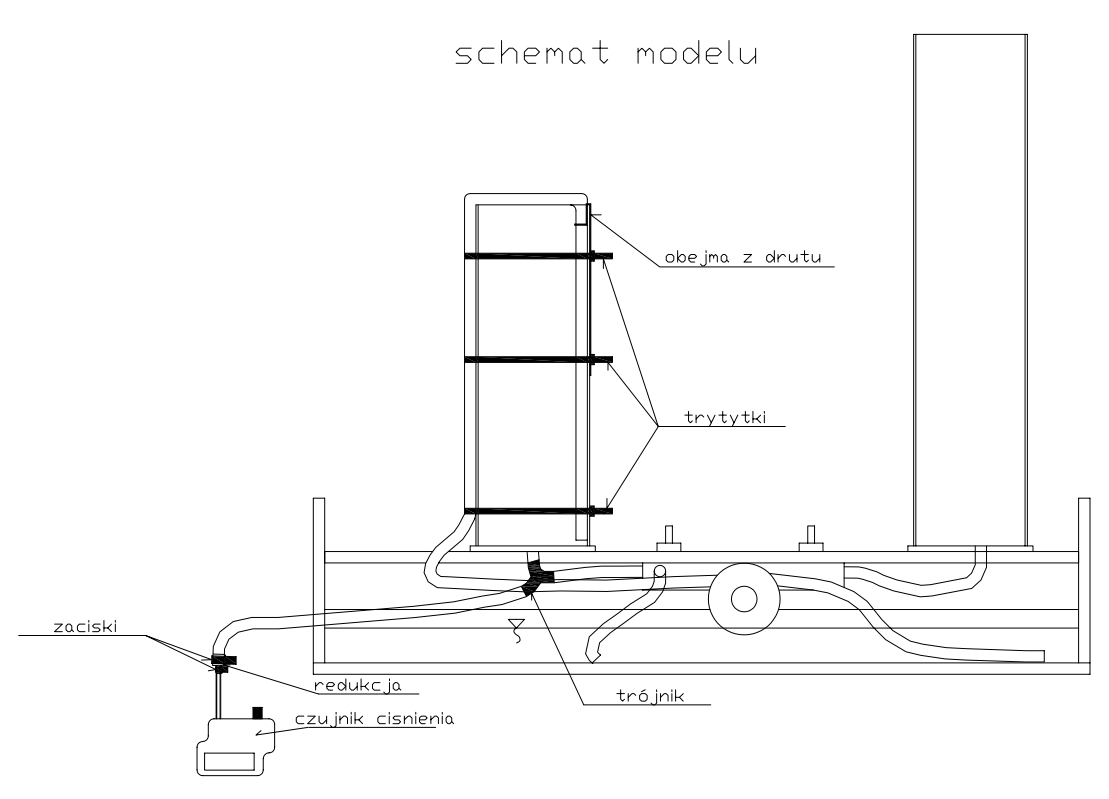
**2. Zakres pracy:**

* Zadanie 1. – skonstruowanie prototypowego modelu układu wraz z terminalem zaciskowym, umożliwiającym podłączenie układu sterującego (makieta plus elementy wykonawcze i elementy pomiarowe).
  + 1a – sposób połączenia zbiorników ze sobą za pomocą przewodów (preferowany sposób – bez konieczności wykonania dodatkowych otworów) oraz wykonanie schematu
  + 1b – połączenie zbiorników z pompą i czujnikiem
  + 1c – zebranie przewodów wyjściowych ze stanowiska w formie wiązki łączonej z PLC za pomocą zacisku
  + 1d – testowanie działania
* Zadanie 2. – wykonanie oprogramowania sterującego.
  + 2a – implementacja struktury danych TankStruct zawierającej: dane o aktualnym poziomie cieczy (z czujnika – aiLevel) oraz docelowym poziomie (informacja od użytkownika – uiLevel)), a także o poziomie użycia pompy (aoPump, uiPump), a także wysokie i niskie poziomy (Lo, LoLo, Hi, HiHi)
  + 2b – implementacja algorytmów w postaci kodu w języku ST
  + 2c – rozwiązanie problemu kalibracji (autokalibracja po uruchomieniu)
  + 2d – rozwiązanie problemu występowania szumu pomiarowego (filtracja, uśrednianie)
  + 2e – obsługa błędów (awaria czujnika, awaria pompy itp.)
* Zadanie 3. – synteza graficznego interfejsu użytkownika.
  + 3a – wykonanie panelu operatora z odpowiednimi kolorami i ikonami
  + 3b – wyświetlanie komunikatorów, czcionki
  + 3c – wykonanie menu dolnego (tryb normalny i serwisowy)
  + 3d – tryb normalny (pole numeryczne z suwakiem do wpisania poziomu cieczy, implementacja limitów)
  + 3e – tryb serwisowy (sterowanie manualne – ustawianie wydajności pompki, kierunku jej działania, diagnostyka urządzeń pomiarowych)
  + 3f – wyświetlanie poziomu cieczy i wydajności pompki w postaci paska lub wskazówki
* Zadanie 4. – wykonanie dokumentacji zawierającej opis stanowiska, schematy oraz dokumentację oprogramowania.
  + 4a – dwutygodniowe raporty
  + 4b – prezentacje na każdych zajęciach
  + 4c – pełna dokumentacja końcowa
* Zadanie 5. – wykonanie makiety 3D (np. przy użyciu SolidWorks, Blender)
* Zadanie 6. – wykonanie instrukcji użytkowania.

|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | | Week | | | | | | | | | | |
| Task | | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 |
| 1 | 1a |  |  |  |  |  |  |  |  |  |  |  |
| 1b |  |  |  |  |  |  |  |  |  |  |  |
| 1c |  |  |  |  |  |  |  |  |  |  |  |
| 1d |  |  |  |  |  |  |  |  |  |  |  |
| 2 | 2a |  |  |  |  |  |  |  |  |  |  |  |
| 2b |  |  |  |  |  |  |  |  |  |  |  |
| 2c |  |  |  |  |  |  |  |  |  |  |  |
| 2d |  |  |  |  |  |  |  |  |  |  |  |
| 2e |  |  |  |  |  |  |  |  |  |  |  |
| 3 | 3a |  |  |  |  |  |  |  |  |  |  |  |
| 3b |  |  |  |  |  |  |  |  |  |  |  |
| 3c |  |  |  |  |  |  |  |  |  |  |  |
| 3d |  |  |  |  |  |  |  |  |  |  |  |
| 3e |  |  |  |  |  |  |  |  |  |  |  |
| 3f |  |  |  |  |  |  |  |  |  |  |  |
| 4 | 4a |  |  |  |  |  |  |  |  |  |  |  |
| 4b |  |  |  |  |  |  |  |  |  |  |  |
| 4c |  |  |  |  |  |  |  |  |  |  |  |
| 5 | |  |  |  |  |  |  |  |  |  |  |  |
| 6 | |  |  |  |  |  |  |  |  |  |  |  |
|  |  | 12-18.03 | 19-25.03 | 26.03-01.04 | 02-08.04 | 09-15.04 | 16-22.04 | 23.04-29.04 | 30.04-06.05 | 07-13.05 | 14-20.05 | 21-27.05 |

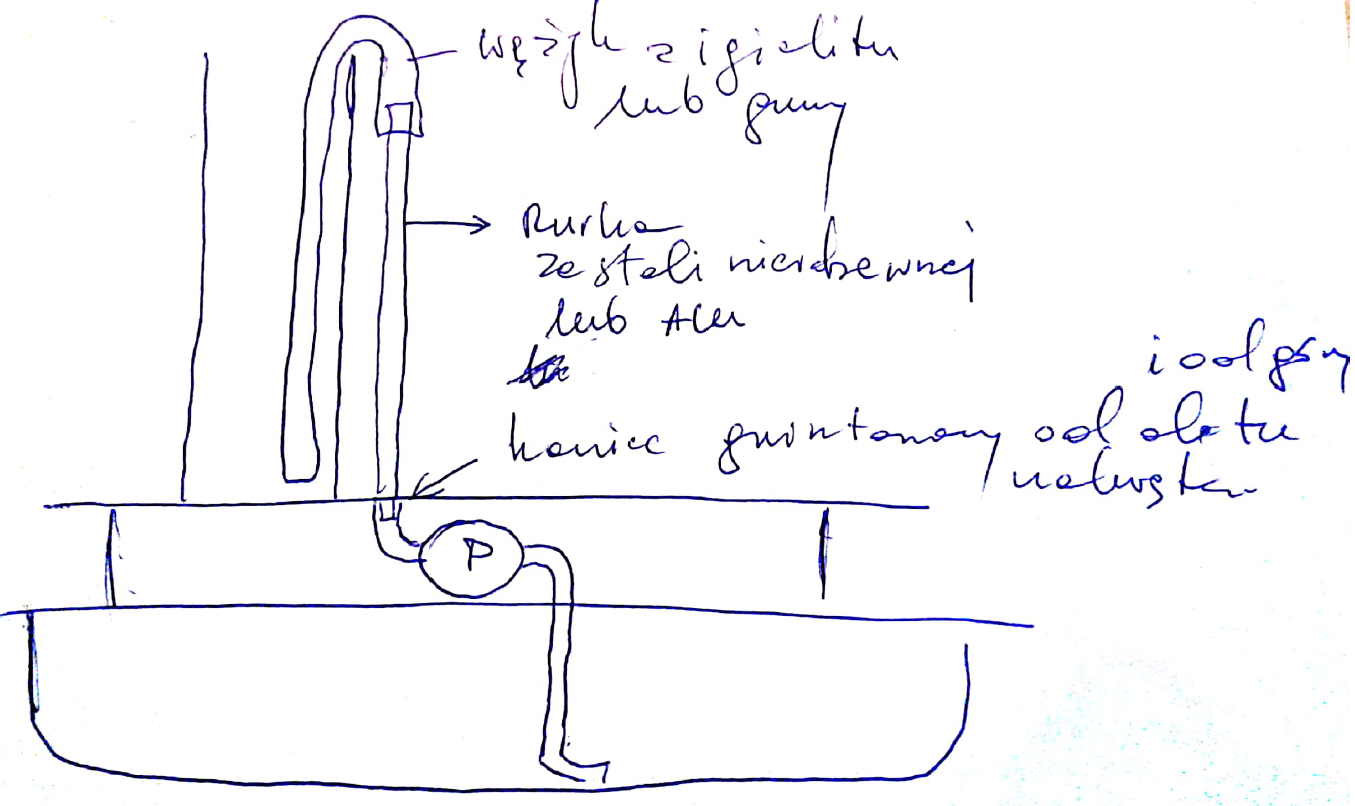
**3. Konstrukcja modelu**

3.1. Wstępny koncept:



* Pompa dwukierunkowa do spryskiwaczy, model SEV LITOVEL APO 050.01, sterowana za pomocą sygnału PWM, 24 VDC, pobierająca wodę ze zbiornika testowego, połączona do powyższych zbiorników za pomocą zacisków.
* Do sterowania pompą użyty zostanie sterownik X20MM2436 Rainer (moduł Berneckera).
* Czujnik ciśnienia – do pomiaru poziomu cieczy, model DPT250-R8-AZ-D, przekazujący na wyjście sygnał 4-20mA przekładany na minimalne i maksymalne ciśnienie (należy przeskalować je na wysokość słupa cieczy).
* Do pomiaru sygnału z czujnika użyty zostanie moduł Berneckera AI4632, przyjmujący na wejściu sygnał 4-20mA.

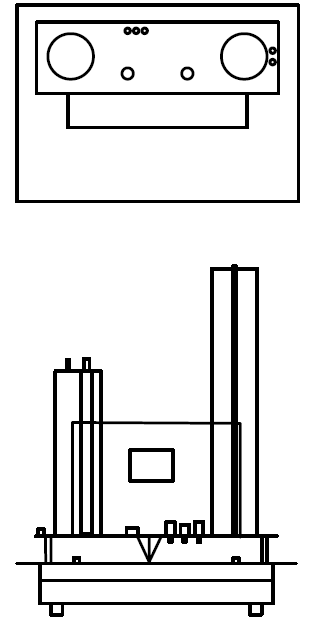
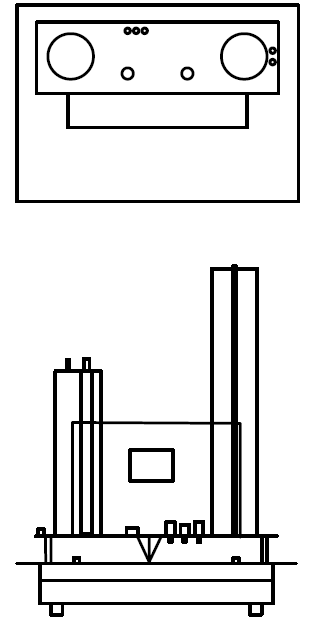
3.2. Proponowane docelowe rozwiązanie:

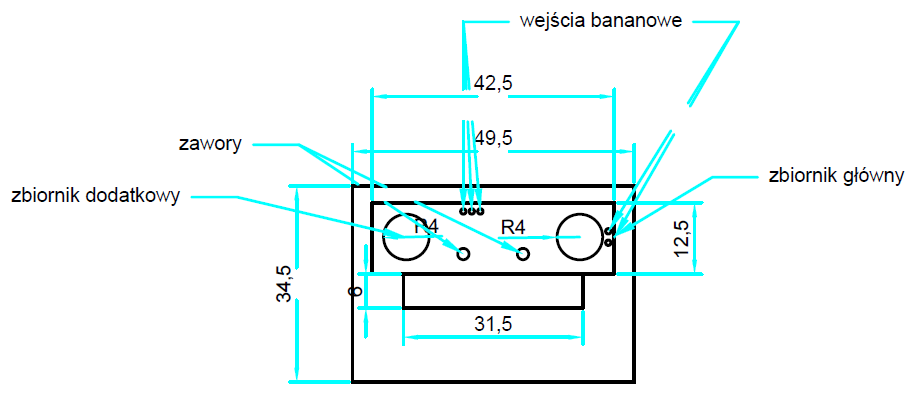


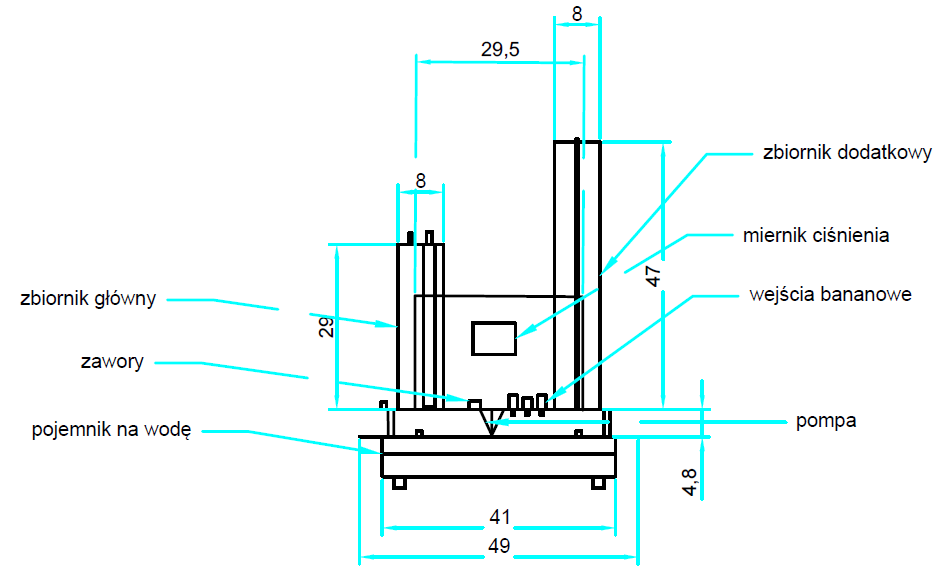
* wyprowadzenie wyjścia pompy za pomocą rurki ze stali nierdzewnej lub aluminium
* rurka byłaby po jednej stronie nagwintowana i mocowana do płyty za pomocą dwóch nakrętek kontrujących
* połączenie pompy z rurką za pomocą wężyka z igielitu lub gumy
* po drugiej stronie wężyk będzie przerzucony do zbiornika. Unikamy w ten sposób opasek, które generalnie nie wyglądają estetycznie
* podłączenie czujnika ciśnienia powietrza – podobnie jak sterowanie wodą; układ z rurką w zbiorniku doprowadzoną aż do dna. Przykładowe rozwiązanie: do zbiornika dodajemy pokrywkę z grubego plexi z wywierconymi otworami pod rurkę czujnika i rurkę zasilającą pompy.

3.3. Ostateczna konstrukcja

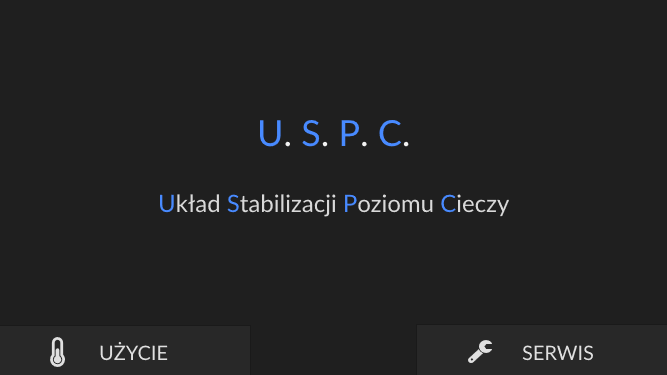
* Mocowanie dla czujnika – płytka aluminiowa, do której został zamocowany za pomocą dwóch śrub.
* Płytka została zamocowana do podstawy trzema śrubami.
* Metalowe zamknięcie komory z odpowietrznikiem, rurą dla pomiaru ciśnienia oraz rurą od pompy (stalowe rury, w które wchodzą elastyczne przewody).
* Wymieniono gniazda na "banany" oraz dodano 1 gniazdo na sygnał z czujnika; gniazda są połączone z czujnikiem.
* Czujnik połączony jest z przewodem elastycznym do pomiaru ciśnienia powietrza.
* Wprowadzono do komory przewód od pompy.







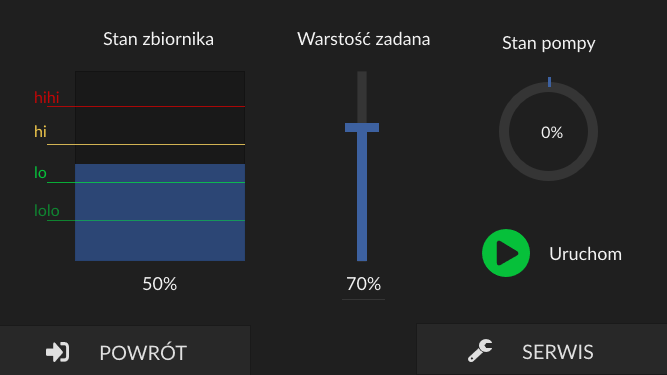
**5. GUI**

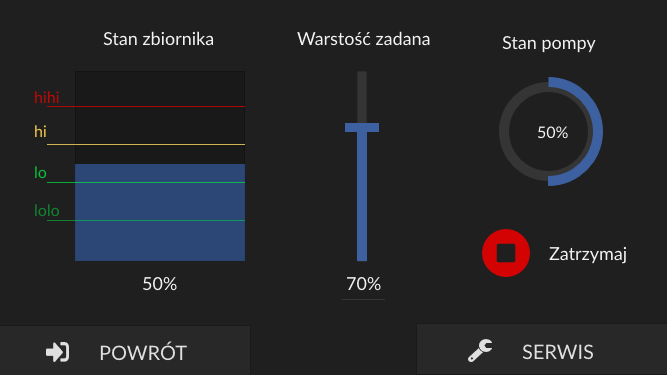


1. Użycie:

Tryb normalnego działania dla użytkownika, umożliwiający:

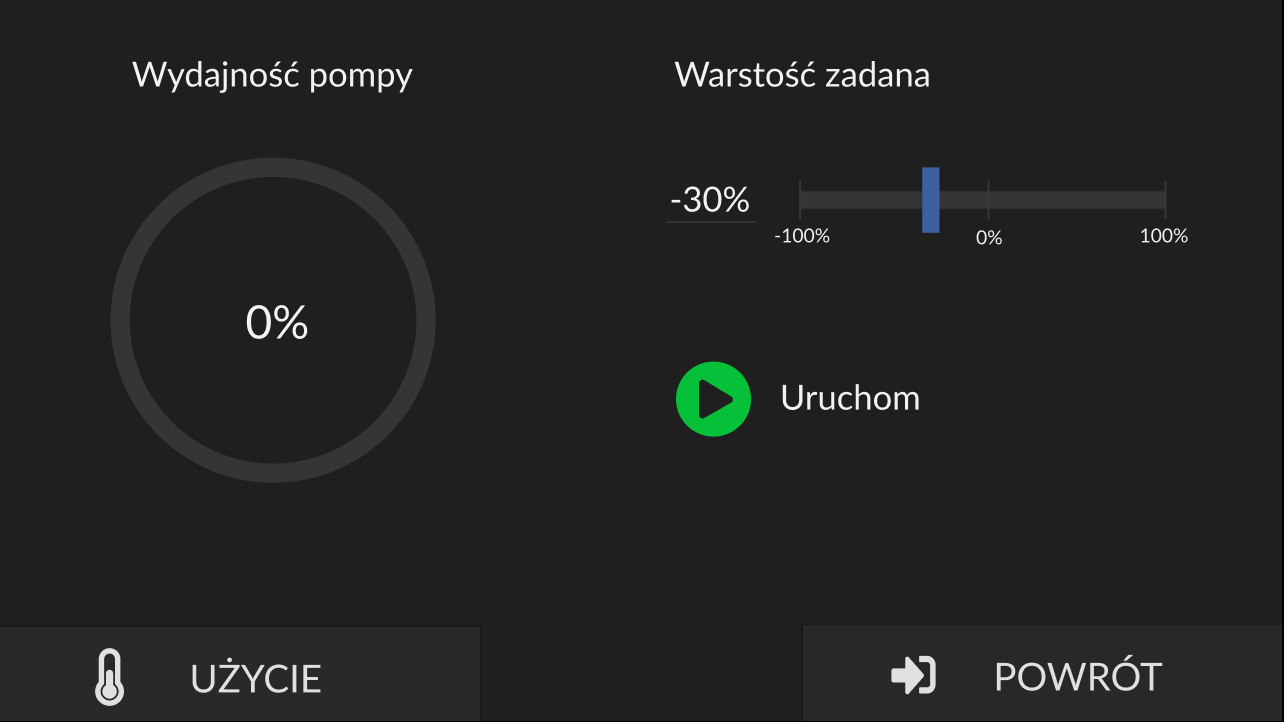
* Wyświetlenie aktualnego stanu wypełnienia zbiornika
* Ustawienie wartości zadanej – procent wypełnienia zbiornika
* Wyświetlenie aktualnej mocy pompy
* Powrót do ekranu głównego
* Możliwość wejścia do trybu serwisowego

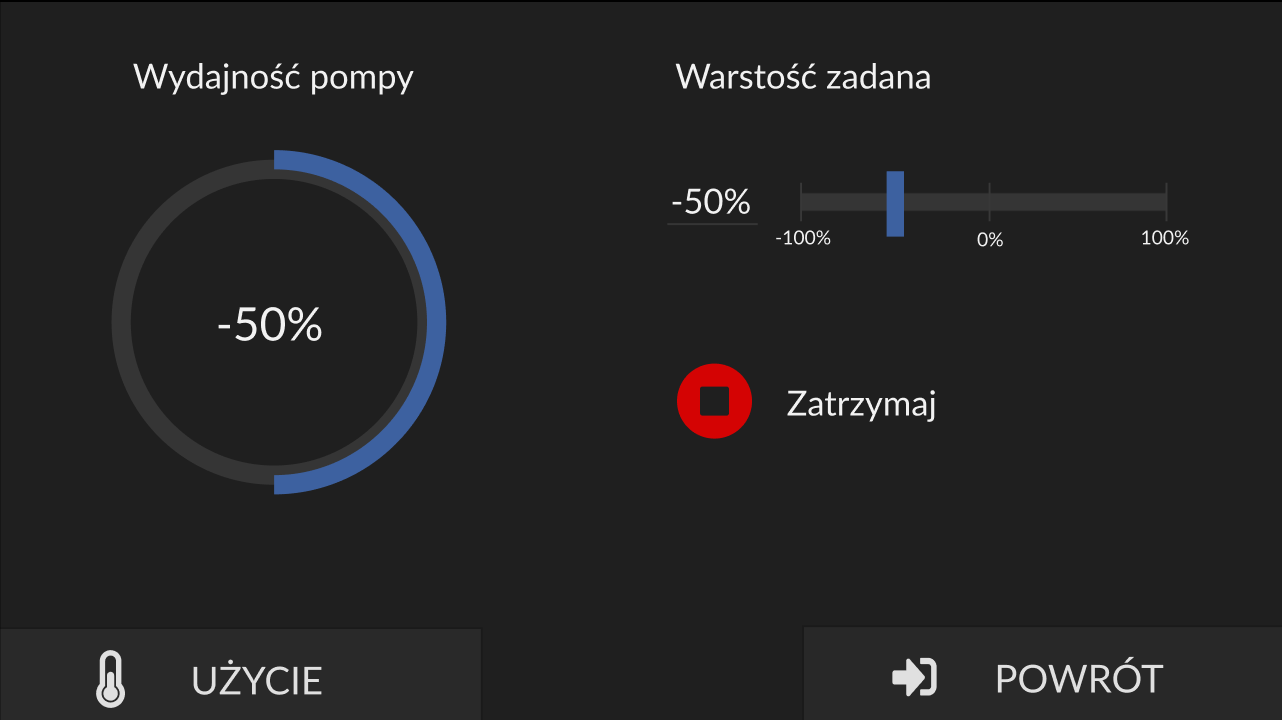




1. Serwis:

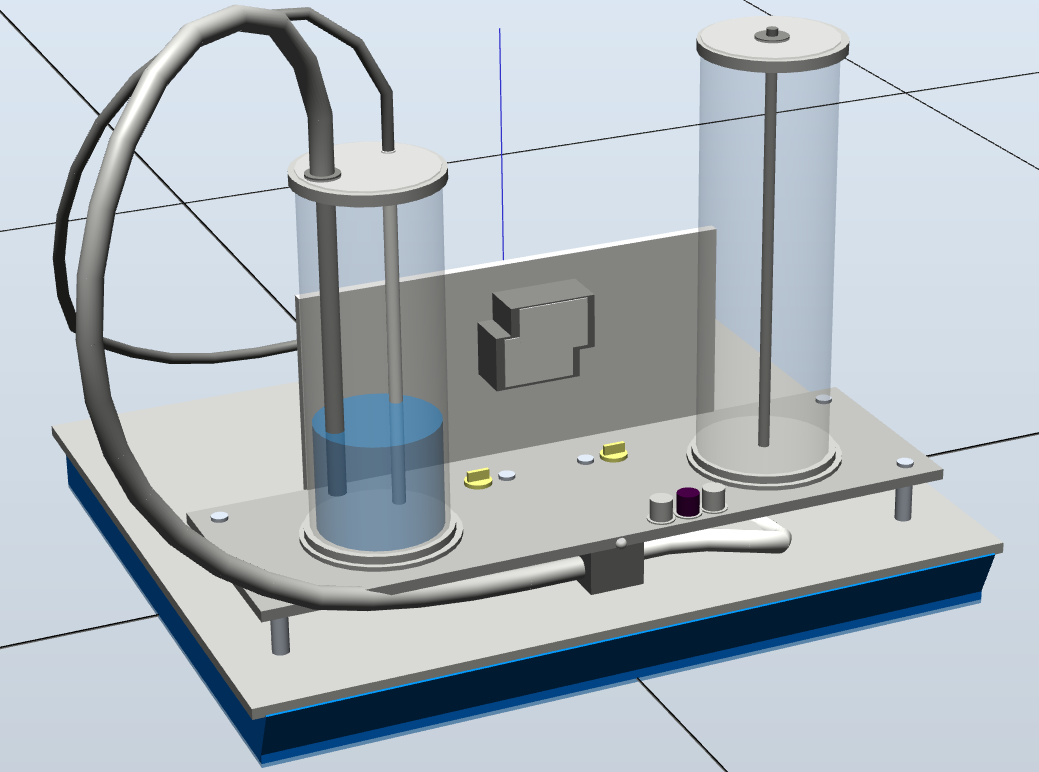
* Tryb dla serwisanta, umożliwiający dodatkowo sterowanie wydajnością pompy
* Wymagane hasło

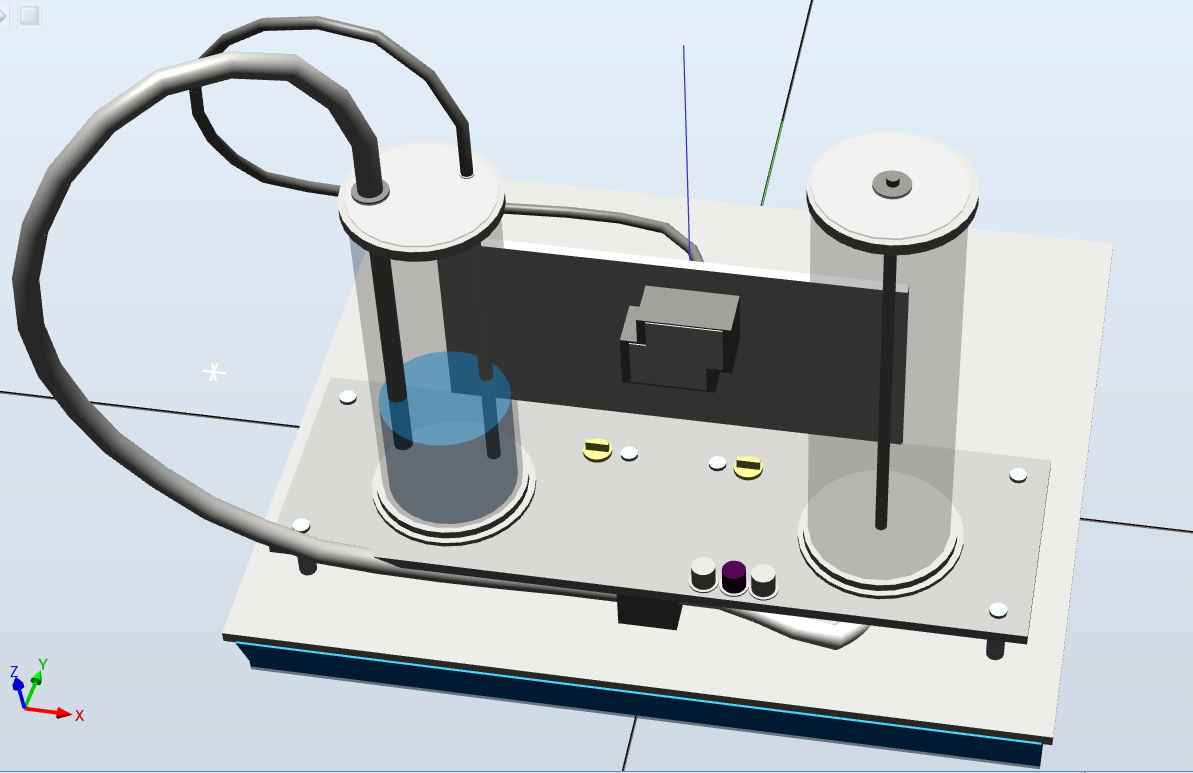




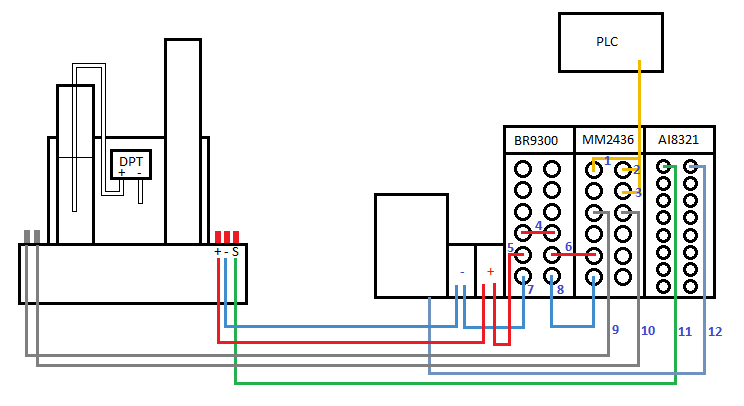
**6. Model 3D**

Model wykonano w programie ABB Robot Studio.





**7. Schemat połączeń modułów**

****

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| MODUŁ | SLOT | ZACISK | NR PRZEWODU | SYGNAŁ |
| X20 BR 9300 | 1 | X2X  X2X\  X2X⊥  +24 V X2X Link  +24 V I/O  +24 V X2X Link  +24 V I/O  GND  GND | 1  2  3  4  4  5  6  7  8 | Połączenie modułu z PLC  Połączenie modułu z PLC  Połączenie modułu z PLC  „Jumper”  „Jumper”  Zasilanie (+)  Zasilanie (+)  Zasilanie (-)  Zasilanie (-) |
| X20 MM 2436 | 2 | DI 1  DI 2  DI 3  DI 4  PWM 1+  PWM 1-  PWM 2+  PWM 2-  24 – 39 VDC  24 – 39 VDC  GND  GND | -  -  -  -  9  10  -  -  6  -  8  - | -  -  -  -  Sygnał PWM+ do pompy  Sygnał PWM- do pompy  -  -  Zasilanie (+)  -  Zasilanie (-)  - |
| X20 AI 8321 | 3 | AI + 1 I  AI - 1 I  AI + 2 I  AI - 2 I  AI + 3 I  AI - 3 I  AI + 4 I  AI - 4 I  AI + 5 I  AI - 5 I  AI + 6 I  AI - 6 I  AI + 7 I  AI - 7 I  AI + 8 I  AI - 8 I | 11  12  -  -  -  -  -  -  -  -  -  -  -  -  -  - | Sygnał z czujnika  Masa zasilania  -  -  -  -  -  -  -  -  -  -  -  -  -  - |