



WYDZIAŁ
ELEKTROTECHNIKI
I AUTOMATYKI

Imię i nazwisko studenta: Kacper Napierski
Nr albumu: 176205
Poziom kształcenia: Studia pierwszego stopnia
Forma studiów: stacjonarne
Kierunek studiów: Automatyka i Robotyka
Specjalność: Automatyka i Systemy Sterowania

PROJEKT DYPLOMOWY INŻYNIERSKI

Tytuł projektu w języku polskim: Projekt systemu sterowania szklarnią

Tytuł projektu w języku angielskim: Greenhouse control system design

Opiekun pracy: dr hab. inż. Robert Piotrowski

STRESZCZENIE

Coraz większa populacja i zmniejszające się tereny uprawne wymusza efektywniejsze metody uprawy roślin. Zaradzić temu mogą układy hydroponiczne, które dzięki rozwojowi techniki są w stanie osiągać znacznie większe oraz bardziej jednorodne plony. Jest to możliwe dzięki zaawansowanym systemom opartym na dokładnych urządzeniach pomiarowych, sterowaniu w zamkniętej pętli oraz mikrokontrolerom umożliwiającym szybką i zdalną analizę oraz archiwizację danych. Jeden układ hydroponiczny może być łatwo skalowalny, co zmniejsza koszty, a znacznie podnosi wydajność systemów. Wiele rodzajów konstrukcji może być rozbudowywanych wertykalnie, co obniża wymagania powierzchniowe względem standardowych upraw.

W ramach pracy zaprojektowano pełny system hydroponiczny. Pierwszym etapem było modelowanie dwupiętrowego układu przepływowego, który jest miejscem wzrostu sadzonek oraz odpowiednich podpór podtrzymujących przepływ cieczy. Następnym krokiem był dobór sterownika oraz jego sposobu komunikacji ze wszystkimi podsystemami elektronicznymi. W ich skład wchodzi czujniki, pompy, przekaźniki, pompki perystaltyczne, kamienie napowietrzające oraz wyświetlacz LCD. Ostatnim etapem było tworzenie algorytmów sterujących w języku Python 3 oraz konfiguracja środowiska monitorującego - nadzorczego.

Słowa kluczowe: hydroponika, Mycodo, Raspberry Pi, uprawa roślin, układ przepływowy, NFT

Dziedzina nauki i techniki, zgodne z wymogami OECD: Nauki inżynieryjne i techniczne; Elektrotechnika, elektronika, inżynieria informatyczna

ABSTRACT

Increasing population and diminishing farmlands force developing more effective plant cultivation methods. The answers are hydroponics systems, which are able to result in bigger and more homogenous crops thanks to technological progress. It is possible with advanced technology based on precise measuring devices, feedback control systems and fast-data-processing microcontrollers. Advantage to using hydroponics systems is scalability, which contributes to reducing costs while increasing system effectiveness. Many variants can be expanded vertically. It decreases land requirement in comparison to standard plant cultivation methods.

Full hydroponic system was designed within scope of this study. The first stage was modeling of two-level flow structure being the place of growing plants and it's flow-keeping supports. Then the controller with it's communication method with all electrical subsystems was selected. Those subsystems contain sensors, submerged pumps, relays, peristaltic pumps, air-stones and LCD screen. At the end control algorithm was created using Python 3 code. Furthermore configured was open-source, supervisory control software similar to the ones used in the commercial industry, but suited for microcontroller solutions.

Keywords: hydroponics, Mycodo, Raspberry Pi, plant cultivation, flow system, NFT

SPIS TREŚCI

WYKAZ NAJWAŻNIEJSZYCH SKRÓTÓW.....	6
1. WSTĘP.....	7
1.1. Informacje ogólne.....	7
1.2. Cel i zakres pracy.....	9
2. BUDOWA SYSTEMU NFT.....	10
3. CZĘŚĆ ELEKTRONICZNA.....	17
3.1. Elektronika centralna.....	18
3.1.1. Wybór protokołu komunikacyjnego.....	19
3.1.2. Schemat łączenia podsystemów.....	27
3.2. System zasilania.....	28
3.3. Podsystem pomiarowy.....	31
3.3.1. Składanie układu.....	31
3.3.2. Schemat połączeń.....	36
3.4. Podsystem dowartościowywania wody.....	37
3.5. Podsystem kondycjonowania i pompowania wody.....	42
4. SYSTEM STEROWANIA.....	47
4.1. Konfiguracja urządzenia.....	47
4.2. Mycodo.....	49
4.3. Konfiguracja platformy.....	51
5. PODSUMOWANIE.....	59
5.1. Podsumowanie projektu.....	59
5.2. Możliwości rozwoju projektu.....	59
WYKAZ LITERATURY.....	60
WYKAZ RYSUNKÓW.....	62
WYKAZ TABEL.....	64



DODATEK A: Klasy wygenerowane przez program Mycodo.....	65
DODATEK B: Rysunki techniczne.....	67
DODATEK C: Pliki modeli 3D.....	75

WYKAZ NAJWAŻNIEJSZYCH OZNACZEŃ I SKRÓTÓW

- NFT - ang. *Nutrient Film Technique* - technika warstwy odżywczej
- CAD - ang. *Computer Assisted Design* - projektowanie wspomagane komputerowo
- PID - ang. *Proportional-Integral-Derivative* - proporcjonalno-całkująco-różniczkujący
- PC - ang. *Personal Computer* - komputer osobisty
- UART - ang. *Universal Asynchronous Receiver-Transmitter* - uniwersalny asynchroniczny odbiornik-nadajnik
- I2C - ang. *Inter-Integrated Circuit* - pośrednik pomiędzy układami scalonymi
- SPI - ang. *Serial Peripheral Interface* - szeregowy interfejs urządzeń peryferyjnych
- 3D - ang. *Three dimensional* - trójwymiarowy
- RJ45 - ang. *Registered Jack - type 45* - rejestrowane złącze typu 45
- GPIO - ang. *General Purpose Input/Output* - wejście/wyjście ogólnego użytku

1. WSTĘP

1.1. Informacje ogólne

Hydroponika- niezależnienie od ziemi i piętrowe uprawy

Uprawa roślin towarzyszy ludziom od tysięcy lat. Wytłacza początek osiadłego trybu życia i formowania pierwszych osad. W dzisiejszych czasach to miasta stanowią znaczną część terytorium państw, a rozbudowana infrastruktura ogranicza miejsce dla możliwych pól uprawnych. Ciągłe rosnąca populacja wymaga stałego rozwoju, nowych mieszkań oraz coraz więcej terenów zagospodarowanych pod budowy. Wymusza to stosowanie coraz efektywniejszych form upraw, które w pełni wykorzystają daną powierzchnię.

Jedną z nowoczesnych technologii plantacji jest hydroponika. Polega na bezglebowej hodowli roślin. Głównym założeniem tej koncepcji jest oparcie upraw na systemach wodnych dowartościowanych pożywkami roztworów soli mineralnych, które zapewniają wszystkie najważniejsze substancje odżywcze i umożliwiają wzrost. Sam pomysł na sadzenie roślin w wodzie jest dość stary, ponieważ pierwsze zapisy na ten temat sięgają 1627 r. [26], ale większe sukcesy pojawiły się dopiero w latach trzydziestych dwudziestego wieku. Do dzisiejszego dnia ta dziedzina hodowli bardzo się rozwinęła i spopularyzowała za sprawą ogromnego skoku technologicznego w przeciągu ostatnich 50 lat. Skutkuje to powstaniem wielu rodzajów systemów hydroponicznych (patrz rys 1.1).



Rys.1.1. System hydroponiczny (Autor: Ryan Somma) [27]

Różne formy hydroponiki rozwiązują inne problemy. Niektóre z nich pozwalają zaoszczędzić wiele miejsca i zwiększyć znacząco zbiory przy zachowaniu podobnych kosztów dzięki świetnej skalowalności oraz wykorzystaniu przestrzeni wertykalnej. Metody te pozwalają na uprawy w przestrzeniach do tego nieprzystosowanych jak miasta, gdzie ściany i dachy budynków mogą zostać wykorzystane do upraw. Ludność zamieszkująca kamieniste obszary lub kraje, gdzie nie ma wiele terenów uprawnych może bardziej uniezależnić się od dostaw świeżych warzyw zza granicy.

Najpopularniejszymi rozwiązaniami są:

- DWC (ang. *Deep Water Culture*) - rośliny rosną nad zbiornikiem pełnym wody, do którego sięgają ich korzenie. Woda jest odżywiana i kondycjonowana. W pewnych odstępach czasu zależnych od warunków środowiskowych należy wymienić całą ciecz znajdującą się w systemie.
- NFT (ang. *Nutrient film technique*) - uprawy umieszczone są w specjalnych rurach przepływowych przez które przepływa cienka warstwa wody z minerałami zapewniającymi wszystkie składniki potrzebne do przeprowadzenia fotosyntezy. Ciecz jest odpowiednio przygotowywana w zewnętrznym zbiorniku, który jest uzupełniany, bez konieczności wymiany całej wody poza serwisowaniem.
- Aeroponika - rośliny znajdują się na podniesieniu i ich korzenie nie stykają się bezpośrednio z wodą. Odpowiednio przygotowana woda jest pompowana poniżej roślin i tam rozpylana jako mgielka wodna.
- Ebb and flow (ang. *Odptyw i przyptyw*) - rośliny znajdują się w medium do wzrostu (np. granulowana glina), która utrzymuje wilgotność na dłużej. Pojemnik z roślinami znajduje się nad zbiornikiem z cieczą, która jest regularnie wpompowywana na poziom roślin zalewając je wodą. Odptyw jest realizowany przez swobodny wypływ z powrotem do zbiornika. Przypomina to klasyczny sposób uprawy, gdzie ziemia jest podlewana z pewną częstotliwością.

Każdy z wymienionych sposobów uprawy roślin ma swoje niewątpliwe zalety i możliwości rozwoju. Do rozważenia jest skomplikowanie projektu, jakie parametry należy kontrolować oraz w jakich warunkach system ma operować. Metody warto również dobrać do uprawianej rośliny. Wszystkie z nich zapewniają dobre plony praktycznie każdego gatunku powszechnie uprawianych hodowli, ale jeśli plantacji zależy na najoptymalniejszych wynikach, warto zbadać metody przed ich użyciem.

Odpowiednio sterowany układ hydroponiczny niezależnie od rodzaju zapewnia jednorodność zbiorów oraz zmniejsza straty spowodowane chorobami lub szkodnikami.



1.2. Cel i zakres pracy

Projekt układu hydroponicznego zakłada stworzenie dwupiętrowego systemu do uprawy roślin w technologii NFT o długości czterech metrów i szerokości siedemdziesięciu centymetrów. Jest to układ mieszczący osiemdziesiąt sadzonek przy wykorzystaniu trzech metrów kwadratowych, co jest największą zaletą tego typu upraw. Obiekt musi zmieścić się w istniejącej szklarni ograniczającej jego maksymalne rozmiary, przy jednoczesnym zachowaniu miejsca do konserwacji. Układ projektowany jest w taki sposób, aby zmieściły się w obiekcie dwa identyczne układy hydroponiczne rozdzielone ścieżką.

Układ zakłada ciągłą pracę pompy zanurzonej wewnątrz zbiornika w celu zapewnienia odpowiednich składników odżywczych roślinom. Ciągły przepływ wspomaga mieszanie się wszystkich substancji w wodzie. Woda jako główny ośrodek odżywczy musi być kondycjonowana na bieżąco. Ze względu na znaczenie działania pompy należy opracować system zwiększający niezawodność dostarczania wody do sadzonek. Warto zabezpieczyć urządzenia na wypadek uszkodzenia spowodowanego suchobiegami.

System dowartościowywania musi podtrzymywać dostarczanie minerałów, utrzymywać parametry pH oraz elektroprowadnictwa w zakresach bezpiecznych dla wzrostu roślin.

Powinien pojawić się również prosty system monitorujący - nadzorczy, który będzie operował na dwóch warstwach: przedniej (ang. *frontend*) i tylnej (ang. *backend*). Przednia pozwala na przekazywanie informacji o systemie i jego działaniu użytkownikowi. Tylna natomiast będzie odpowiedzialna za sterowanie wszystkimi parametrami systemu hydroponicznego w zamkniętej pętli sterowania. W tym celu należy oczylnikować odpowiednio cały układ, aby sterowanie mogło zostać wykonane prawidłowo.

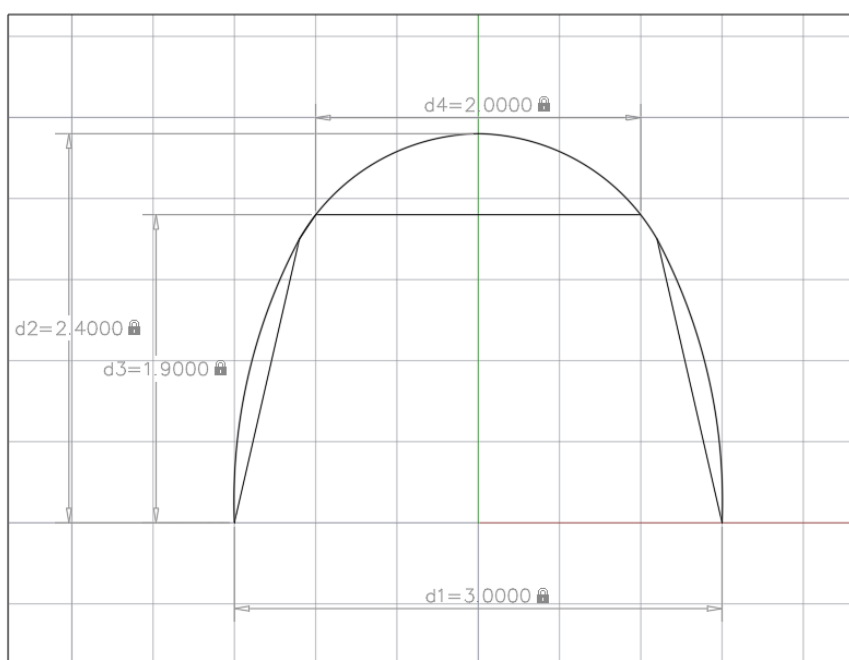
Proces projektowania został podzielony na trzy główne części:

- Mechaniczna - zaprojektowanie układu przepływowego oraz jej podpór.
- Elektryczna - sporządzenie połączeń układów elektrycznych, wybór sposobu komunikacji z kontrolerem, dobór urządzeń pomiarowych i wykonawczych.
- Programowa - stworzenie systemu sterowania oraz konfiguracja systemu monitorującego.

2. BUDOWA SYSTEMU NFT

Rozdział ten zawiera omówienie projektu systemu hydroponicznego oraz konstrukcję mechaniczną wraz z doбором elementów wymaganych do poprawnego funkcjonowania całego układu.

Proces projektowania rozpoczął się od zapoznania się z ograniczeniami fizycznymi wynikającymi z wymiarów szklarni oraz jej konstrukcji wewnętrznej, tj. stelaży, ramy i podpór. Wymiary szklarni z zewnątrz: długość = 6 m, szerokość = 3 m, wysokość = 2,4 m. Uwzględniając wcześniej wymienione obostrzenia oraz kształt obiektu zwięzający się w górnych jego częściach, należy dobrać konstrukcję oraz rozmiary układu, aby zapewniły duże wykorzystanie powierzchni roboczej szklarni. Ważnym aspektem jest również komfort użytkowania, utrzymywania i serwisowania, który wymaga pozostawienia odpowiednio dużo miejsca na swobodny ruch i dostęp do wszystkich kluczowych elementów konstrukcji. Poglądowy przekrój pionowy szklarni wzdłuż szerokości pokazano na rys. 2.1.



Rys. 2.1. Poglądowy przekrój pionowy szklarni wzdłuż szerokości (rysunek wykonany w środowisku Autodesk AutoCAD 2019)

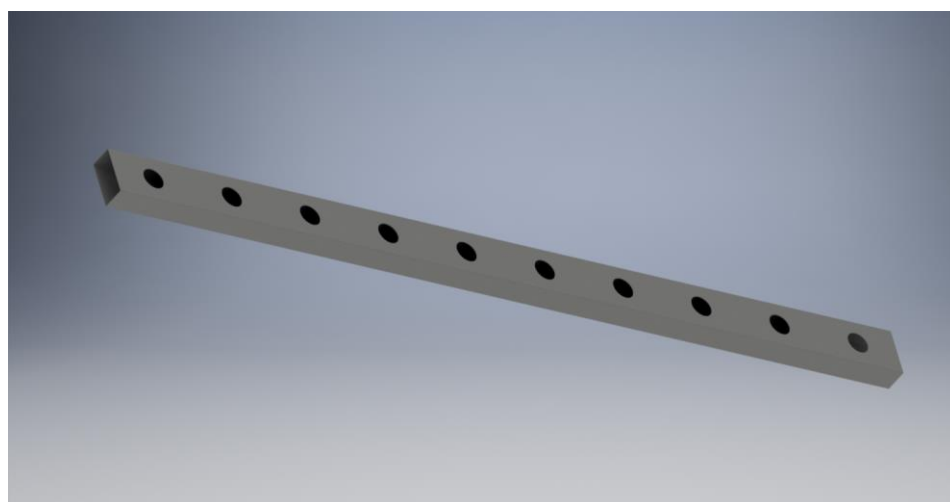
Znając ograniczenia, należy na ich podstawie dobrać strukturę układu, aby zapewnić komfort użytkownika, co pozwala na kontrolowanie i swobodne poruszanie się wewnątrz szklarni. Zachowanie ścieżki na środku szklarni pozwala na postawienie systemów hydroponicznych po obu stronach, dalej zapewniając do nich dostęp. Jeden z najważniejszych celów stosowania tego typu rozwiązań uprawy to maksymalizacja liczby sadzonek na ograniczonym terenie. Jest to związane z wykorzystaniem osi Z jako osi sadzenia.



Aby zapewnić odpowiednie warunki rozwoju roślin oraz możliwie najlepiej wykorzystać wertykalność rozwiązania NFT skonstruowano dwupoziomowy układ przepływowy. Na każdym z pięter znajdują się dwa rzędy sadzonek z koszykami do sadzenia co 20 cm, aby zapewnić miejsce dla systemów korzeniowych. Długość systemu została ograniczona do 4 m, ponieważ musi pozostać wystarczająca ilość miejsca na zbiornik jak i resztę obowiązkowych systemów znajdujących się wewnątrz szklarni. Jednym z kluczowych aspektów konstrukcji jest odpowiednie pochylenie pięter, aby stymulować przepływ oraz zapobiegać stagnacji i psuciu się wody wewnątrz rur, co mogłoby skutkować gniciem korzeni oraz chorobami sadzonek.

Na temat doboru kąta w systemie hydroponicznym prowadzone są badania. Wykazują one, że w zależności od rodzaju roślin najlepsze wyniki wzrostu odnotowuje się dla różnych kątów [12]. Niemniej jednak jest pewien zakres kątów, który zapewnia pewność działania niezależnie od gatunku sadzonki. Zalecany jest przedział pochyłu 1-4% co odpowiada kątom $0,6^{\circ}$ - $2,29^{\circ}$. Ze względu na długość rzędów wybrany został kąt $1,15^{\circ}$ co odpowiada nachyleniu 2%, ponieważ jest to wartość pojawiająca się w wielu publikacjach [10,11,12,22] oraz zapewnia poprawne wyniki rozwoju. Poza tym przy tak niskiej wartości pochylenia eliminujemy ryzyko zbyt dużej prędkości wody, która mogłoby skutkować nieodpowiednim poziomem przyswajania wartości odżywczych i nawodnieniem, a co z tym idzie obumieraniem roślin.

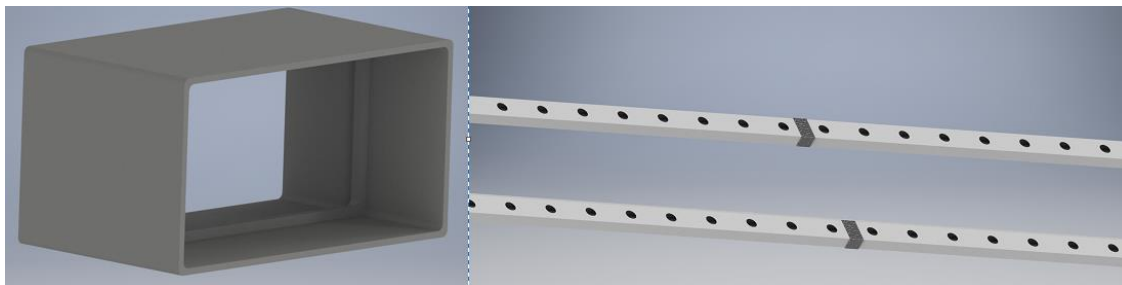
System przepływu wody oparto na prostokątnych elementach (patrz rys.2.2.) o wymiarach 110x55mm, które mają wszystkie elementy umożliwiające złożenie wymaganego układu zapewniając prostotę konstrukcji oraz łatwość dopasowywanie długości poszczególnych elementów. W każdej z podłużnych rur należy wycinać otwory służące do umieszczenia w nich sadzonek w koszykach wydrukowanych na drukarce 3D. Korzenie powinny być tylko częściowo zanurzone w wodzie, aby nie doprowadzić do ich utopienia.



Rys. 2.2. Rura do systemu przepływowego (render wykonany w środowisku Autodesk Inventor Pro 2018)



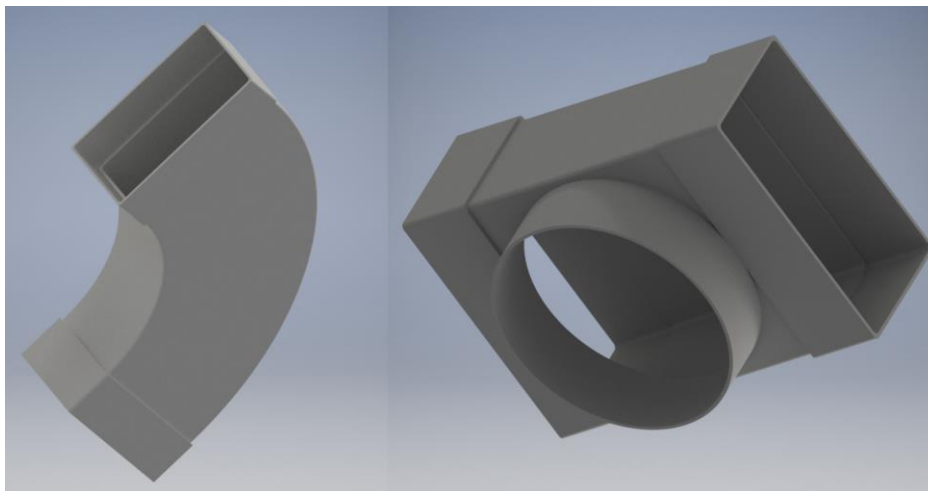
Aby maksymalnie wykorzystać dostępną powierzchnię, należy złożyć cztery 2 m rury przepływowe, aby uzyskać jak najdłuższe zbcze do sadzenia roślin jednocześnie mieszczące się wewnątrz szklarni. Złożenie powinno skutkować uzyskaniem dwóch 4 m szczelnych przewodów. (patrz rys. 2.3.)



Rys.2.3. Łącznik rur oraz montaż (rendery wykonane w środowisku Autodesk Inventor Pro 2018)

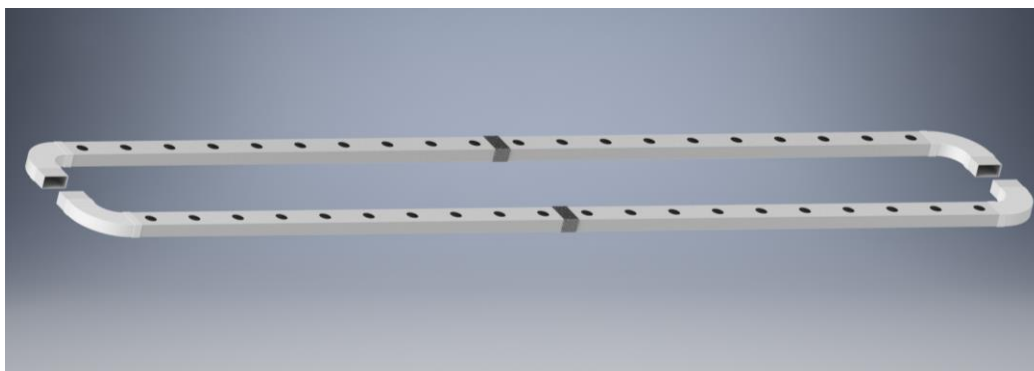
Warto zwrócić uwagę na spasowanie elementów w celu zapewnienia szczelności. W przypadku przecieków trzeba użyć silikonu, gumy w spray lub w przypadku małego wycieku powinna wystarczyć taśma izolacyjna.

W celu dalszego montażu piętra systemu należałoby połączyć ze sobą obie rury. W tym celu wykorzystano kolano 90° oraz ucięty kawałek rury w celu regulacji szerokością (patrz rys.2.4.-2.5.). Piętro wyższe powinno być węższe, aby zapewnić roślinom rosnącym na niższych piętrach wystarczająco miejsca na wzrost w przypadku długiej łodygi.



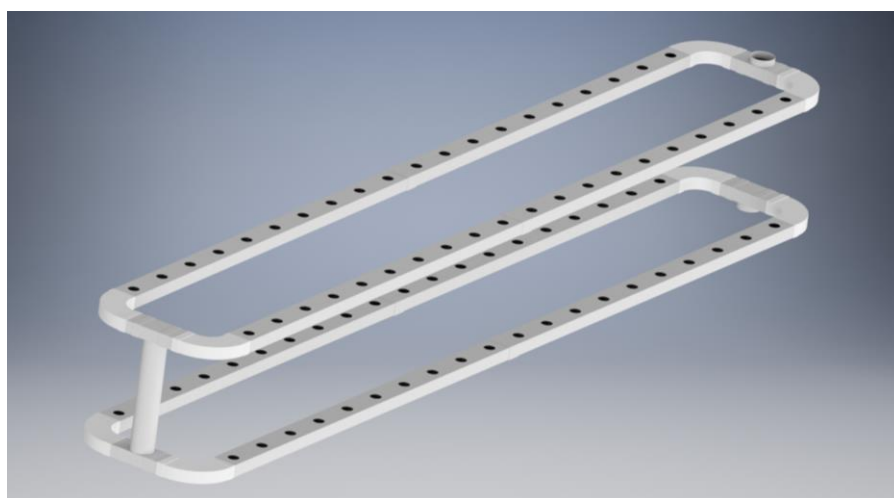
Rys. 2.4. Kolano 90° oraz trójnik (rendery wykonane w środowisku Autodesk Inventor Pro 2018)





Rys. 2.5. Złożenie obu stron piętra (render wykonany w środowisku Autodesk Inventor Pro 2018)

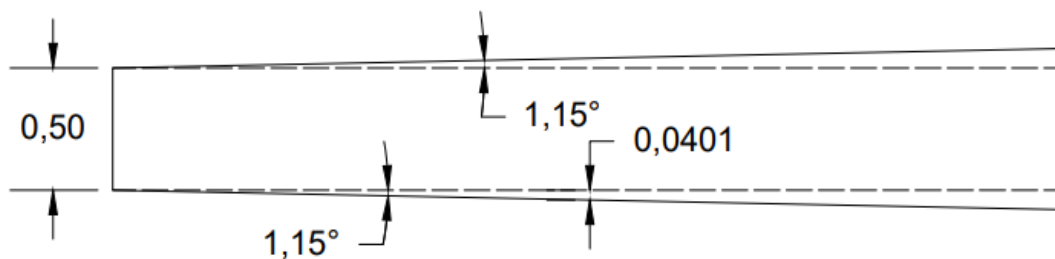
W celu zapewnienia piętrowości wykorzystano trójkąt rozdzielający prostokąt-okrąg oraz rurę o średnicy 100 mm i długości 50 cm. Długość dobrano tak, aby zapewnić wystarczająco miejsca na wzrost roślin, ale jednocześnie nie przyspieszyć znacząco prędkości wody, ponieważ mogłoby to spowolnić rozwój roślin w dolnych piętrach układu. Przy zbyt dużych prędkościach mogłoby to również doprowadzić do wymywania korzeni roślin. Układ przepływowy pokazano na rys 2.6.



Rys. 2.6. Układu przepływowy bez podpór (render wykonany w środowisku Autodesk Inventor Pro 2018)

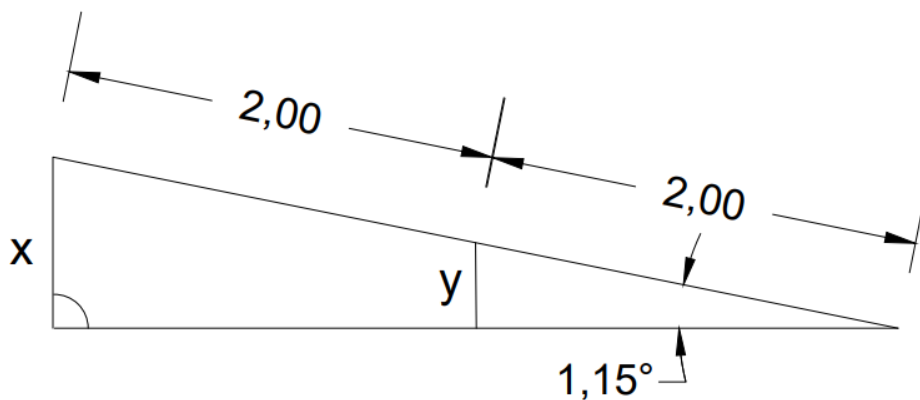
Mając zbudowany układ przepływowy, należałoby dobrać odpowiednie podpory zapewniające stabilność i utrzymanie odpowiedniego kąta na obu poziomach jednocześnie. W tym celu pomoże przedstawiony wykonano ideowy rysunek pokazujący, jak rozchodzą się rury układu przepływowego (patrz rys. 2.7.).





Rys. 2.7. Ideowy rzut boczny układu przepływowego bez podpór (rysunek wykonany w środowisku Autodesk AutoCAD 2019)

Przy długości 4 m składającej się z dwóch 2 m rur z łącznikiem należałoby zastosować co najmniej 3 podpory na wysokości połączeń. Ze względu na przesuwanie się rur podpory będą różniły się od siebie w zależności w którym miejscu będą montowane. Należy wyliczyć różnicę między nimi przed zaprojektowaniem ich w środowisku Autodesk Inventor (patrz rys.2.8.).



Rys. 2.8. Reprezentacja odchyłu rur poziomu (rysunek wykonany w środowisku Autodesk AutoCAD 2019)

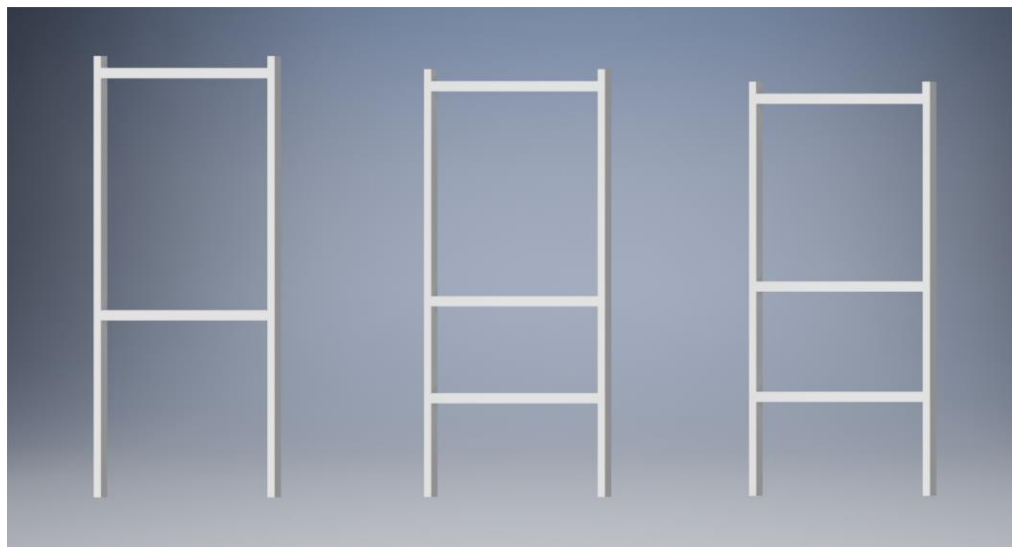
Widoczne odchylenie rur poziomu (patrzy rys.2.8.) potraktowano jako przeciwprostokątna trójkąta w celu wyliczenia zmiany wysokości podpór w wybranych miejscach względem najniższego z punktów. W pierwszej kolejności należy wyliczyć długość różnicy wysokości oznaczonej na rys.2.8. jako x korzystając z podstawowych funkcji trygonometrycznych i znanej długości rury.

$$x = 4 \cdot \sin(1,15) \approx 0,080279 \approx 0,08 [m] \quad (2.1)$$

Po otrzymaniu wartości x i y można obliczyć korzystając z twierdzenia Talesa.

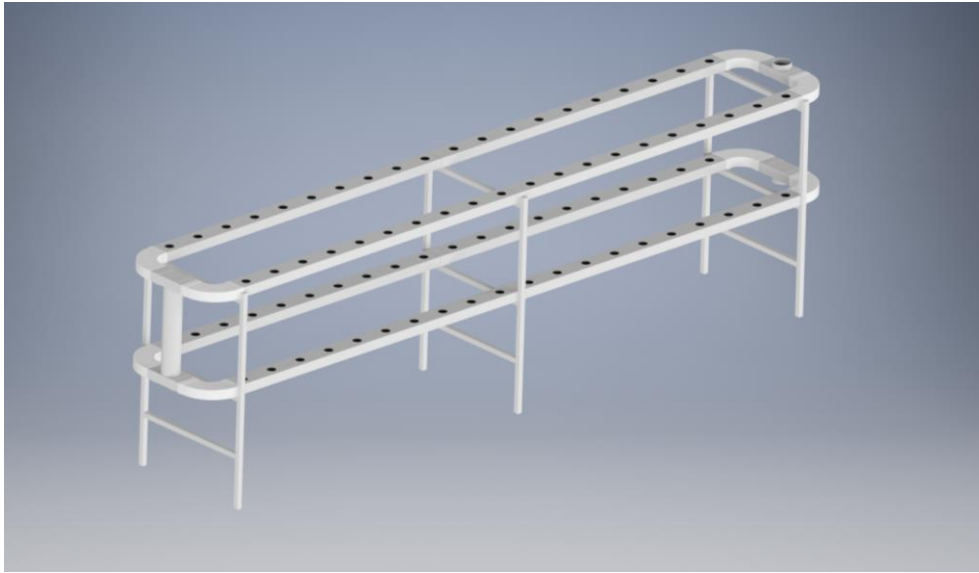
$$y = \frac{x}{4} \cdot 2 \approx 0,04 [m] \quad (2.2)$$

Wysokości x i y otrzymane w równaniach (2.1) i (2.2) oznaczają różnicę wysokości uzyskaną przez pochylenie rur. Należy brać je pod uwagę, aby zapewnić odpowiednie warunki i spełnić założenia. Obliczenia te pozwalają na modelowanie układu podpór, które podtrzymują całą przepływową część konstrukcji. Docelowo planowane są 3 niezależne ramy różniące się wysokością, aby były w stanie zapewnić na stałe odpowiedni kąt oraz utrzymać oba piętra jednocześnie (patrz rys. 2.9.). Ramy mogą zostać wkopane w ziemię na odpowiednią głębokość w celu regulacji wysokości. Trzeba pamiętać o zapasie wysokości, w którym zmieści się zbiornik wodny. Zapewnia on recykling wody w układzie i jest główną siłą techniki NFT.



Rys. 2.9. Elementy stelażu utrzymującego układ przepływowy (render wykonany w środowisku Autodesk Inventor Pro 2018)

Dzięki lekkiej konstrukcji stelaż zapewnia jednocześnie małe zużycie materiału i wystarczającą sztywność względem konstrukcji podtrzymywanej. Jedna z podpór nie posiada usztywniającej belki poprzecznej, aby zapewnić wystarczająco miejsca na wsunięcie pod spód zbiornika. Podpory należy połączyć z resztą układu najlepiej w miejscach styku rur (patrz rys. 2.10.) w celu zapewnienia największej stabilności konstrukcji, bo są to punkty styku. Zmniejszając momenty grawitacyjne działające na konstrukcję, teoretycznie poprawi się również jej szczelność (brak napięć na przejściach elementów).

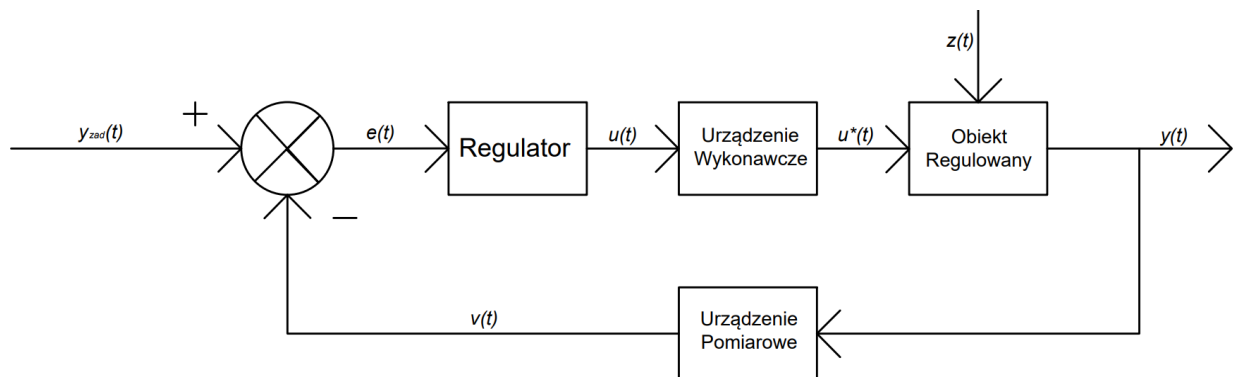


Rys. 2.10. Gotowa konstrukcja przepływowa (render wykonany w środowisku Autodesk Inventor Pro 2018)

Na otrzymanym modelu 3D można zdiagnozować większość problemów konstrukcyjnych przed zbudowaniem fizycznego obiektu, co jest dużą zaletą środowisk CAD (ang. *Computer Assisted Design*) i jednym z głównych powodów ich popularności. Wizualizacja konstrukcji pozwala również użyć jej jako podstawa przy symulacjach w innych środowiskach programistycznych.

3. CZĘŚĆ ELEKTRONICZNA

Układ przepływowy jest obiektem o charakterze liniowym. Świadczy o tym ilość wypływającej wody w zamkniętym obiegu. Wraca ona do zbiornika pomniejszona o proporcjonalną ilość zależną od zapotrzebowania roślin. Faktem jest, że w różnych okresach roku zapotrzebowanie zmienia się, ale tempo przemian jest na tyle wolne, że odchylenie powinno mieścić się w zapasie zadanej ilości wody. Klasyczny układ regulacji (patrz rys.3.1.) składa się z czterech podstawowych elementów: regulatora, urządzenia wykonawczego, obiektu sterowanego i urządzenia pomiarowego. Należy stworzyć odpowiednie podsystemy realizujące poszczególne funkcje.



Rys.3.1. Klasyczny układ regulacji - schemat blokowy (rysunek wykonany w środowisku Autodesk AutoCAD 2019)

gdzie:

$y_{zad}(t)$ - sygnał zadany,

$y(t)$ - sygnał regulowany,

$e(t)$ - sygnał uchybu regulacji,

$u(t)$ - sygnał regulujący,

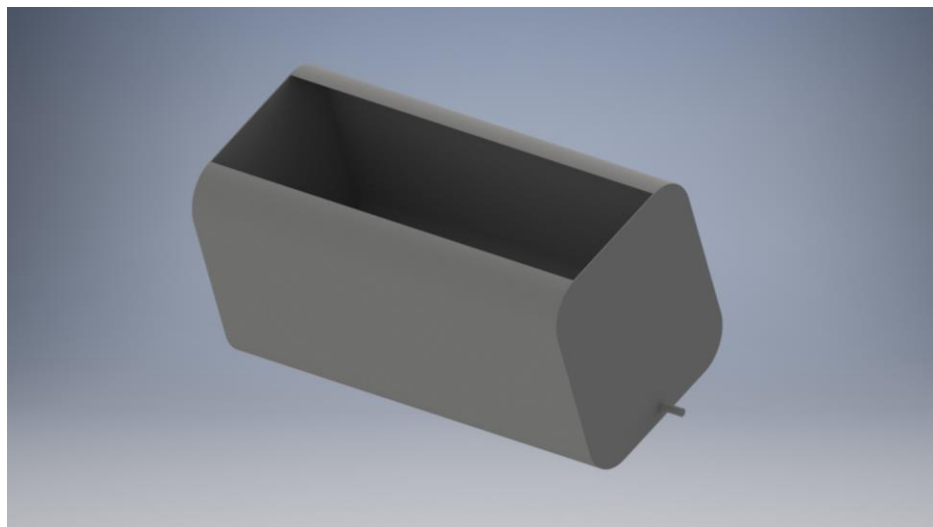
$u^*(t)$ - sygnał regulujący nastawiający,

$z(t)$ - sygnał zakłócający,

$v(t)$ - sygnał regulowany mierzony.

Każdy ze wskaźników układu otrzyma własny układ regulacji, co przełoży się na łatwość doboru sposobu sterowania oraz brak ograniczeń związanych z wpływem pomiędzy parametrami. Wszystkie pętle będą realizowane przez jeden fizyczny regulator posiadający wiele wejść i wyjść, co uprości schemat elektryczny układu oraz pozwoli na jednoczesne podłączenie wszystkich wymaganych czujników.

W dalszej części procesu projektowania należy skupić się na opomiarowaniu i sterowaniu całym układem. Większość systemów umieszczono w okolicy zbiornika (patrz rys. 3.2). Takie ułożenie jest najwygodniejsze, ponieważ czujniki wymagają dostępu do wody, aby realizować pomiary. Do tego system sterowania blisko urządzeń wykonawczych i pomiarowych zwiększy przejrzystość układu. Pompy perystaltyczne wytwarzają niewielkie ciśnienia, więc nie transportują koncentratów na duże odległości. To drugie jest istotne ze względu działanie i kalibrację tych pomp na odpowiednie suplementowanie i oczyszczanie wody. Wymagane jest to do poprawnego funkcjonowania układu.



Rys. 3.2. Zbiornik wody (render wykonany w środowisku Autodesk Inventor Pro 2018)

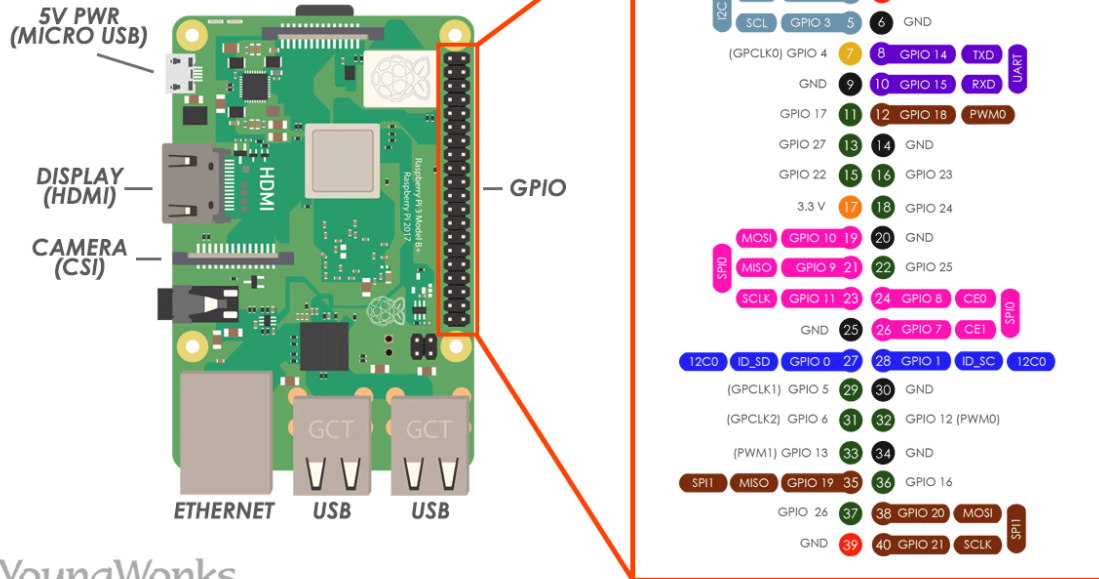
3.1. Elektronika centralna

Centralnym ośrodkiem odpowiedzialnym za sterowanie wszelkimi podsystemami w układzie oraz monitorowanie środowiska jest mikrokontroler. W tym projekcie zostanie wykorzystana płytki Raspberry Pi 4 B ze względu na popularność i elastyczność platformy. Zapewnia ona możliwości dalszego rozwoju projektu, dzięki zapasowi mocy obliczeniowej. Posiada dużą ilość GPIO (ang. *General Purpose Input/Output*) obsługujących wiele standardów komunikacyjnych takich jak:

- UART (ang. *Universal Asynchronous Receiver-Transmitter*),
- I2C (ang. *Inter-Integrated Circuit*),
- SPI (ang. *Serial Peripheral Interface*)

oraz złącza USB, Ethernet i wbudowane moduły Wi-Fi(2,4GHz, 5GHz) i Bluetooth 5,0 (patrz rys. 3.3). Mnogość wyboru sposobów jednoczesnej komunikacji oraz znaczna moc obliczeniowa przy zachowaniu małych wymiarów sprawia, że można zastosować ten mikrokontroler w wielu projektach.





YoungWonks

Rys.3.3. Rozkład GPIO w mikrokontrolerze Raspberry PI 4 B (obraz autorstwa Abhishek Aggarwal)

[02]

Przewagą Raspberry Pi nad konkurencją w postaci innych otwartych mikrokontrolerów jak ESP, STM32 czy Arduino jest to, że Raspberry Pi jest pełnoprawnym komputerem z ogromnym zapasem mocy obliczeniowej. Poza szybkością działania pozwala również na instalację systemu operacyjnego lub przygotowanego przez społeczność, darmowego specjalistycznego oprogramowania. Przykładami takich programów może być open plc pozwalający symulować działanie PLC (ang. *Programmable Logic Controller*), Mycodo lub Domestic będące gotowymi środowiskami do sterowania i monitorowania przy jednoczesnym zdalnym dostępie do pomiarów i gotowych wykresów.

3.1.1 Wybór protokołu komunikacyjnego

Warto nadmienić, że najważniejszym aspektem jest dobór czujników. Zdecydowana większość z nich obsługuje tylko jeden z protokołów komunikacyjnych, a sensory łączące się na więcej sposobów są sporadyczne. W celu poprawnego wybrania najlepszego protokołu komunikacyjnego do zastosowania należały omówić cechy każdego z obsługiwanych przez mikrokontroler. Poniżej znajduje się krótkie omówienie poszczególnych protokołów.

➤ UART

Asynchroniczny protokół UART oznacza uniwersalne asynchroniczne połączenie odbiornik - nadajnik. Asynchroniczność tego rozwiązania wynika z braku wspólnego zegara, od którego zależałaby prędkość przesyłu danych jak w przypadku I2C, gdzie od częstotliwości pulsacji zegara zależy długość trwania jednego bitu. Jest używany w zastosowaniach wymagających jednoczesnej komunikacji w obie strony, ponieważ obsługuje komunikację pełnego duplexu (ang. *full duplex*). Zapewnia jednoczesny przesył między dwoma urządzeniami bez możliwości rozbudowy struktury. Skutkuje to dużą prędkością wymiany informacji. Jest to możliwe poprzez fizyczne połączenie przy pomocy dwóch przewodów nadających jednostronnie. Dzięki temu komunikacja jest uproszczona i nie wymaga podawania adresu odbiorcy, ale wymagana jest poprawna konfiguracja obu urządzeń, aby operowały one na tym samym tempie Bauda (ang. Baud rate). Bez tego niemożliwa jest poprawna wymiana danych, ponieważ każde urządzenie dzieliłoby sygnał na inną ilość bitów, przez co odczytywałyby inne wartości i uzyskałyby inne informacje. Długość jednego bitu można wyliczyć ze wzoru (3.1) [04]

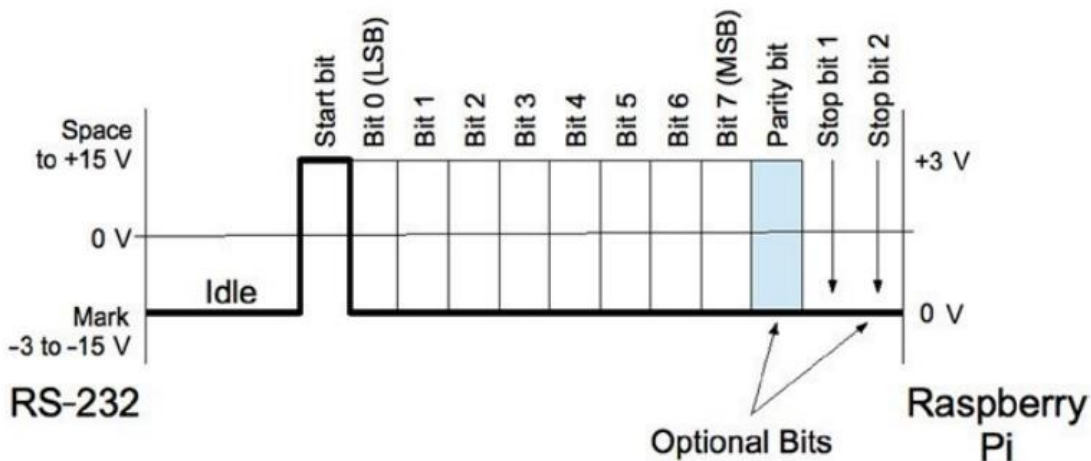
$$\text{czas trwania bitu} = \frac{1}{\text{Baud rate}} \text{ [s]} \quad (3.1)$$

gdzie:

Baud rate - liczba zmian symboli na sekundę [Bd]

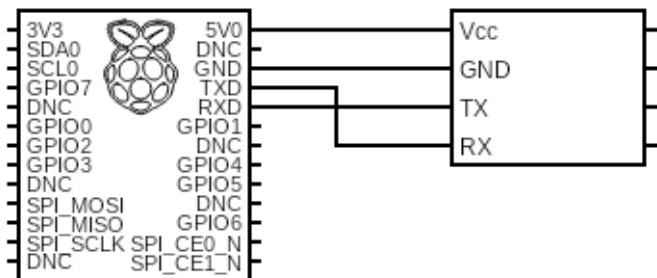
Poprawnie działający układ wymaga co najmniej czterech przewodów: dwóch zasilających oraz skrzyżowane przewody danych. Skrzyżowanie jest konieczne, ponieważ każde z urządzeń ma po jednym wejściu oznaczonym RX (ang. *receiver*) i wyjściu TX (ang. *transmitter*), które muszą zostać połączone (patrz rys.3.5.).

Zalecane napięcia na stanach logicznych dla Raspberry Pi przy tym rodzaju komunikacji jest w zakresie 0-3,3 V. Jeżeli odbiornik działa na innych zakresach można przykładowo użyć konwertera RS-232, który operuje na napięciach -15 do 15 V. Przed rozpoczęciem nadawania danych między urządzeniami należy ustalić wspólną konfigurację obejmującą tempo Bauda, długość danych oraz bity START i STOP. Transmisja zaczyna się od nadania bitu START, który polega na podniesieniu napięcia do 3,3V, po czym następuje przekazanie danych. W tym momencie odbiorca dostaje sygnał, po którym zaczyna nasłuchiwanie. Urządzenia nadające natychmiast przekazuje binarnie treść wiadomości (patrz rys. 3.4.). Metoda jest prosta, ponieważ nie wymaga podawania adresu odbiorcy, rejestrów ani odbierania potwierdzenia przekazania całego pakietu danych. Opcjonalne jest dodanie do szeregu danych bitu parzystości. Chroni on komunikację przed szumem i błędami. Rezygnując z jego implementacji, przyspieszana jest transmisja, przez skrócenie wiadomości.



Rys.3.4. Struktura transmisji UART [06]

Bity STOP służą do koordynacji odbiornika przed następnym bitem START. W ten sposób wykonana jest synchronizacja komunikacji w protokole UART. Przed wykorzystaniem tego sposobu przekazywania danych należy sprawdzić schemat połączenia urządzeń (patrz rys.3.5).



Rys. 3.5. Podłączenie urządzeń przez UART

Schemat z rys. 3.5. przedstawia połączenie bezpośrednie dwóch urządzeń, bez dodatkowych elementów elektronicznych wymaganych do poprawnego działania. Przed podłączeniem należy sprawdzić notę katalogową w celu sprawdzenia napięcia zasilania odbiornika. Istnieje możliwość wymaganego wyższego napięcia niż obsługiwane przez GPIO Raspberry Pi. Wymagałoby to użycia stabilizatora liniowego, np. stabilizatora LDO (ang. *Low-Dropout*) z serii LM78XX. Stabilizuje on napięcie na określonej wartości przy większym napięciu wejściowym.

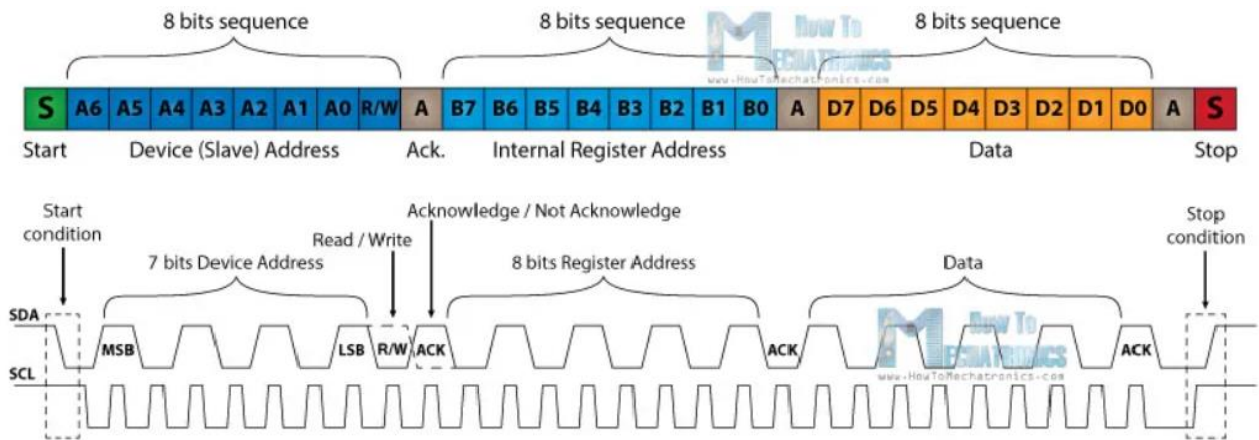
➤ I2C

Protokół synchroniczny I2C nazywany również IIC jest skrótem od angielskiego *Inter-integrated Circuit* co oznacza "pośrednik pomiędzy układami scalonymi". Jest on używany głównie do komunikacji z modułami oraz czujnikami przy pomocy magistrali. Obsługiwana jest transmisja półdupleks (ang. *half duplex*), co oznacza, że dane mogą być przekazywane w obie strony. Informacja może zmierzać zarówno z urządzenia nadrzędnego (ang. *master*) do podrzędnego (ang. *slave*) oraz w przeciwną stronę. Zmniejsza to prędkość przekazywania informacji, ponieważ nadaje tylko jedno urządzenie jednocześnie. Inaczej sygnał musiałby być kierowany w dwie strony po tym samym przewodzie. Spowodowałoby to niedostarczenie żadnej informacji przez wzajemne zakłócanie. I2C może obsługiwać wiele urządzeń typu master oraz slave na jednej linii magistrali. Master ma za zadanie koordynowanie komunikacją pomiędzy urządzeniami w magistrali. Urządzenia wiedzą, kiedy informacja jest kierowana do nich, ponieważ przed wysłaniem samych danych udostępniany jest adres odbiorcy, aby to urządzenie odczytało i zachowało dane w swoim buforze [06, 13, 14].

W celu poprawnego działania protokołu I2C potrzebne są cztery przewody: dwa zasilające i dwa transmisji SCL (ang. *Serial Clock Line*) oraz SDA (ang. *Serial Data Line*) odpowiadające kolejno za zegar do synchronizacji komunikacji oraz linie bitów danych (patrz rys. 3.7). Dzięki wspólnemu sygnałowi zegara oba komunikujące się urządzenia wiedzą, jak długo trwa jeden bit bez potrzeby definiowania wspólnego tempa Baud'a jak w przypadku komunikacji UART.

Domyślnymi wartościami wyjściowymi napięć na liniach jest stan wysoki [06]. Protokół działa na zasadzie ograniczonego czasu na zmianę stanu na szynie danych. Oznacza to, że stan logiczny reprezentowany przez napięcie podawane na wyjściu urządzenia może zmieniać się tylko w czasie, gdy stan na linii zegara jest niski. Przy zmianie stanu zegara na wysoki dane nie mogą już modyfikować swojej wartości poza wyjątkowymi sytuacjami opisanymi przy omawianiu struktury komunikacji.

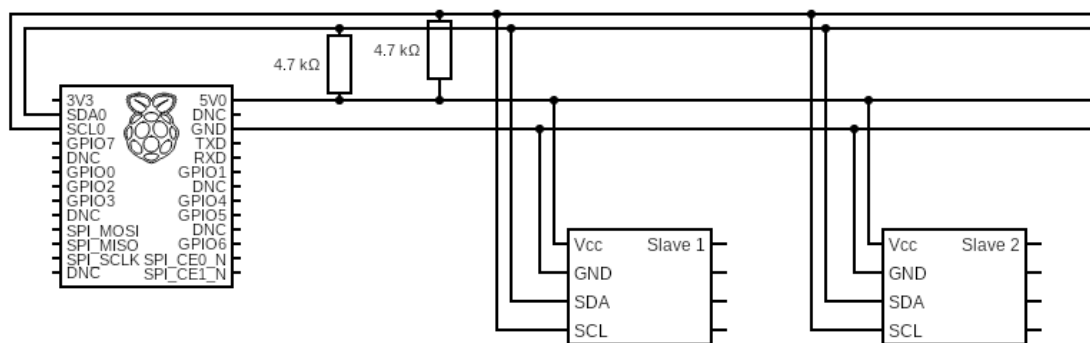
Struktura przesyłanych danych zaczyna się od specjalnego symbolu start rozpoczynającego każdą komunikację. Jest to zmiana stanu na linii danych na niski przy wysokim napięciu zegara. Następnie występuje podanie kilku bitowego adresu urządzenia, do którego kierowana jest informacja oraz to, czy master będzie zapisywał, czy odczytywał dane ze slave'a. Oznaczone to jest odpowiednim stanem logicznym na linii danych, gdzie stan niski oznacza zapis, a wysoki - odczyt. Kolejnym etapem jest podanie adresu rejestru wewnętrznego urządzenia odbiorczego, z którego dane mają zostać pobrane lub zapisane. Na koniec następuje transfer informacji. Wtedy wysyłany jest znak stop, czyli podniesienie linii SDA do stanu wysokiego podczas wysokiego stanu na linii SCL (patrz rys. 3.6.). I2C po każdym z etapów (poza znakami specjalnymi start i stop) wykonuje potwierdzenie czy sekwencja bitów została poprawnie odebrana. Dzieje się to przez zwolnienie linii danych dla urządzenia podrzędnego, który poprzez ustawienie odpowiedniego stanu logicznego na linii potwierdza otrzymanie danych [05, 06]



Rys.3.6. Struktura komunikacji protokołu I2C [05]

Znając zasadę działania, należy zapoznać się z zasadami fizycznego połączenia, aby poznać dodatkowe wymagania systemu w celu wyboru najodpowiedniejszej magistrali.

Schemat (patrz rys. 3.7) przedstawia typowy układ o topologii magistrali, ale jedynym odstępstwem jest wystąpienie dwóch rezystorów łączących linie I2C z zasilaniem. Rezystory te nazywają się podciągającymi (ang. *pull-up resistors*) i zapewniają domyślny wysoki stan logiczny na liniach sygnałowych oraz GPIO. Zwykle używa się w tym celu rezystorów z zakresu 2-10 k Ω . Najpopularniejszymi są jednak rezystancje 4,7k Ω oraz 10k Ω . Tak wysoka rezystancja zapewnia ograniczony pobór prądu przez rezystor, ponieważ doprowadziłoby to do jego niepotrzebnego nagrzewania i strat energii. Niektóre układy obsługujące I2C mają wbudowane te rezystory w swojej płycie, dlatego należy przed użyciem sprawdzić notę katalogową producenta.



Rys.3.7. Schemat połączeń magistrali I2C dla jednego urządzenia nadrzędnego

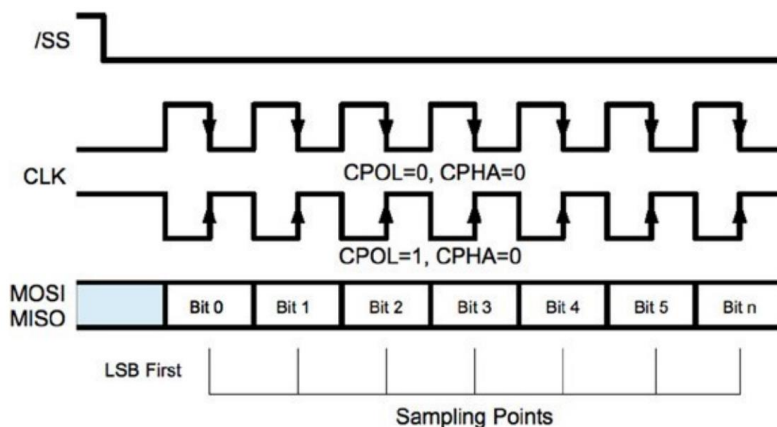
➤ SPI

Protokół synchroniczny SPI jest skrótem z angielskiego i oznacza szeregowy interfejs urządzeń peryferyjnych. Używa się go do komunikacji z urządzeniami slave takimi jak moduły lub czujniki przy pomocy magistrali. Jest to transmisja typu duplex, ponieważ dane mogą być przesyłane jednocześnie w obie strony pomiędzy slave'm i master'em. Znacznie zwiększa to prędkość komunikacji względem podobnego I2C. Używa się go, kiedy krytyczna jest szybka reakcja układu lub następują nagle zmiany parametrów. Ilość podłączanych urządzeń jest ograniczana przez ilość wyjść CS lub inaczej SS na mikrokontrolerze (ang. *Chip Select*). Odpowiada on za otwarcie komunikacji z urządzeniem podrzędnym. Protokół ten nie wymaga konfiguracji tempa Bauda czy bitu START i STOP, dzięki wspomnianym pinowi CS. Za synchronizację odpowiada zegar [06].

W celu poprawnego podłączenia urządzeń działających na magistrali SPI potrzebne są co najmniej 6 przewodów: 2 zasilające oraz minimum 4 przewody sygnałowe: odpowiadający za zegar do synchronizacji SCLK (ang. *Serial Clock*), odpowiadający za transmisję danych MISO (ang. *Master Input Slave Output*) i MOSI (ang. *Master Output Slave Input*) oraz wspomniany wcześniej CS odpowiadający za wybranie urządzenia do komunikacji z masterem (patrz rys. 3.9.).

Domyślnymi wartościami stanami logicznymi na pinach są stany wysokie, a całe trwanie transmisji jest możliwe tylko w czasie, gdy master utrzymuje stan niski na pinie CS w połączeniu z tylko jednym urządzeniem. Gdy nastąpi zmiana na wyższe napięcie przekazywanie informacji kończy się i wartości są odczytywane.

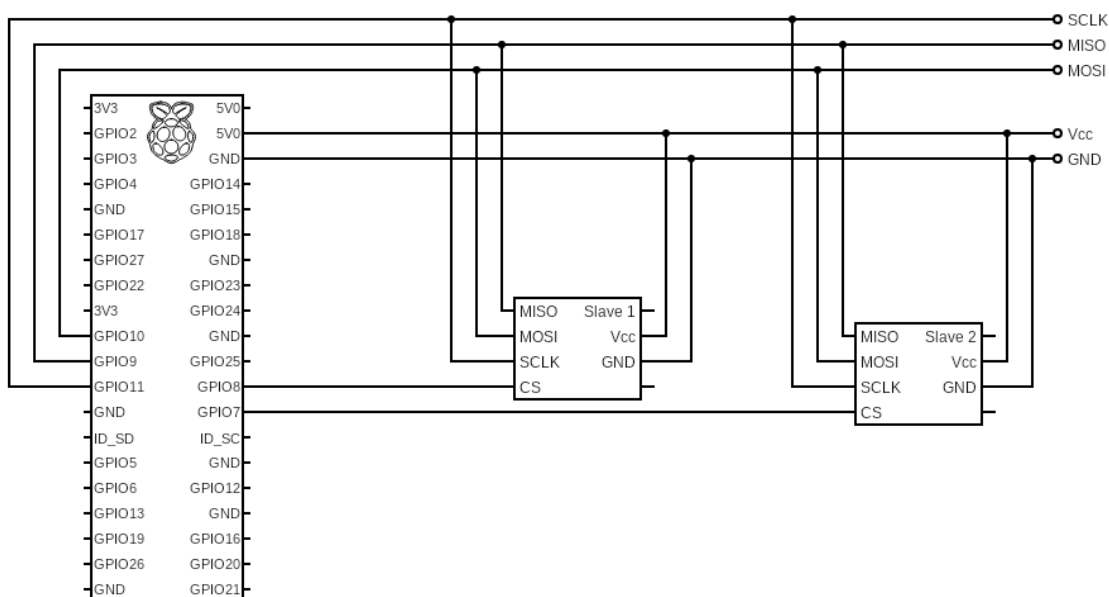
Struktura przesyłanych danych jest znacznie prostsza niż w przypadku podobnego rozwiązania I2C. Nie wymaga wysyłania adresów lub precyzowania rejestrów wewnętrznych. Odpowiedzialny za to jest wspomniany pin CS nazywany też SS. Kiedy mikrokontroler zmieni sygnał pinu na niski, rozpoczynając komunikację z wybranym urządzeniem, przesyłane są sygnały: zegara oraz sygnałowe MOSI i MISO, które jednocześnie w obie strony przekazują informacje (patrz rys.3.8.). Bity wysyłane mogą być zależnie od konfiguracji wysyłane od LSB (ang. *Least Significant Bit*) czyli najmniej znaczącego bitu lub MSB (ang. *Most Significant Bit*) najbardziej znaczącego bitu. Skonfigurować można również stan zegara, dla którego zmiana odpowiada za wywołanie próbkowania wartości bitów sygnałowych. Oznacza to, że można skonfigurować ten rodzaj komunikacji na cztery sposoby, co daje dużą elastyczność i dostosowywanie do potrzeb projektu [06].



Rys.3.8 Struktura komunikacji protokołu SPI [06]

Jedną z wad tego protokołu jest znacznie większy stopień skomplikowania połączeń niż w przypadku podobnej magistrali I2C ze względu na liczbę przewodów oraz mniejszą czytelność w przypadku fizycznego obiektu. Jest to cena za szybką komunikację pełnego duplexu.

Schemat (patrz rys.3.9) przedstawia układ magistrali bardziej rozbudowanej niż w przypadku poprzednich protokołów, ponieważ wymaga znacznie większej liczby przewodów. Jak zawsze warto sprawdzić noty katalogowe urządzeń, aby sprawdzić jakie zasilanie jest wymagane dla zapewnienia niezawodnego działania urządzeń podporządkowanych oraz nie spalenia mikrokontrolera.



Rys.3.9. Schemat połączeń układu magistrali SPI dla dwóch urządzeń podrzędnych

Po zapoznaniu się z zasadami działania każdego rodzaju magistrali w celu porównania cech każdego typu komunikacji sporządzono tabelę podsumowującą (patrz tabela 3.1) najważniejsze informacje, aby ułatwić wybór magistrali.

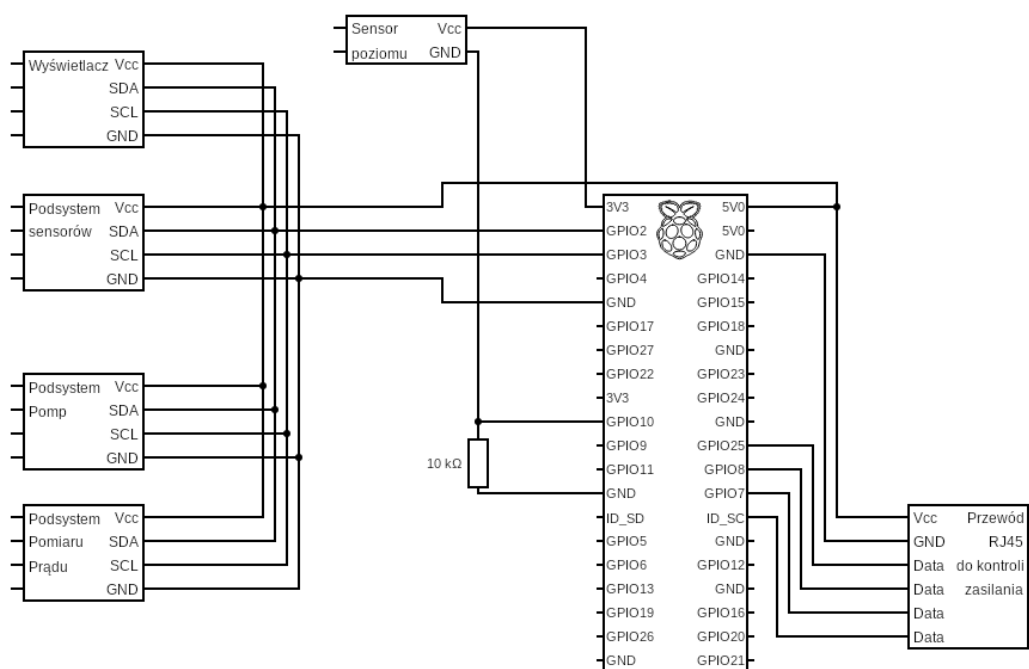
Tabela 3.1. Podsumowanie cech protokołów komunikacyjnych

	UART	I2C	SPI
Wymagana liczba pinów do transmisji	1/2	2	4+
Maksymalna prędkość komunikacji (Raspberry PI)	1,44kb/s	400kb/s	5Mb/s
Zasięg maksymalny	15m	Kilka metrów	Kilka metrów
Duplex	Simplex	Półdupleks	Dupleks
Synchronizacja	Asynchroniczny	Synchroniczny	Synchroniczny
Liczba masterów	0	wiele	1
Skomplikowanie połączeń	niskie	średnie	wysokie
Sposób synchronizacji	Tempo Bauda, bity START/STOP	Bity START/STOP, zegar	Zegar, stan pinu CS
Liczba obsługiwanych urządzeń	1	128	Tyle ile pinów CS (Raspberry PI = 2)

Najważniejszą cechą eliminującą magistralę SPI jest liczba pinów CS, która jest niewystarczająca na potrzeby tego projektu. Istniejące moduły rozszerzające, zwiększają niepotrzebnie koszt i skomplikowanie systemu, a dostępna jest alternatywa zapewniająca te parametry, bez dodatkowego sprzętu. Prędkości również nie muszą być największe, ze względu na raczej wolne tempo procesów związanych z wodą. Z tego samego powodu odpada użycie UART. Biorąc pod uwagę wszelkie cechy protokołów komunikacyjnych, na potrzeby projektu wybrano protokół I2C. Można pomieszczać protokoły i część urządzeń przenieść na różne magistralę, ale użycie jednej magistrali I2C zapewni większą przejrzystość układu, ale ograniczy możliwości zamawiania osprzętu do obsługujących tylko ten protokół. Zaletą użycia I2C jest również możliwość przyszłej rozbudowy poprzez obsługę wielu masterów, co zapewni rozszerzenie możliwości sterowania w systemie.

3.1.2 Schemat łączenia podsystemów

Mając wybrany protokół komunikacji szeregowej, można przystąpić do projektowania wstępnego schematu układu centralnego (patrz rys.3.10.). Zapewni to podział procesu projektowania na mniejsze wydzielone sektory oraz ułatwi ich opracowanie.



Rys.3.10. Poglądowy schemat sektora centralnego układu sterowania

Centralną część schematu stanowi Raspberry Pi, który przyjmuje rolę urządzenia nadrzędnego i sterownika. Jednocześnie może pełnić funkcję przetwarzania danych i wizualizacji, aby ułatwić użytkownikowi monitorowanie układu. Każdy z pozostałych podzespołów zostanie rozszerzony w dalszych podrozdziałach. Większość z podsystemów połączona jest z mikrokontrolerem przy pomocy pinów GPIO 2 i 3. Z rys.3.3 można odczytać, że odpowiadają one za komunikację I2C. Ze względu na brak wejść analogowych w płytce Raspberry każdy podsystem będzie potrzebował co najmniej jednego przetwornika analogowo-cyfrowego.

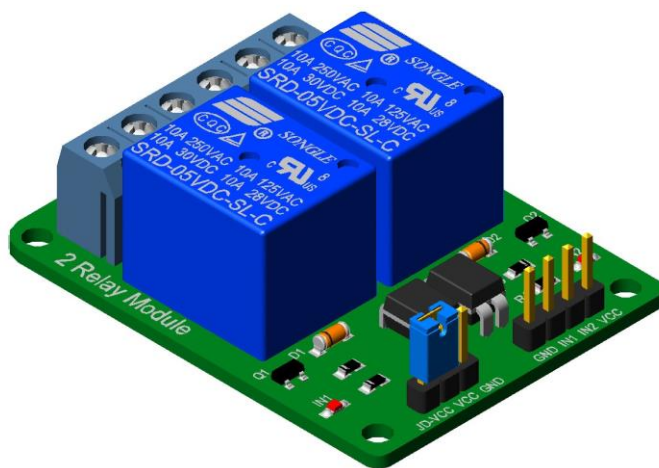
Czujnik poziomu przy niskim poziomie płynu otworzy się, co przerwie obwód. GPIO 10, które jest podłączone przez rezystor do uziemienia odbierze na wejściu stan niski. Rezystor używany jest w celu obniżenia pobieranego prądu do bezpiecznego zakresu w czasie zamknięcia pętli.

Przewód Rj45 jest połączony z układem zasilania i używa pinów cyfrowych odpowiedzialnych również za komunikację, ale żadna z magistral nie będzie wykorzystywana przy tym łączeniu.

3.2. System zasilania

Celem systemu zasilania jest dodanie do układu sterowanych gniazd elektrycznych. Pozwala to na automatyczne załączanie urządzeń podłączonych do głównego zasilania. Może posłużyć do włączania pomp, napowietrzacza lub przyszłych urządzeń rozszerzających możliwości projektu. Układ na podstawie sygnału z Raspberry Pi przełącza zasilanie urządzeń prądu stałego. System oparto na dwóch rodzajach modułów przekaźnikowych mechanicznych i półprzewodnikowych ze względu na ich mocne strony. Zabezpieczają one również mikrokontroler przed spalaniem, gdyby został podłączony do napięcia 240 V.

Przekaźniki mechaniczne (patrz 3.11) są w stanie załączać znacznie wyższe napięcia. Zostaną wykorzystane do przełączania urządzeń wymagających większych prądów oraz przełączających się rzadziej. Do projektu został wybrany moduł z cewką zasilaną napięciem 5 V umożliwiającą przepływ prądu do 10 A. W przypadku zwiększania częstotliwości zmiany zmniejsza się skuteczność przekaźnika oraz wydawało niepożądany dźwięk [08].



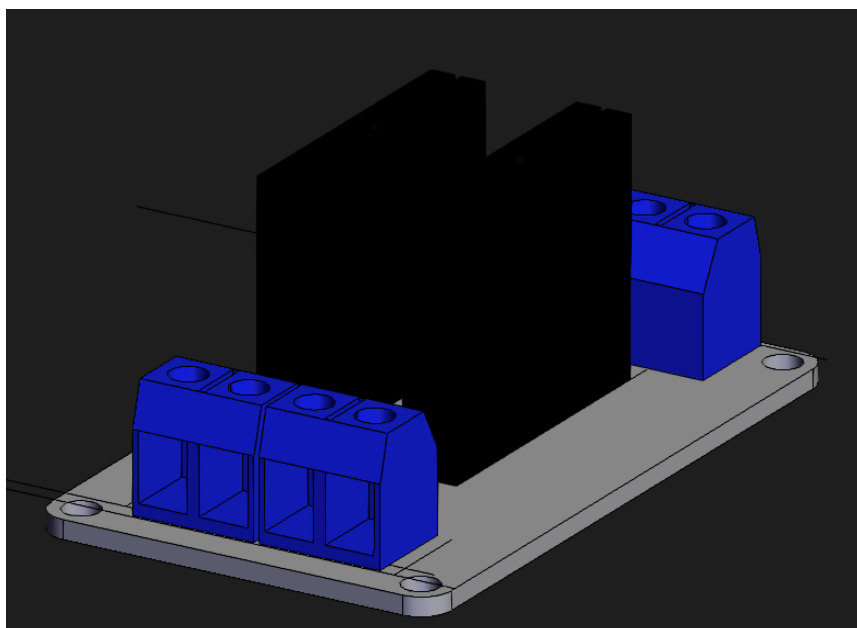
Rys. 3.11. Moduł przekaźnikowy mechaniczny

Źródło: Grabcad.com Richard Hornung SaintSmart 2 Channel Relay Module

https://grabcad.com/library/sainSMART-2-channel-relay-module-1/details?folder_id=615868

Półprzewodnikowe oznaczane również SSR (ang. *Solid State Relays*) (patrz rys. 3.12) mają znacznie wyższy współczynnik MTBF (ang. *Mean Time Before Failure*) przez brak ruchomych elementów mechanicznych. Oznacza to wysoką niezależność i długowieczność działania [17]. Są w stanie również szybciej się przełączać w porównaniu z mechanicznym odpowiednikiem (patrz tabela 3.2). W systemie zastosowano przekaźnik do prądu maksymalnego 2A przy napięciu zmiennym 240 V [09].





Rys. 3.12. Moduł przekaźnikowy półprzewodnikowy

Źródło: Grabcad.com Harry Cui OMRON - G3MB-202P - 2 Channel Solid State Relay Module

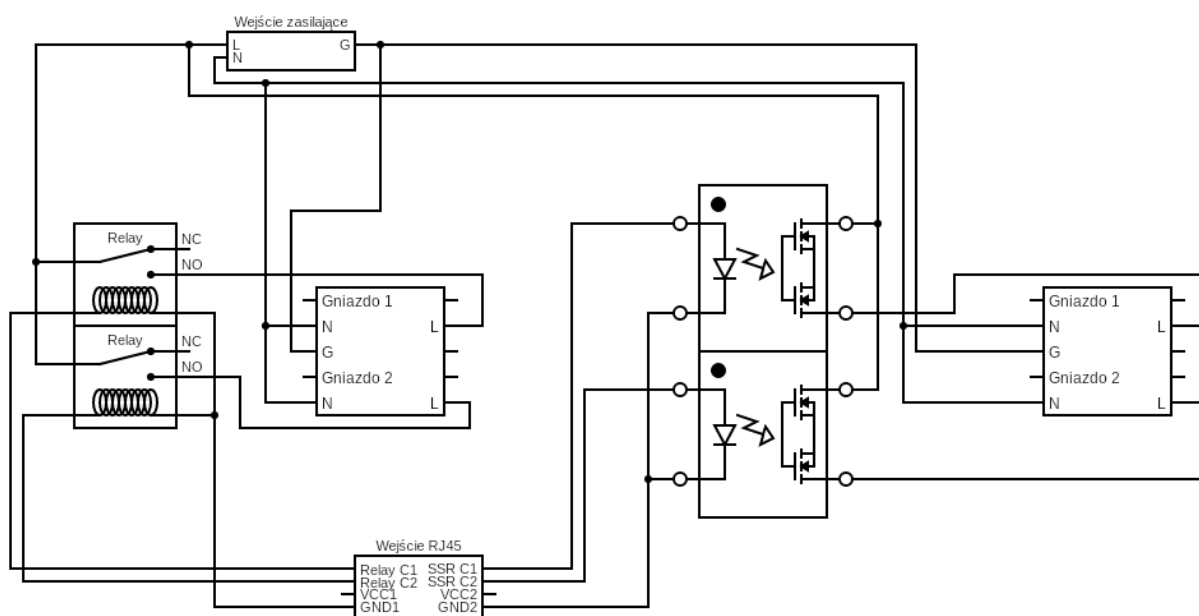
<https://grabcad.com/library/omron-g3mb-202p-2-channel-solid-state-relay-module-1>

Tabela 3.2. Parametry modułów przekaźnikowych

	Przekaźnik mechaniczny [16]	Przekaźnik półprzewodnikowy [15]
Liczba przekaźników	2	2
Napięcie sterujące	5 V	5 V
Maksymalny prąd przełączany	10 A	2A (240 V prądu przemiennego)
Maksymalne napięcie styków	250 V(prąd przemienny) 110 V (prąd stały)	240 V (prąd przemienny)
Prędkość przełączania	Mechanicznie 300 przełączeń/min Elektrycznie 30 przełączeń/min	1 ms
Zakres temperatur pracy	-40 - +85%	-30 - +80°C
Zakres wilgotności	45-85%	45-85%



Moduły te są kluczowym elementem skrzynki kontroli zasilania. W jej skład wchodzi również: wejścia kabla zasilającego, wejście RJ45 oraz dwa podwójne gniazda ścienna z uziemieniem. Układ ma na celu przekazywać sygnały sterujące przy pomocy przewodu ethernet. Z wejścia ethernet przewody rozchodzą się następująco: po dwa sygnały sterujące, masie i zasilaniu na każdy moduł przekaźnikowy (patrz rys. 3.13). Każdy ze sterowanych kanałów przekaźnika odpowiada jednemu sterowanemu gniazdku. Dla przejrzystości wszystkie przekaźniki mechaniczne zostały połączone z domyślnym ustawieniem NC (ang. *Normally Closed*). Wybudzanie polega na podaniu wysokiego stanu z GPIO mikrokontrolera. Załączane jest zasilanie doprowadzane przy pomocy wejścia na zasilanie i to ono jest w gniazdku, gdy urządzenie zostanie podłączone.



Rys. 3.13 Schemat podłączeń skrzynki kontroli zasilania

gdzie:

L- przewód liniowy 240 V prądu przemiennego,

G - przewód uziemienie,

N - przewód neutralny,

Relay C1/2 - sygnał sterujący kanały przekaźnika mechanicznego odpowiednio 1 i 2,

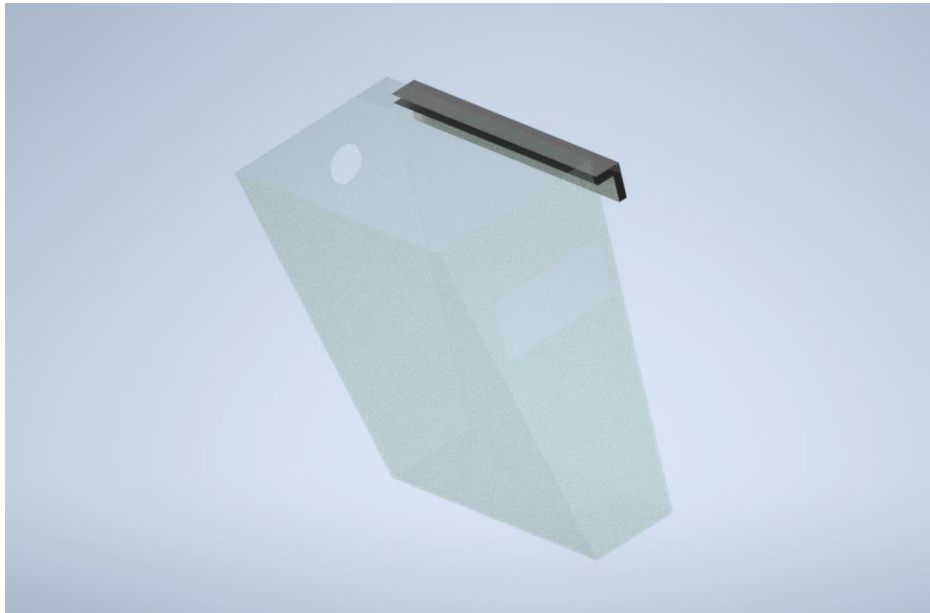
SSR C1/2 - sygnał sterujący kanały przekaźnika półprzewodnikowego odpowiednio 1 i 2.

Dzięki wykorzystaniu przewodu ethernet układ mógłby zostać rozbudowany o dwa gniazdzka, gdyby pojawiła się potrzeba sterowania dwoma kolejnymi gniazdami.

3.3. Podsystem pomiarowy

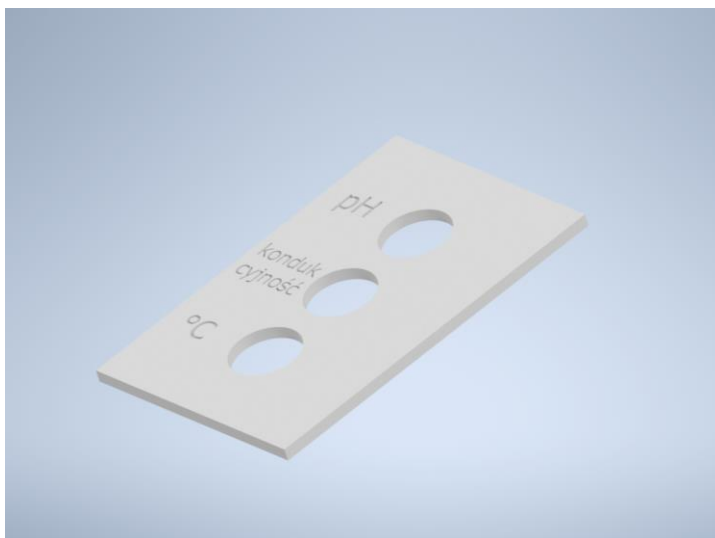
3.3.1 Składanie układu

System pomiarowy mieści się w zaprojektowanym pojemniku (patrz rys. 3.14), który zamontowany jest wewnątrz zbiornika głównego. Jego celem jest pomiar najważniejszych parametrów wody. Układ jest oddzielony od głównego zbiornika, aby w łatwy sposób wykonywać pomiary aktualnego stanu wody doprowadzanej do roślin.



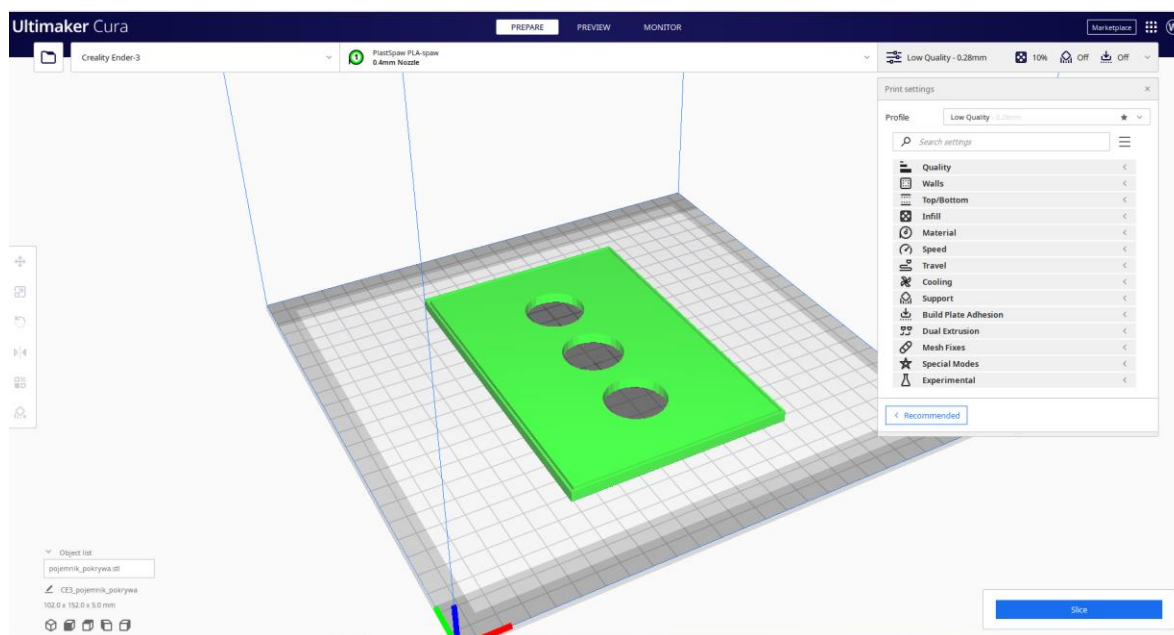
Rys. 3.14. Pojemnik pomiarowy (render wykonany w środowisku Autodesk Inventor Pro 2022)

Pojemnik ma 20 cm wysokości i 15 cm szerokości oraz dwa otwory po bokach. Jeden mniejszy wlotowy oraz większy wylotowy. Ciecz wpływa przez miernik przepływowego (dla niezanieczyszczonego pomiaru) do pojemnika pomiarowego. Drugi prostokątny otwór służy do odprowadzania wody do zbiornika głównego poprzez wypływ swobodny. Ustala to również poziom cieczy na stałym poziomie. W celu podtrzymania czujników zaprojektowano gotową do wydruku i modyfikowania przykrywkę z otworami (patrz rys. 3.15).



Rys. 3.15. Pokrywa pojemnika (render wykonany w środowisku Autodesk Inventor Pro 2022)

Dla czytelności użytkowników otwory zostały opisane. Do nich wsuwane są czujniki. Ze względu na różnice szerokości czujników plik pokrywki jest łatwo edytowalny w programie modelowania 3D. Przygotowanie do druku jest wykonywane przy pomocy oprogramowania typu slicer, odpowiedzialnego za podział modelu na warstwy. W tym projekcie użyto Ultimaker Cura w wersji 4.12.0 (patrz rys. 3.16), aby wygenerować G-code (ang. *Geometric code*). Kod ten zawiera ustawienia i parametry wydruku (patrz tabela 3.3), które są zrozumiałe dla maszyny. W tym przypadku drukarki 3D firmy Creality - Ender 3.



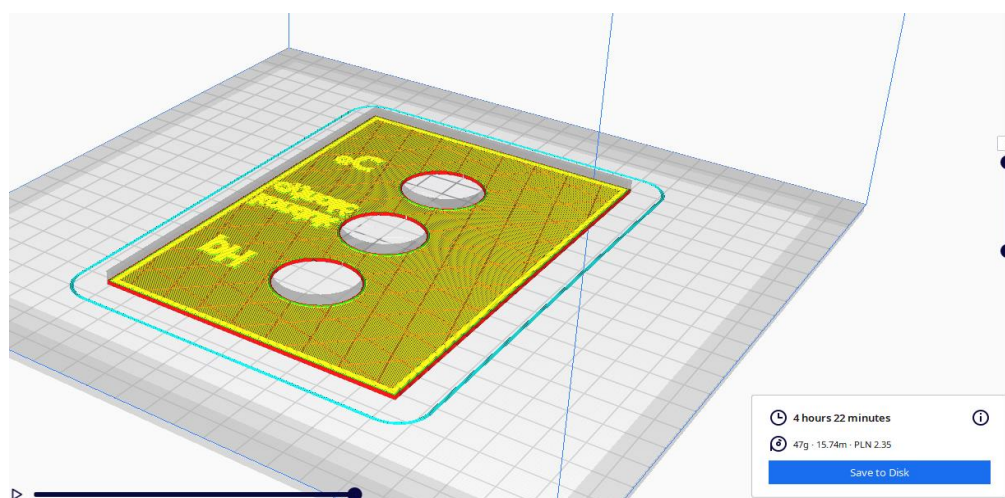
Rys. 3.16. Środowisko Ultimaker Cura 4.12.0



Tabela 3.3. Parametry wydruku drukarki 3D

Parameter	Wartość parametru
Wysokość warstwy [mm]	0,28
Szerokość ściany [mm]	0,8
Liczba ścian	2
Wypełnienie [%]	10
Wzór wypełnienia	Kwadratowy
Temperatura głowicy hot end [°C]	200
Temperatura stołu [°C]	70
Wyływ [%]	100
Chłodzenie [%]	100
Prędkość druku [mm/s]	50
Podpory	brak

Ustawienia zostały dostosowane do wydruku z plastiku PLA (ang. *Polylactic acid*), zwanym też polilaktydem o szerokości 1,75 mm. Materiał ten jest rozpuszczalny w wodzie, ale przykrywka nie ma bezpośredniego kontaktu z wodą. Zanurzenie tego plastiku nie ma konsekwencji, ponieważ żeby wpłynąć na wytrzymałość tego plastiku musiałby on być długotrwale zanurzony. Po pocięciu modelu otrzymuje się podgląd interpretacji kodu w wizualnej formie (patrz rys. 3.17). Użytkownik oprogramowania jest w stanie sprawdzić poprawność działania programu, estymowany czas wydruku oraz zużycie materiału.

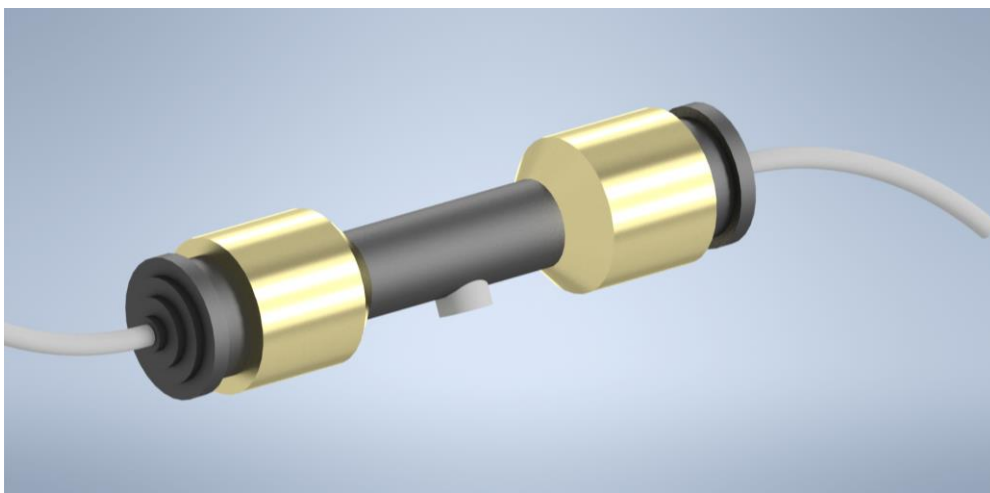


Rys. 3.17. Ultimaker Cura - cięcie modelu

Powstały element stanowi podporę dla czujników (patrz rys. 3.18) wysuniętych przez górne otwory. W połączeniu ze zbiornikiem pomiarowym stanowią środowisko do pomiarów parametrów wody dostarczanej przez rurę z czujnikiem przepływowym (patrz rys. 3.19).

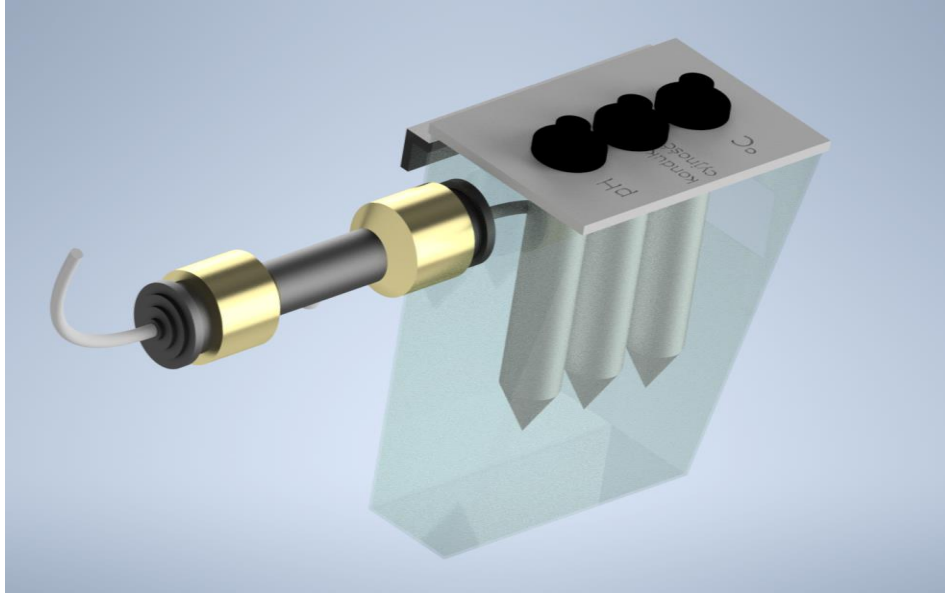


Rys. 3.18. Uproszczony model czujnika (render wykonany w środowisku Autodesk Inventor Pro 2022)

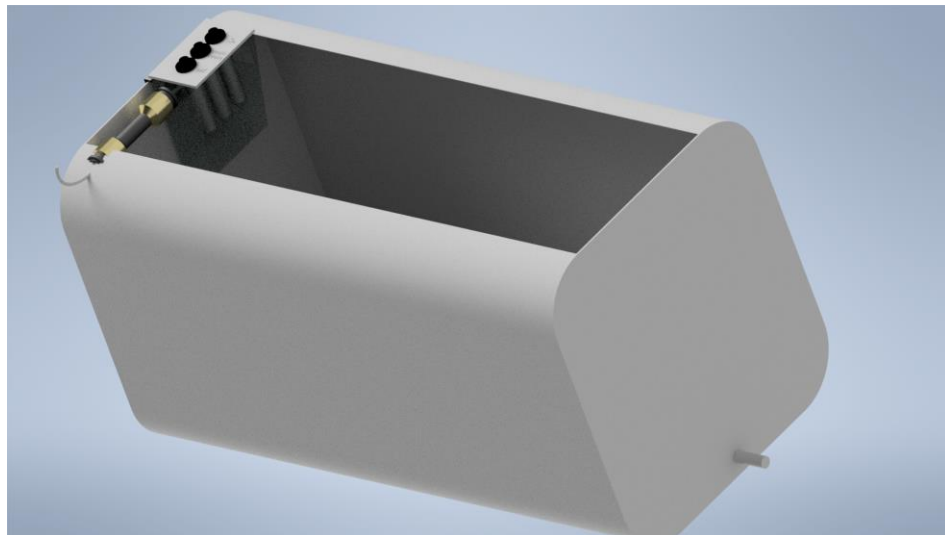


Rys. 3.19. Model miernika przepływu (render wykonany w środowisku Autodesk Inventor Pro 2022)

Mając wszystkie elementy, należy złożyć je w jeden gotowy system (patrz rys. 3.20) oraz umieścić w docelowym miejscu w zbiorniku głównym (patrz rys.3.21).



Rys. 3.20. Złożenie pojemnika pomiarowego (render wykonany w środowisku Autodesk Inventor Pro 2022)

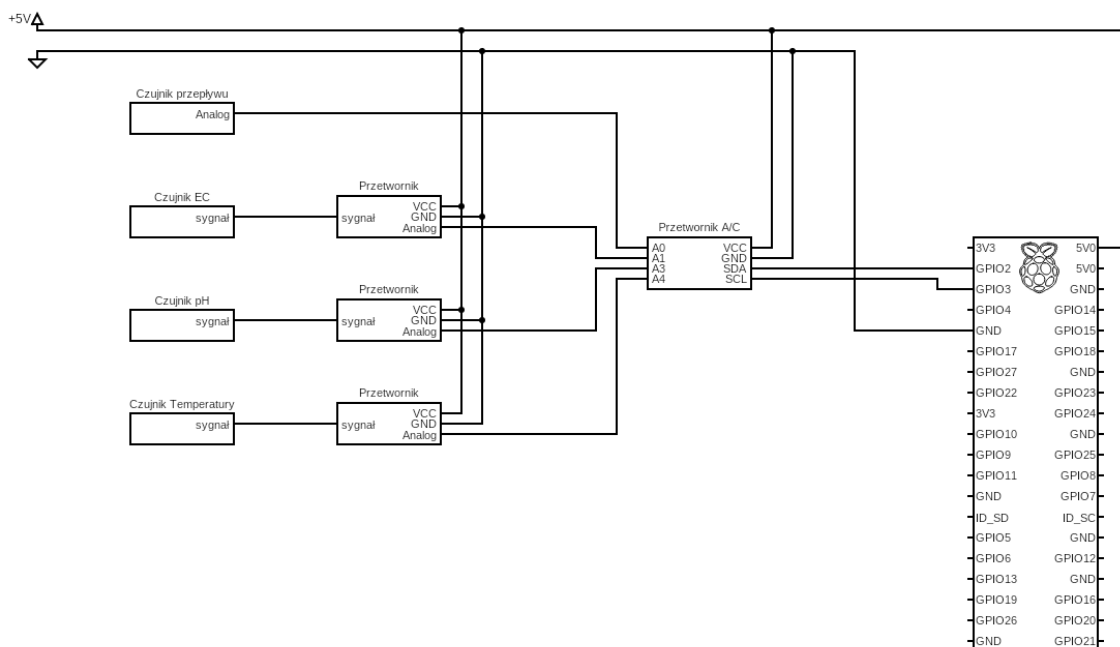


Rys. 3.21. Umieszczenie pojemnika pomiarowego w zbiorniku głównym (render wykonany w środowisku Autodesk Inventor Pro 2022)

Podsystem zaczepiony jest na ścianie zbiornika od wewnętrznej strony, aby woda wypływem swobodnym trafiała do zbiornika głównego. Druga rura wychodząca z czujnika przepływowego prowadzi do otworu wywierconego na początku górnego piętra, aby woda była pobierana przed roślinami, ponieważ może to wpływać na wyniki.

3.3.2 Schemat połączeń

Mając zaprojektowany model fizyczny czujnika, należy odpowiednio przygotować układ połączeń (patrz rys. 3.22). Ma on na celu zbieranie pomiarów i przekazanie ich do mikrokontrolera, aby móc regulować odpowiednimi parametrami procesu. Na wyjściu z czujnika lub jego przetwornika otrzymywany jest sygnał analogowy. Specyfikacja czujników firmy DFRobot wskazuje, że napięcie wyjściowe oscyluje w zakresie 3-3,5V. Stwarza to wymaganie użycia konwertera analogowo-cyfrowego, który umożliwi komunikację po magistrali I2C.



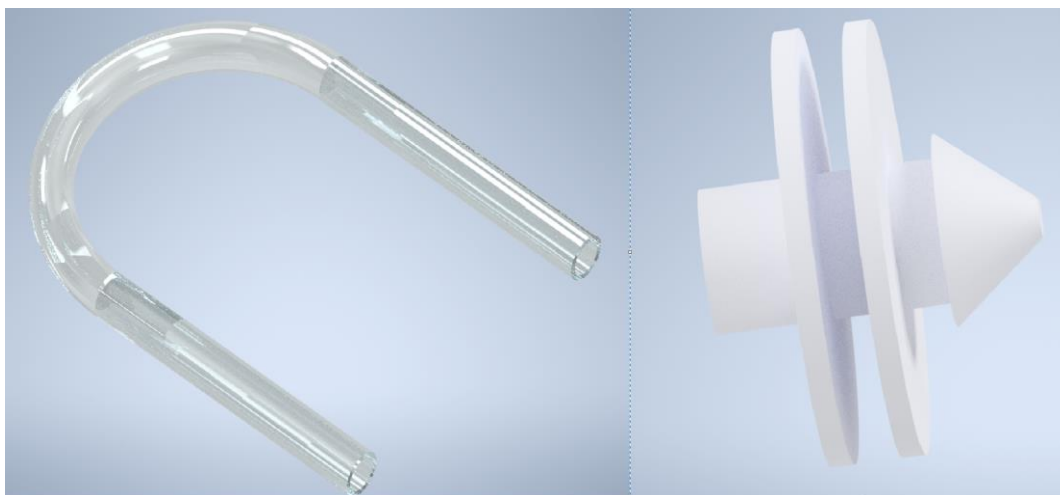
Rys. 3.22. Schemat połączeń podsystemu pomiarowego

Dzięki komunikacji całej sekcji przez magistralę redukuje się znacznie liczbę przewodów, które muszą transportować dane na odległość zbiornika. Dodatkowo wszystkie czujniki są blisko siebie, co ułatwi podłączenie kolejnych urządzeń do szyn SDA i SCL.

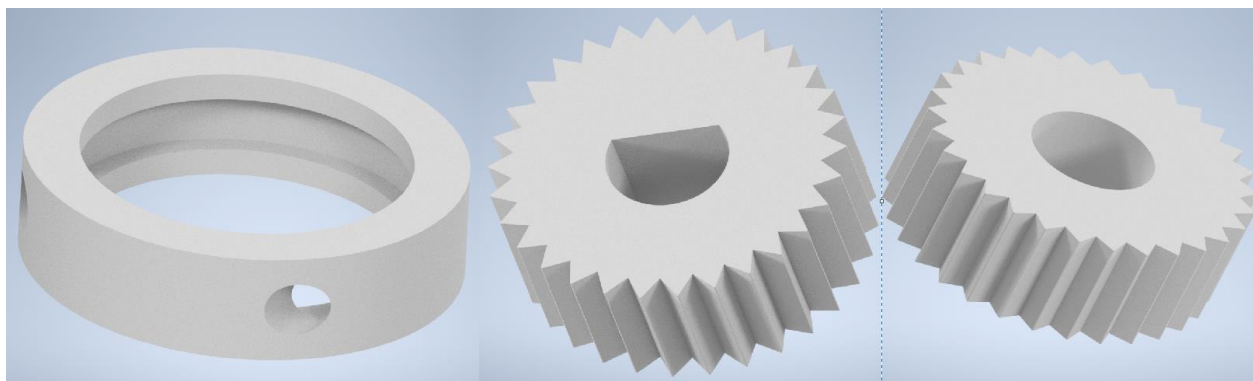
3.4. Podsystem dowartościowywania wody

Najważniejszymi elementami podsystemu dowartościowywania wody są pompy perystaltyczne i małe zbiorniki z odpowiednimi substancjami chemicznymi. Układ znajduje się bardzo blisko zbiornika, ponieważ jego celem jest dowartościowywanie wody przepływającej przez układ z roślinami.

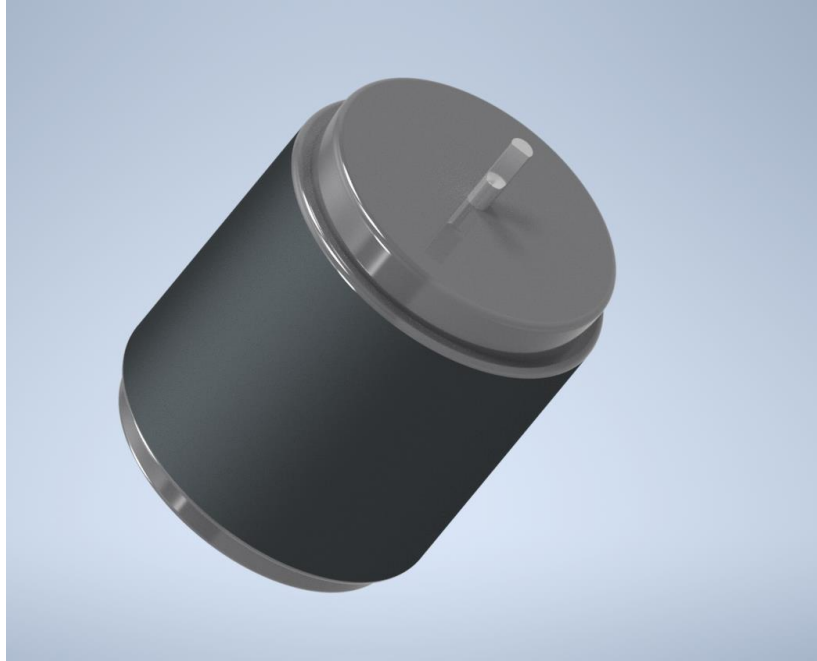
Pompa perystaltyczna składa się z trzech części: rurki (patrz rys. 3.23), rotatora (patrz rys. 3.24) oraz napędu (patrz rys.3.25). Konstrukcja urządzenia jest bardzo prosta. Rurka z jednej strony jest zanurzona w cieczy, która ma być przepompowywana przechodzi przez wnętrze rotatora (patrz rys. 3.26). Rotor może przyjmować wiele postaci od specjalnie wyprofilowanych łożysk przyczepionych do stałego stelaża po zwykłą przekładnię planetarną. Niezależnie od formy wszystkie działają jednakowo. Nacisk na rurkę w kilku punktach doprowadza do punktowych sprężeń i relaksacji. Zmiany te sprawiają, że podczas obrotu rotatora wytwarzane jest podciśnienie. Zasysa ono wodę w kierunku obrotu i przekierowuje na drugi koniec pompy. Napęd służy do obracania przymocowanym do niego rotatorem (patrz rys. 3.27).



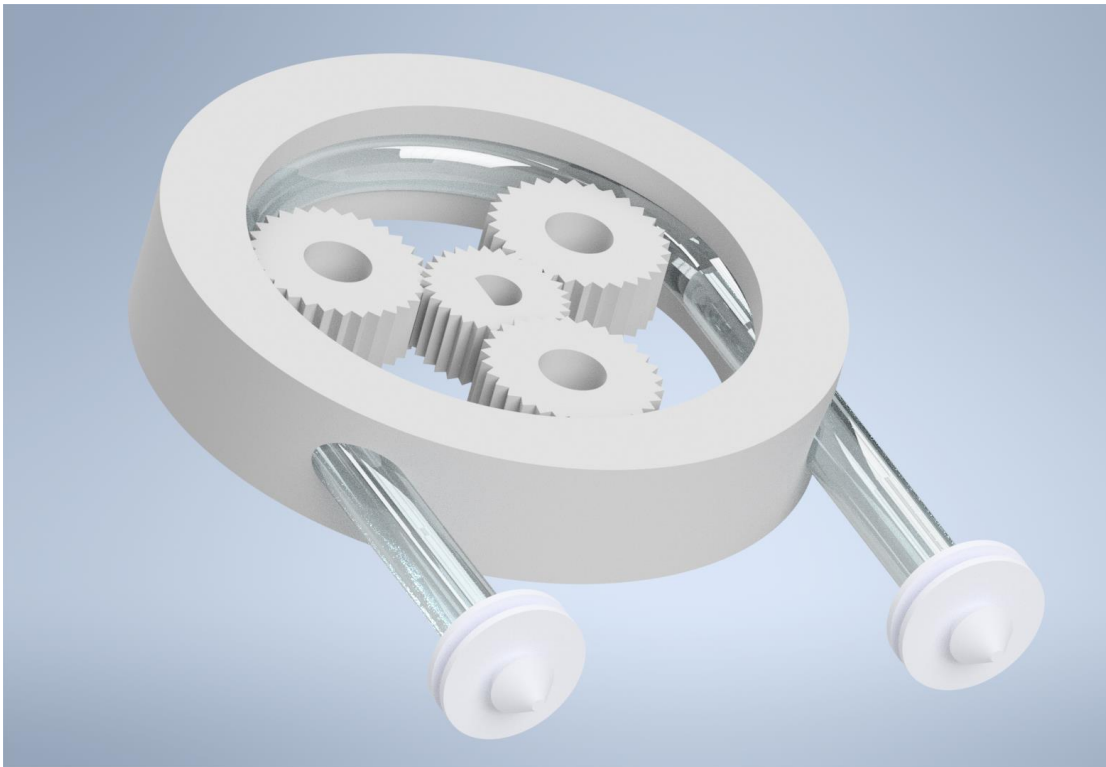
Rys. 3.23. Rura i jej złączka (render wykonany w środowisku Autodesk Inventor Pro 2022)



Rys. 3.24. Elementy przekładni planetarnej (render wykonany w środowisku Autodesk Inventor Pro 2022)



Rys. 3.25. Silnik napędzający pompę (render wykonany w środowisku Autodesk Inventor Pro 2022)



Rys. 3.26. Złożenie przekładni (render wykonany w środowisku Autodesk Inventor Pro 2022)



Rys. 3.27. Pompa perystaltyczna (render wykonany w środowisku Autodesk Inventor Pro 2022)

Ten typ pompy wyporowej ma wiele zalet, które są kluczowe w tym projekcie. Ze względu na wymagane niezanieczyszczenie substancji, konstrukcja zapewnia szczelność i sterylność przepływającej cieczy. Sama pompa również nie musi być zanurzona w wodzie, co nie zanieczyści substancji. Do tego możliwy jest przesył nawet bardzo gęstych i lepkich środków. Zapewnia to elastyczność w przypadku podawanych stężeń. W przypadku tego projektu gwarantuje to duży zapas co do kompatybilności używanych substancji. Nawet podczas pompowania wymagających cieczy zapewniony jest stały i proporcjonalny przepływ. Umożliwia to łatwą kontrolę ilości wypompowywanej substancji przez kontrolę prędkości obrotu silnika. Do tego środek może poruszać się w dwie strony. Ponadto nie ma możliwości cofnięcia się podczas pompowania. Jedynym elementem, który może ulec uszkodzeniu jest rurka przeznaczona do celów spożywczo-farmaceutycznych. Jest ona narażona ze względu na mechaniczne sprężania i rozprężania spowodowane uciskiem przez elementy rotora. W przypadku pęknięcia ciecz zostaje w komorze pompy niezanieczyszczona. Wymiana elementu jest prosta i tania, więc można uznać tego typu pompę za wysoce niezawodną. Dobrana została pompa o parametrach wystarczających do dostarczania niedużych ilości substancji regulujących parametry cieczy (patrz tabela 3.4). Dzięki temu możliwa jest prosta i dokładna regulacja użytych stężeń.

Tabela 3.4. Parametry pompy perystaltycznej

Parameter	Wartość
Napięcie wejściowe sterownika [V]	5-6
Maksymalny ciągły pobór prądu sterownika [A]	1,8
Maksymalny prąd sterownika [A]	2,5
Rodzaj napędu	Silnik DC
Napięcie pompy [V]	5-6
Średnica tuby wewnętrzna / zewnętrzna [mm]	2,5 / 4,5
Przepływ [ml/min]	45
Zakres temperatury otoczenia [°C]	0-40
Ilość rolek dociskowych rotatora	3

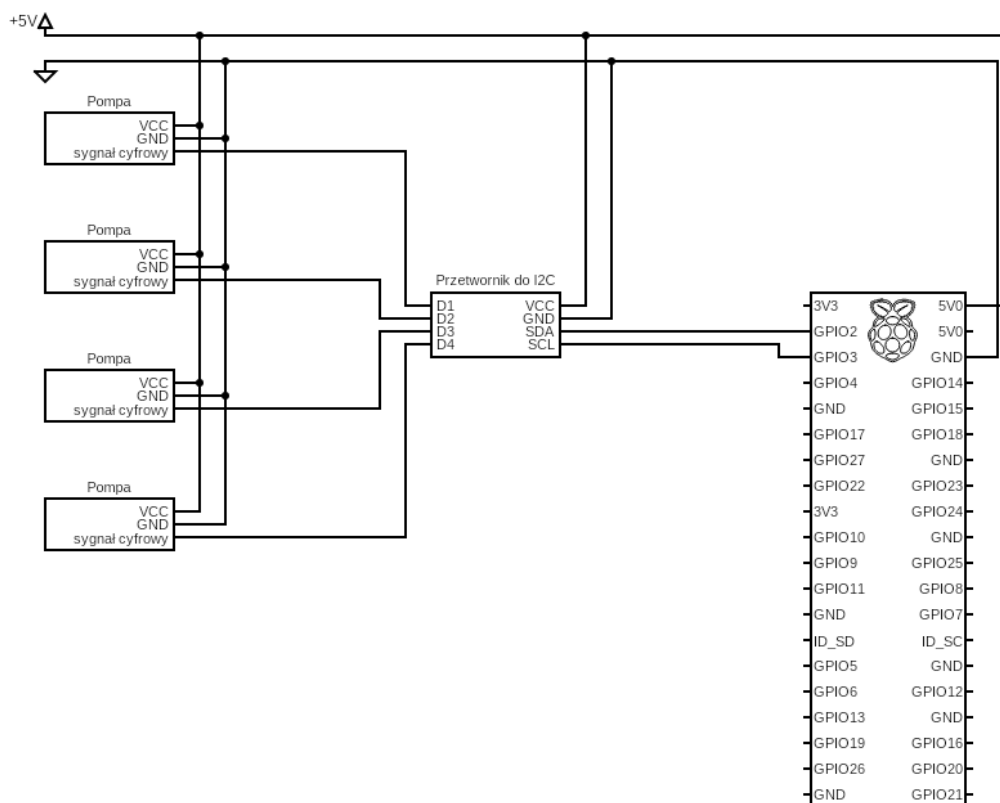
Pompa perystaltyczna zostanie zamontowana w wydrukowanym na drukarce 3D elemencie służącym za uchwyt montażowy oraz butelka będąca zbiornikiem na substancje (patrz rys. 3.28). Drugi koniec rurki prowadzi do zbiornika głównego.



Rys 3.28. Cała struktura pompująca (render wykonany w środowisku Autodesk Inventor Pro 2022)



Mając zaprojektowane urządzenie, należy przygotować schemat połączeń. Uwzględnione zostanie połączenie do mikrokontrolera (patrz rys. 3.29).

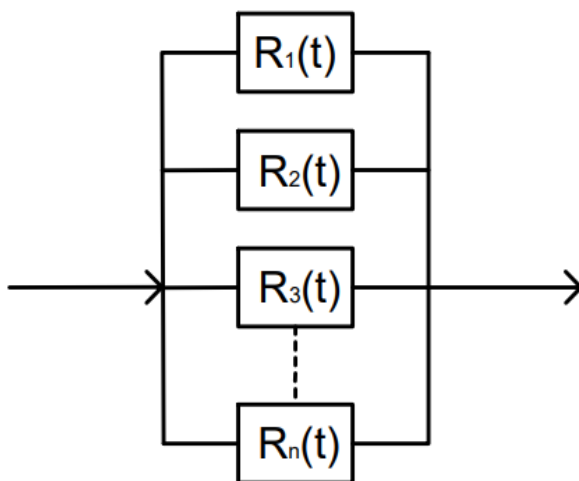


Rys. 3.29. Schemat połączeń podsystemu dowartościowywanie wody

Cały układ składający się z czterech pomp powinien zostać zawieszony bardzo blisko górnej krawędzi zbiornika. Takie umiejscowienie powinno również być relatywnie blisko jednostki centralnej, aby zmniejszyć długość przewodów potrzebnych do komunikacji. Montaż powinien być stabilny oraz gwarantować wypływ cieczy z pompy do zbiornika głównego. Można w tym celu użyć przedłużenia rurki, ale warto pamiętać, że powinna być sterylna i przeznaczona do zastosowań spożywczo-farmakologicznych. Zapewni to dostarczenie jak najczystszej substancji do wody i odporność na więcej substancji chemicznych.

3.5. Podsystem kondycjonowania i pompowania wody

W tym podrozdziale przedstawiony zostanie zaprojektowany podsystem, którego zadaniem jest dbanie o stan wody w zbiorniku oraz dostarczanie jej do korzeni roślin. Są to elementy kluczowe dla działania całego systemu hydroponicznego. Ze względu na skutki do jakich może powodować awaria urządzeń tego podsystemu zalecane jest zwiększenie niezawodności systemu. Opierając się na teorii niezawodności, można to zrobić przy pomocy struktury równoległej jednakowych urządzeń (patrz rys. 3.30).



Rys. 3.30. Przykładowa struktura równoległa (rysunek wykonany w środowisku Autodesk AutoCAD 2019)

gdzie:

$R(t)$ - niezawodność dla przedziału czasu t .

W przypadku tego układu niezawodność pomp jest głównym elementem wymagającym uwagi. Dlatego dla nich zostanie sprawdzona korzyść z zastosowania struktury. W pierwszej kolejności należy zapoznać się z zależnościami pomiędzy zawodnością (Q) i niezawodnością (R), które wyrażają prawdopodobieństwo. Są one swoim przeciwieństwem, a zatem ich suma wynosi 1. Dlatego zawodność elementu można ustalić przy pomocy

$$Q_n = 1 - R_n \quad (3.2)$$

gdzie:

n - liczba elementów w systemie.



Używając otrzymanego we wzorze (3.2) parametru, należy obliczyć wartość dla całego systemu.

$$Q_s = \prod_{i=1}^n Q_i \quad (3.3)$$

Mając całościową zawodność systemu (3.3), można wyznaczyć niezawodność systemu przy pomocy odwrotnego wzoru (3.2) można wyznaczyć całkowitą niezawodność systemu. Dla układu badanego do tego projektu należy przyjąć posiadanie 2 elementy oraz jednakowość urządzeń, więc

$$R_1 = R_2 = R \quad (3.4)$$

Następnie należy wyznaczyć wzór dla niezawodności, zaczynając od wyliczenia zawodności.

$$Q_{s \text{ systemu}} = (1 - R)^2 \quad (3.5)$$

$$R_{s \text{ systemu}} = 1 - Q_{s \text{ systemu}} \quad (3.6)$$

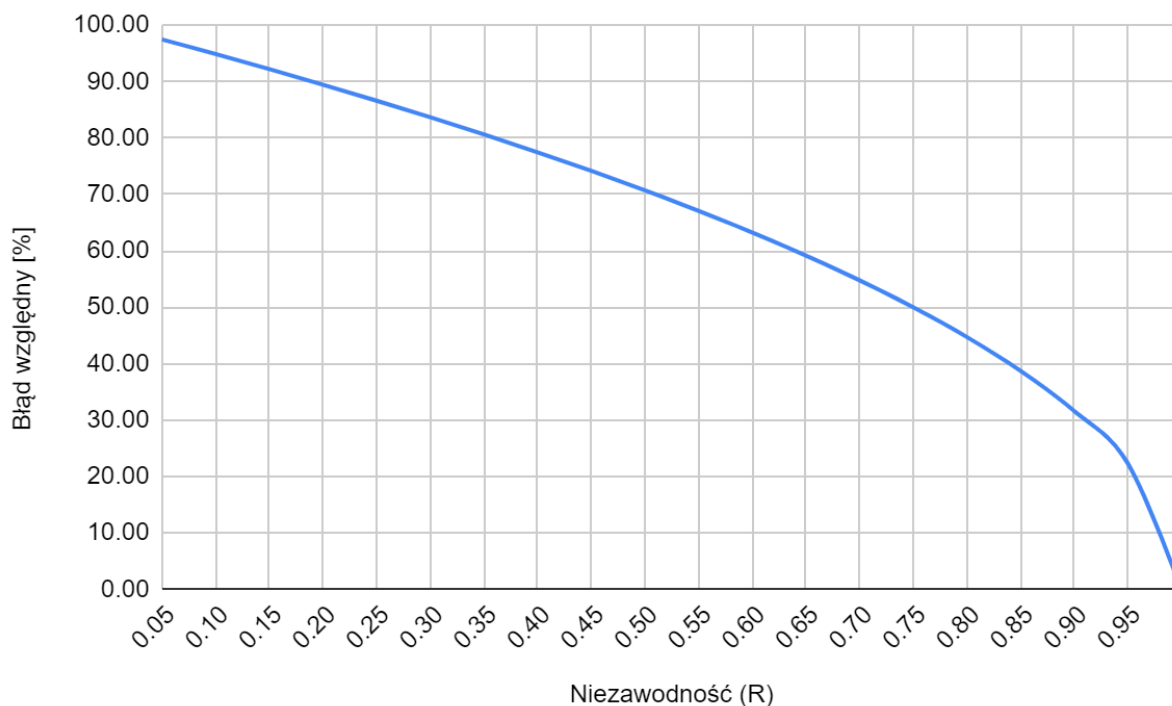
Otrzymany został wzór niezawodności systemu równoległego złożonego z dwóch jednakowych elementów (3.6). Przed implementacją warto zobaczyć potencjalne zyski niezawodności związane z podwójną inwestycją. W tym celu użyty zostanie błąd względny z porównaniem do użycia pojedynczej pompy.

$$\text{błąd względny} = \frac{|R_{s \text{ systemu}} - R|}{R_{s \text{ systemu}}} \cdot 100\% = \frac{|1 - (1 - R)^2 - R|}{1 - (1 - R)^2} \cdot 100\% = \frac{|R - R^2|}{2R - R^2} \cdot 100\% \quad (3.7)$$

Ze wzorem błędu względnego warto sprawdzić potencjalne zyski zależne od wartości niezależności (patrz rys. 3.31). W tym celu zostanie sporządzona tabela (patrz tabela 3.5) dla pełnego zakresu wartości z rozdzielczością 0,05. Tabela służy do sprawdzenia, w jakich zakresach niezawodności opłacalne jest rozważanie implementacji drugiego urządzenia lub pomoże podczas szukania odpowiedniej pompy.

Tabela 3.5. Zależność błędu względnego od parametru niezawodności

Niezawodność (R)	Niezawodność systemu (Rs)	Błąd względny	Błąd względny (%)
0,05	0,03	0,97	97,47
0,10	0,05	0,95	94,87
0,15	0,08	0,92	92,20
0,20	0,11	0,89	89,44
0,25	0,13	0,87	86,60
0,30	0,16	0,84	83,67
0,35	0,19	0,81	80,62
0,40	0,23	0,77	77,46
0,45	0,26	0,74	74,16
0,50	0,29	0,71	70,71
0,55	0,33	0,67	67,08
0,60	0,37	0,63	63,25
0,65	0,41	0,59	59,16
0,70	0,45	0,55	54,77
0,75	0,50	0,50	50,00
0,80	0,55	0,45	44,72
0,85	0,61	0,39	38,73
0,90	0,68	0,32	31,62
0,95	0,78	0,22	22,36
1	1	0	0



Rys. 3.31. Błąd względny od niezawodności

Jak widać nawet w sytuacji, gdy niezawodność pompy to 95% implementacja drugiej pompy i tak zapewni poprawę. Wynika z tego, że nawet wtedy warto pomyśleć o implementacji drugiego urządzenia. Pompy przeważnie nie posiadają tak wysokich parametrów jak 0,95, co zapewnia znacznie większe wyniki.

Znając liczbę używanych urządzeń, należy dobrać taki model, aby parametry zapewniały odpowiednie warunki rozwoju roślin. Źródła rekomendują różne wartości wypompowywanej wody w zależności od roślin, pochylenia, wielkości rur przepływowych oraz temperatury otoczenia. Zalecany przepływ waha się pomiędzy $20 - 120 \frac{l}{h}$ [10,11]. Należy go dostosować ręcznie do obiektu. W pracy [13] powołano się pozycję [22] gdzie w 1999 r. w wyniku badań uzyskano najlepsze warunki do hydroponicznej uprawy sałaty. Warunki te obejmowały rurę o szerokości 80 mm, nachyleniu 1,5% i natężeniu przepływu $30 \frac{l}{h}$. Chcąc przenieść to na większy system, należałoby zwiększyć przepływ. Z odniesienia wynika, że kształt rur użytych w badaniach był prostokątny, zatem aby oszacować przybliżoną wielkość i przeskalować proporcjonalnie, należałoby użyć co najmniej $41,25 \frac{l}{h}$. Jest to wartość na jedną rurę. W przypadku konstrukcji zaprojektowanej w rozdziale 2 warto rozdzielić strumień przy pomocy trójnika, co oznacza również podwojenie ilości wypompowywanej wody. Wymagany przepływ to $82,5 \frac{l}{h}$. Nie należy się martwić o ciśnienie, ponieważ woda nie jest wypuszczana bezpośrednio na korzenie, a musi zwolnić na kolanie 90° . Są to oczywiście wielkości przybliżone i



należy dopasować przepływ do układu, ale daje to pogląd w jakich zakresach należy poszukiwać urządzenia.

Mając dobrane zakresy przepływu pomp zanurzeniowych, należy połączyć je z zasilaniem. Połączone są one do sterowanych wejść z sekcji zasilania. Nie są one sterowane z poziomu kontrolera, ponieważ działają ciągle z tą samą nastawą i reguluje się je ręcznie tylko w przypadku zmiany rośliny. Z dwóch połączonych pomp jednocześnie działa tylko jedna. Kiedy czujnik przepływu wykrywa, że pompa przestaje dostarczać wystarczającą ilość wody, a czujnik poziomu nie sygnalizuje braku wody, pompa powinna być wyłączona przez odcięcie zasilania za pomocą przekaźnika, a w jej miejsce mikrokontroler powinien zasilić następną. Należy pamiętać, że założenie kilku filtrów na wypływie z rur przepływowych zadziała dobrze na żywotność pomp. Zapobiegnie to ich zapychaniu się przez drobinki roślin, które mogą spłynąć ze strumieniem. W przypadku takich mechanicznych elementów wystarczy może kilka warstw drobnej siateczki.

Następnym ważnym elementem jest również element kondycjonujący wodę, co ma spowolnić rozwój mikroorganizmów w zbiorniku. Pomimo, że woda jest w obiegu to duża część jest stojąca przez większość czasu. Są to idealne warunki dla rozwoju niepożądanych glonów i bakterii, które zagrażają zdrowiu roślin. Chcąc temu zapobiec, należy zainstalować kamienie napowietrzające, które swoim działaniem wzrusza wodę i pomaga uwolnić metan, azot i dwutlenek węgla z wody. Dzięki tej prostej filtracji kamień napowietrzający nie pozwala osiąść niechcianym organizmom osiąść na dnie zbiornika. Nie jest to rozwiązanie idealne, ponieważ po czasie organizmy i tak mogą osiedlić się na roślinach i przenieść do wody. Zapobiec temu mógłby drogi filtr odwróconej osmozy założony na wejściu lub wyjściu układu przepływowego, ale zakładając zbiornik wystarczający na 5-7 dni nie ma potrzeby na taką inwestycję. Należy dobrać dwa kamienie przystosowane do zbiorników 300l i zasilić je przy pomocy pozostałych dwóch wejść przekaźnikowych.



4. SYSTEM STEROWANIA

Oprogramowanie jest ważną częścią każdego automatycznego systemu. To ono rozszerza możliwości sprzętu i pozwala sterować wszystkimi parametrami. Dzięki programowej części projektu można wydobyć maksymalny potencjał fizycznych urządzeń poprzez procesy doboru najlepszych wartości sygnałów sterujących oraz samych metod sterowania. W dzisiejszym przemyśle oprogramowanie używane jest do zwiększenia efektywności. Osiąga się to poprzez zwiększanie prędkości wykonywania zadań, optymalizacji drogi ruchów robota i automatyzacji całych linii produkcyjnych. Poza szybkością pracy oprogramowanie pozwala zachować najwyższe standardy bezpieczeństwa. To program odpowiada za szybkie reagowanie na sygnały z czujników oraz parametrów prądowych urządzeń. Zapewnia to błyskawiczne odpowiedzi na zmiany otoczenia i pozwala unikać nieszczęśliwych wypadków. Co więcej cały system akwizycji, zarządzania i nadzoru oparty jest na oprogramowaniu rozwijanym przez firmę na własne potrzeby lub przez firmę zewnętrzną i dystrybuowaną przy pomocy subskrypcji. Systemy te nazywane zbiorczo SCADA (ang. *Supervisory Control and Data Acquisition*), które jest w dzisiejszych czasach podstawowym wyposażeniem większości zakładów produkcyjnych. W tym rozdziale opisany zostanie proces projektowania systemu sterowania i monitorowania całego systemu hydroponicznego. Sterowanie zostanie wykorzystane gotowe rozwiązanie, oparte na otwartej licencji.

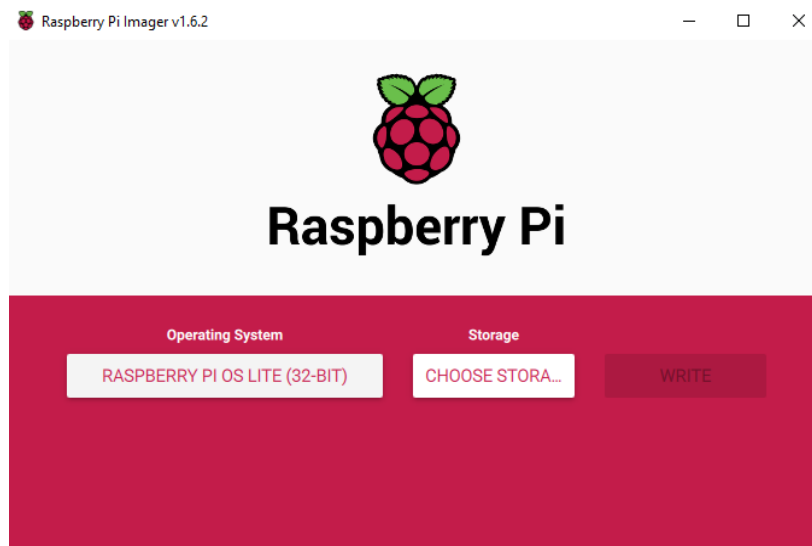
4.1. Konfiguracja urządzenia

Pierwszym krokiem w przygotowaniu urządzenia do pracy jest wgranie systemu operacyjnego, który będzie stanowił platformę dla oprogramowania. W celu poprawnej instalacji należy przygotować elementy (patrz tabela 4.1), które pozwolą bez przeszkód przygotować stanowisko.

Tabela 4.1. Lista elementów do konfiguracji

Lp.	Element
1	Zasilacz USB C 5V 3A
2	Przewód HDMI - mikro HDMI
3	Monitor
4	Myszka na USB
5	Klawiatura na USB
6	Karta microSD(min 16Gb) z adapterem
7	Przewód Ethernet (opcjonalnie)

Używając strony producenta mikrokontrolera [23] należy pobrać instalator lub program Raspberry Pi Imager (patrz rys. 4.1) przygotowujący gotową partycję boot odpowiadającą za uruchamianie systemu.



Rys. 4.1 Raspberry Pi Imager

Zalecana do tego projektu jest pełna wersja systemu OS (ang. *Operating System*) (sugerowana przez producenta sprzętu). Podczas testu w pierwszej kolejności zainstalowany był system Raspberry PI OS Lite, ale powstały problemy z instalacją programów oraz łączeniem z siecią. Pełna wersja, pomimo większego obciążenia procesora spowodowanego GUI (ang. *Graphical User Interface*), pozwala na bezproblemową instalację oprogramowania monitorującego. Mając wgrany system na karcie pamięci należy stworzyć dwa pliki w partycji boot: "ssh" (bez rozszerzenia) oraz "wpa_supplicant.conf". W pliku "wpa_supplicant.conf" należy wprowadzić dane (patrz rys. 4.2). Posłużą one do konfiguracji połączenia z siecią Wi-Fi. Ten krok jest szczególnie ważny dla użytkowników wersji Lite, ponieważ wersja pełna umożliwia łatwą i intuicyjną konfigurację sieci.

```
wpa_supplicant.conf x
1 country=PL
2 ctrl_interface=DIR=/var/run/wpa_supplicant GROUP=netdev
3 update_config=1
4 network={
5     ssid="NazwaSieci"
6     psk="HasloDoSieci"
7 }
```

Rys. 4.2. Plik wpa_supplicant.conf

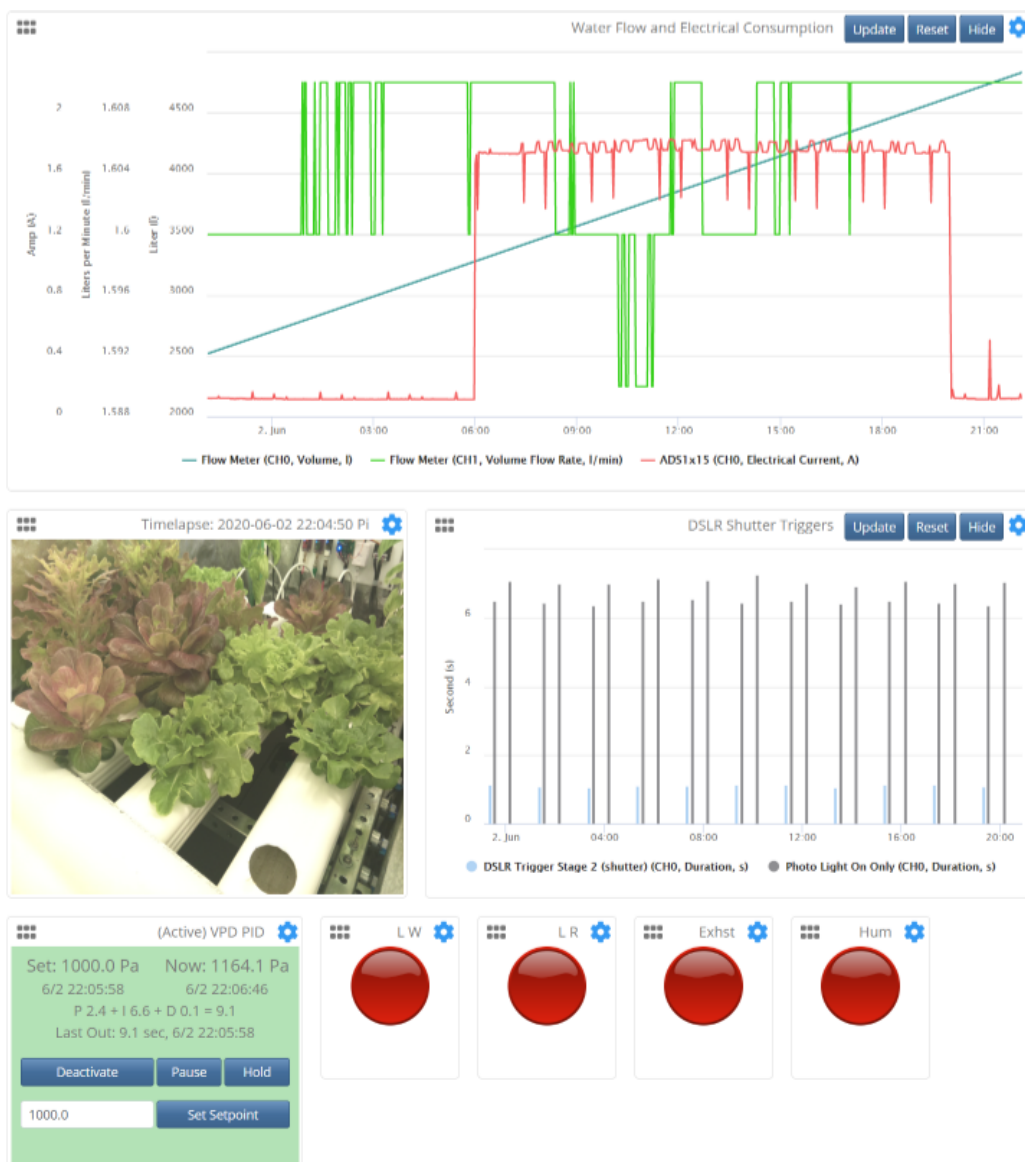
Po wprowadzeniu plików należy przełożyć kartę do mikrokontrolera, podłączyć klawiaturę, myszkę, monitor oraz opcjonalnie przewód ethernet. Na koniec podłączane jest zasilanie przy pomocy przewodu USB C (patrz rys. 4.3). Po uruchomieniu należy ustawić strefę czasową, język oraz połączyć się z siecią. Używając komendy “ip a” w terminalu w należy wydobyć adres IP urządzenia. Pojawi się obok napisu ‘inet’ i należy go zapisać, ponieważ będzie przydatny w dalszej części przygotowywania urządzenia.



Rys. 4.3. Połączenie przewodów do Raspberry Pi

4.2. Mycodo

Mycodo jest systemem monitorowania i akwizycji danych podobnym do profesjonalnych rozwiązań ale przygotowanym pod możliwości urządzeń Raspberry Pi. Jest to oprogramowanie w pełni otwarte, a kod źródłowy jest dostępny dla wszystkich na githubie twórcy [24]. Jest to system, który przechowuje dane w bazie danych InfluxDB opartej na szeregach czasowych (ang. *time series data*). Dane pochodzą z obserwowanych wejść do których przypisane są modele czujników, dzięki czemu możliwa jest łatwa konfiguracja i wizualizacja danych dla użytkowników (patrz rys. 4.3).



Rys. 4.4. Przykładowy system w Mycodo [24]

Część obliczeniowa systemu posiada wiele funkcji [25] takich jak:

- odczytywanie wartości z czujników,
- modulowane wyjścia (generowanie PWM, przełączanie przekaźników, obsługa MQTT (ang. *MQ Telemetry Transport*) itp.),
- regulacja przy pomocy PID,
- harmonogramy,
- nagrywanie i nadawanie na żywo obrazu,
- sterowanie wielokryterialne.

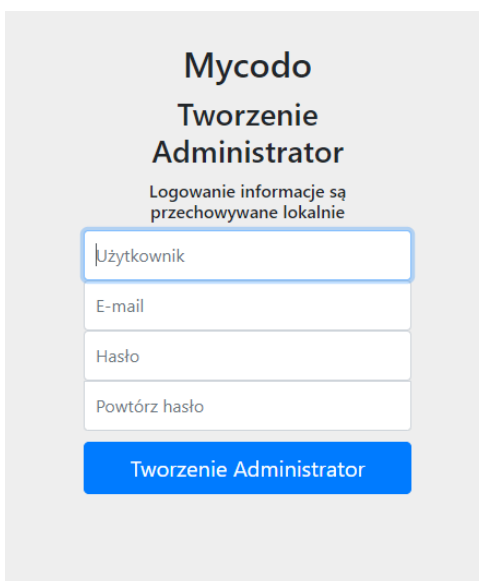
4.3. Konfiguracja platformy

Rozpoczęcie konfiguracji oprogramowania Mycodo należy rozpocząć od instalacji przy pomocy komendy w terminalu (patrz rys. 4.4).

```
curl -L https://kizniche.github.io/Mycodo/install | bash
```

Rys. 4.5. Komenda instalacyjna [24]

Proces ten zajmuje od 30 do 60 minut. W przypadku poprawnego zakończenia instalacji, będzie możliwość stworzenia administratora (patrz rys. 4.5). W tym celu można użyć innego urządzenia podłączonego do tej samej sieci poprzez wpisanie w wyszukiwarce "https://IP_Raspberry_Pi/".



Rys. 4.6. Tworzenie administratora

Następnie trzeba dodać urządzenia wejścia i wyjścia z listy do systemu (patrz. 4.6). W przypadku, gdy nie ma konkretnego modelu warto wybrać najbardziej podobny z tym samym sposobem komunikacji.

Wejście [?](#)

Texas Instruments: ADS1*	Dodaj			
		CPU/GPU Temperature	CPU/GPU Temperature	Dezaktywuj
		Atlas pH	Atlas pH	Dezaktywuj
		Atlas PT-1000	Atlas PT-1000	Dezaktywuj
		Atlas Flow Meter	Atlas Flow Meter	Dezaktywuj
		Atlas EC	Atlas EC	Dezaktywuj
		GPIO State	GPIO State	Dezaktywuj
		ADS1115	ADS1115	Dezaktywuj

Rys. 4.7. Wprowadzone do systemu wejścia

Każde z wejść można dodatkowo dostosować przy pomocy ustawień (patrz 4.7).

25.0	Set Temperature Compensation		
jasne Kalibracja			
Mid Point pH			
7.0	Calibrate Mid		
Low Point pH			
4.0	Calibrate Low		
High Point pH			
10.0	Calibrate High		
The I2C address can be changed. Enter a new address in the 0xYY format (e.g. 0x22, 0x50), then press Set I2C Address. Remember to deactivate and change the I2C address option after setting the new address.			
Nowy adres I2C			
0x63	Zestaw I2C Adres		
Opcje			
Nazwa	Log Level: Debug Interfejs	Adres I2C	Magistrala I2C
Atlas pH	<input type="checkbox"/> I2C	0x63	1
Cykl (sekundy)	Wyjście przed pomiarowe	Pre Out Duration	Odwróć operacje przedpomiarową
15,0	Disabled	10,0	<input checked="" type="checkbox"/>
Opcje niestandardowe			
Temperature Compensation: Pomiary	Temperature Compensation: Max Age		
Wybierz jeden	120		
Ustawienia pomiarowe			
Configure Measurements			

Rys. 4.8. Kalibracja czujnika w środowisku Mycodo

Na tym etapie konfiguracji istnieje możliwość konfiguracji urządzeń przy pomocy dokładnie pomierzonych warunków pracy. Proces ten zostanie opisany na podstawie czujnika pH. Metodyka przygotowywania następných urządzeń jest identyczna, a instrukcja dobrze opisuje znaczenie każdego parametru.

Kompensacja temperaturowa względem temperatury wzorcowej, podanej przez producenta, pozwala skonfigurować pomiary na podstawie dokumentacji producenta wybranego czujnika w celu zapewnienia wyższych dokładności. W tym przypadku wartością wzorcową jest 25°C.

Następnie podawane są zakresy pH w jakich będzie pracował czujnik. W celu podania dokładnych wartości należy wsadzić go do zbiorników z wzornikami odczynników pH (4, 7 i 10) i wpisać w odpowiednie miejsca pomiary. Pozwala to zniwelować niedokładności pomiarowe związane z odchyleniami odczytów czujnika. Jest to programowa kompensacja niedoskonałości fizycznego urządzenia.

Nadanie unikalnego adresu w formacie 0xYY, gdzie Y jest cyfrą jest standardowym elementem komunikacji I2C. Dzięki niepowtarzalnym adresom czujnik i mikrokontroler wiedzą jak wywołać odpowiednie urządzenie do przekazania danych.

W części opisanej "Opcje" można zmienić nazwę urządzenia, adres I2C oraz magistralę, ponieważ program może obsługiwać więcej niż jedną magistralę I2C. Można wybrać okres z jakim wykonywane są pomiary. Domyślnie ustawiony czas cyklu to 15 sekund. System pozwala również wykonywać określone akcje przed wykonaniem samego pomiaru. Akcjami możliwymi jest włączenie określonego wyjścia. Ma to znaczenie, gdy ważne jest uzyskanie najświeższego pomiaru. Można również ustawić czas włączenia wyjścia.

Istnieją również opcje niestandardowe, które pozwalają wybrać urządzenie będące wzorcem dla temperatury kompensacji. Można tu wybrać czujnik temperatury co dynamicznie będzie regulować temperaturę wzorcową. Wybrany w tym celu zostanie czujnik temperatury cieczy jako dynamiczny punkt odniesienia.

Ze skonfigurowanymi urządzeniami wejścia i wyjścia należy wytworzyć funkcje, które umożliwią sterowanie, wyświetlanie oraz załączanie warunkowe konkretnych akcji. Trzema najważniejszymi funkcjami są regulacja pH, elektroprowadnictwa oraz funkcja bezpieczeństwa przełączająca pompy.

System umożliwia wiele metod regulowania parametrów pH i elektroprowadnictwa. Najlepsze sposoby to PID z możliwościami doboru nastaw, PID autotune oraz wielowarunkowy kod utrzymujący wartości parametrów w zalecanych zakresach. Regulacja tych parametrów nie wymaga dużej dokładności, ponieważ zakresy akceptowalne są dość duże, gdyż rośliny nie wymagają bardzo dokładnych parametrów wody, aby przeżyć i rosnąć efektywnie. W przypadku użycia regulatora PID na rzeczywistym obiekcie, do strojenia nastaw należałoby użyć np. pierwszej metody Zieglera-Nicholsa opartej na analizie odpowiedzi skokowej co bez fizycznego systemu jest niemożliwe. Wyniki bez przeprowadzenia tego procesu byłyby akceptowalne, ale mogły by być niezadowolające. Dlatego jako



pierwsze ustawienie zostanie wybrany kod w języku python 3 utrzymujący parametry w zalecanych zakresach (patrz rys. 4.8).

```
import time

#zalecane zakresy parametrów
EcZakresGora = 1300
EcZakresDol = 1000
pHZakresDol = 5.5
pHZakresGora = 6.2
pHZakresKrytycznyGora = 7.0
pHZakresKrytycznyDol = 5.0

#pomiary
PomiarEc = self.condition("{762d2b57}")#czujnik Ec
PomiarPH = self.condition("{04b00ae8}")#czujnik pH

#sprawdzanie wartosci krytycznych
if PomiarPH < pHZakresKrytycznyDol:
    self.run_action("{7ad4f6bf}")#pompa2

elif PomiarPH > pHZakresKrytycznyGora:
    self.run_action("{c438692a}")#pompa1

# regulacja parametrow
elif PomiarEc < EcZakresDol:
    self.run_action("{6ccf55f3}")#pompa3
    self.run_action("{e9d361e8}")#pompa4

elif PomiarPH < pHZakresDol:
    self.run_action("{7ad4f6bf}")#pompa2

elif PomiarPH > pHZakresGora:
    self.run_action("{c438692a}")#pompa1
```

Rys. 4.9. Kod regulujący pH i przewodność

Tworząc ten kod i wprowadzając go do systemu generuje on z niego klasę i uzupełnia o wymagane własne biblioteki dla poprawnego działania. Zmienne zostały sformatowane przy pomocy CamelCase, aby ujednolicić całość programu. Ten sposób zapisywania pozwala na zwiększenie czytelności poprzez rozpoczynanie każdego słowa wielką literą bez spacji. Jest to praktyka używana przez programistów, aby osoby poza twórcą mogły zapoznać się łatwiej z logiką programu.

Napisany kod na początku zawiera zdefiniowane wartości parametrów, które odpowiadają granicznym akceptowalnym wartościom. W następnych liniach deklarowane są zmienne, które przechowują wartości wykonanych pomiarów. Funkcja self.condition jest zwrotem wartości pomiaru ze wskazanego ID urządzenia. Przy każdej linii używającej ID zamieszczony jest komentarz wskazujący urządzenie z systemu w czytelnej formie. Sprawdzanie wartości krytycznych jest ważne, ponieważ zbyt duża kwasowość lub zasadowość może doprowadzić do śmierci rośliny. Pompa dwa zaczyna podawać



roztwór zwiększający pH, a pompa 1 zmniejsza w przypadku za dużej wartości wskaźnika. Gdy wartości mieszczą się w bezpiecznych ramach regulowany jest poziom elektroprowadnictwa cieczy w zakresach od 1000 do 1300 μS (ang. "Micro Salinity" - mikro zasolenie). Wartość podnoszona jest przy pomocy pomp 3 i 4, które jednocześnie są roztworami odżywczymi. Zmniejszenie tego parametru wymaga dolewania wody do zbiornika. Na koniec następuje ponowna regulacja pH do zakresu zalecanego. Sterowania pH są oddzielone, ponieważ zapewni to czas substancji na rozejście się po systemie po wyregulowaniu cieczy ze stanu krytycznego.

ID używane w programie zostały wygenerowane po zadeklarowaniu warunków i akcji w bločku funkcji (patrz rys. 4.9).

Name	Period (seconds)	Start Offset (seconds)	Timeout (seconds)	Log Level: Debug
pH i Ec regulacja	300,0	10,0	30,0	<input type="checkbox"/>
Message Includes Code <input type="checkbox"/>				
Conditions ?				
Select One <input type="button" value="Add Condition"/>				
Measurement		Max Age (seconds)		
<input type="button" value="Save"/>	<input type="button" value="Delete"/>	[Input 05 CH0] pH Ion Concentration (pH) <input type="button" value="v"/>	360	
Measurement (Single, Last): self.condition("{04b00ae8}") returns the last value found within the Max Age, otherwise returns None.				
Measurement		Max Age (seconds)		
<input type="button" value="Save"/>	<input type="button" value="Delete"/>	[Input 02 CH0] Elektroprowadnictwo Electrical Conductivity ($\mu\text{S}/\text{cm}$) <input type="button" value="v"/>	360	
Measurement (Single, Last): self.condition("{762d2b57}") returns the last value found within the Max Age, otherwise returns None.				
Actions ?				
Select One <input type="button" value="Add Action"/>				
Controller ID		State	Duration (seconds)	
<input type="button" value="Save"/>	<input type="button" value="Delete"/>	[Output 04 CH0] Pompa1 <input type="button" value="v"/>	On <input type="button" value="v"/>	1,0
Output (On/Off/Duration): Turn an Output Off, On, or On for a duration. self.run_action("{c438692a}") will execute only this action. self.run_all_actions() will execute all actions sequentially.				
Controller ID		State	Duration (seconds)	
<input type="button" value="Save"/>	<input type="button" value="Delete"/>	<input type="button" value="v"/>	<input type="button" value="v"/>	0,0
Output (On/Off/Duration): Turn an Output Off, On, or On for a duration. self.run_action("{2e35a3bc}") will execute only this action. self.run_all_actions() will execute all actions sequentially.				
Controller ID		State	Duration (seconds)	
<input type="button" value="Save"/>	<input type="button" value="Delete"/>	[Output 03 CH0] Pompa2 <input type="button" value="v"/>	On <input type="button" value="v"/>	1,0
Output (On/Off/Duration): Turn an Output Off, On, or On for a duration. self.run_action("{7ad4f6bf}") will execute only this action. self.run_all_actions() will execute all actions sequentially.				
Controller ID		State	Duration (seconds)	
<input type="button" value="Save"/>	<input type="button" value="Delete"/>	<input type="button" value="v"/>	<input type="button" value="v"/>	0,0
Output (On/Off/Duration): Turn an Output Off, On, or On for a duration. self.run_action("{6ccf55f3}") will execute only this action. self.run_all_actions() will execute all actions sequentially.				
Controller ID		State	Duration (seconds)	
<input type="button" value="Save"/>	<input type="button" value="Delete"/>	<input type="button" value="v"/>	<input type="button" value="v"/>	0,0
Output (On/Off/Duration): Turn an Output Off, On, or On for a duration. self.run_action("{e9d361e8}") will execute only this action. self.run_all_actions() will execute all actions sequentially.				

Rys. 4.10. Deklaracja urządzeń do funkcji

Następna funkcja będzie zapewniała bezpieczeństwo systemu i pomp wody transportujących ciecisz do systemu przelewowego (patrz rys. 4.10).

```
PomiarPrzeplywuSuma = self.condition("{3cdb3e39}")#czujnik przeplywu suma pomiarow
PomiarPrzeplywu = self.condition("{db92fb77}")#czujnik przeplywu ostatni pomiar
StanPoziomu = self.condition("{e4290ece}")#czujnik poziomu

#zapobieganie suchobiegowi
if StanPoziomu == 0: #brak wody
    self.run_action("{3dfce204}") #przekaznik 1 wylaczony
    self.run_action("{70893796}")#przekaznik 2 wylaczony

elif PomiarPrzeplyw < 0.7*(PomiarPrzeplywuSuma - PomiarPrzeplywu):
    self.run_action("{3dfce204}") #przekaznik 1 wylaczony
    self.run_action("{21eced14}") #zalaczenie drugiej pompy

elif self.condition("{b4ba677b}") == "off" and PomiarPrzeplyw < 0.7*(PomiarPrzeplywuSuma - PomiarPrzeplywu):
    self.run_action("{70893796}") #przekaznik 2 wylaczony
    self.run_action("{d7a35229}") #przekaznik 1 wlaczony
```

Rys. 4.11. Kod sterujący przełącznikami systemu zasilania

Kod zbiera aktualne i poprzednie pomiary czujnika przepływu, aby zweryfikować czy pompa spełnia swoją rolę i działa wydajnie. Następnie sprawdzany jest stan czujnika poziomu wody, który daje sygnał boolowski. W pierwszej kolejności kod zapobiega suchobiegowi. Jeśli nie ma wody w zbiorniku wyłączane są obie pompy poprzez wysterowanie przełączników. Następnie sprawdzane są przepływy. Jeśli aktualny pomiar przepływu spadł o 30% względem poprzedniego zmieniana jest działająca pompa na zastępczą. Sprawność pompy mogła spaść przez zapchanie jakimś drobnym elementem lub uszkodzeniem elektrycznym. W tym przypadku pomiar historycznej wartości jest realizowane poprzez różnicę sumy dwóch pomiarów i aktualnej wartości.

W celu łatwego podglądu parametrów będąc na miejscu zostaje skonfigurowany ekran LCD. Na ekranie będą wyświetlane 3 zestawy danych zmieniające się co 5 sekund. Lista danych wyświetlanych na ekranach zostanie określona w tabeli 4.2.

Każda z wyświetlanych danych jest ostatnim pomiarem lub ostatnim zaobserwowanym stanem urządzenia. Maksymalny historyczny stan obserwowany przez ekran to 6 minut. Odświeżanie wszystkich pomiarów jest częstsze, ale tak długi zapas czasowy daje dużo czasu na działanie i przełączanie czujników.

Tabela 4.2. Tabela parametrów na ekranach LCD

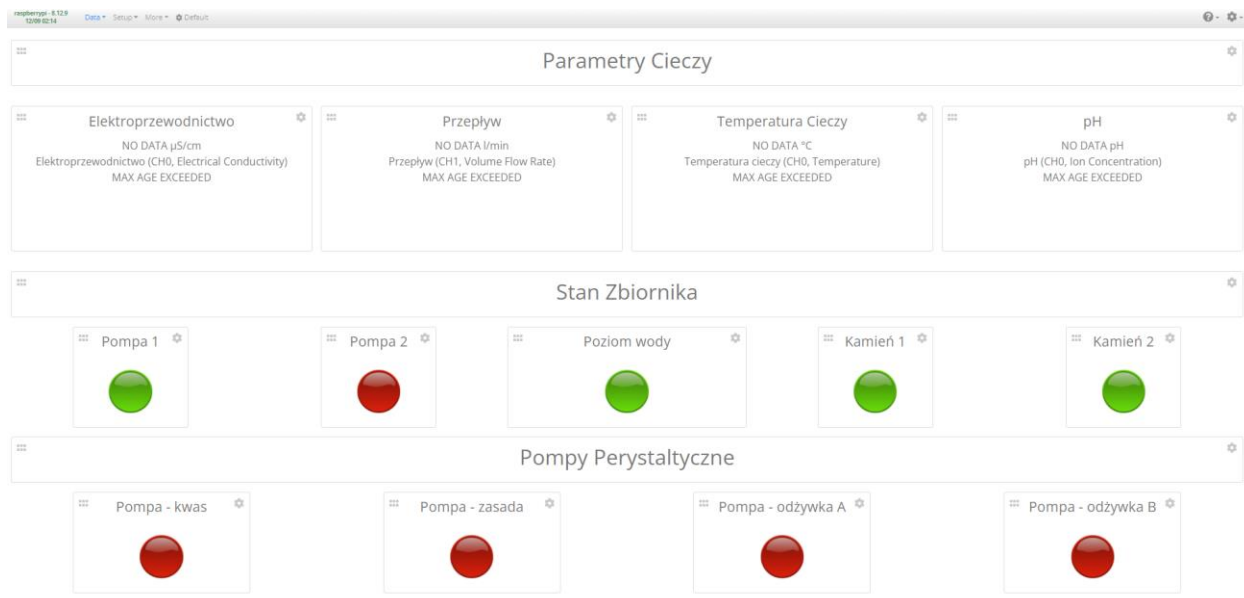
Numer	Parameter
Ekran 1	
1	Elektroprzewodnictwo [$\mu\text{S}/\text{cm}$]
2	pH
3	Temperatura cieczy [$^{\circ}\text{C}$]
4	Stan czujnika poziomu
Ekran 2	
1	Przepompowana objętość (Pompa perystaltyczna 1) [ml]
2	Przepompowana objętość (Pompa perystaltyczna 2) [ml]
3	Przepompowana objętość (Pompa perystaltyczna 3) [ml]
4	Przepompowana objętość (Pompa perystaltyczna 4) [ml]
Ekran 3	
1	Stan przekaźnika pompy 1
2	Stan przekaźnika pompy 2
3	Stan przekaźnika SSR 1
4	Stan przekaźnika SSR 2

Ekran 1 podzielone są tematycznie. Pierwszy ekran pokazuje najważniejsze parametry wody w zbiorniku. Poza poziomem wody operator nie jest w stanie samodzielnie określić bez użycia sprzętu. Drugi ekran skupia się na ilości przepompowanej wody w czasie 6ciu minut, co wraz z wzrokowym określeniem ilości substancji w zbiornikach pozwala estymować, kiedy wystąpi konieczność dolania



cieczy. Ostatni ekran informuje o stanie urządzeń znajdujących się pod wodą. Operator jest w stanie zaobserwować, że woda jest pompowana do rur, ale może nie wiedzieć, czy doszło do przełączenia urządzenia. Dzięki temu ekranowi wie, że wystąpił problem z domyślną pompą. To samo dotyczy kamieni napowietrzających. Ekran oraz ilość bąbli powietrza wydobywających się ze zbiornika pozwala określić czy działają oba urządzenia.

Mając przygotowany system należy przygotować panel monitorujący (patrz rys. 4.11), który stanowić będzie odpowiednik oprogramowania SCADA w tym systemie.



Rys. 4.12. Panel monitorujący na PC

Panel jest podzielony na trzy części, podobnie jak na ekranie LCD, ale ze względu na graficzne połączenie w pierwszej kolejności są pokazywane dokładne dane, które wyróżniają się wielkością, aby wygodnie monitorować parametry. Następne części to głównie indykatory, które informują binarnie o działaniu konkretnych urządzeń. Zielony indykator symbolizuje aktualne działanie elementu. Druga część skupia się na wyświetlaniu głównych urządzeń zbiornika zasilanych przez system zasilania. Znajduje się tam również indykator poziomu wody, ponieważ jest to kluczowa informacja, o części procesu, która nie została zautomatyzowana, a mianowicie wypełnianiu zbiornika wodą. O poziomie cieczy w zbiorniku zależy poprawne działanie pomp. Ostatnią częścią jest informowanie o działaniu pomp perystaltycznych.

5. PODSUMOWANIE

5.1. Podsumowanie projektu

W wyniku wykonanej pracy udało się zaprojektować systemy umożliwiające automatyczną uprawę roślin wewnątrz szklarni. Wszystkie systemy mogą być zdalnie monitorowane i zarządzane przy pomocy otwartego systemu SCADA na Raspberry Pi. Proces projektowania wymagał używania wielu narzędzi oraz efektywnego korzystania z środowisk inżynierskich i badawczych jak CAD, symulatory przepływu. Temat uprawy hydroponicznej nie jest tak dobrze przebadany jak tradycyjne metody hodowli, co wymusiło sięganie do wielu alternatywnych źródeł wiedzy wykraczających poza badania naukowe.

5.2. Możliwości rozwoju projektu

Projekt ma wiele możliwości dalszego rozwoju w każdej z omawianych części procesu projektowania. Najwięcej miejsca na usprawnienia przekładające się na efektywność systemu jest w części programowej. Oto kilka propozycji rozwoju na podstawie wniosków po zakończeniu pracy:

- Ustalenie stałego adresu IP dla Raspberry Pi w celu posiadania stałej domeny dla systemu monitorującego.
- Wprowadzenie systemów powiadomień dla użytkownika, który nie monitoruje stale systemu, np. poprzez maile.
- Zamknięcie elektroniki w małej skrzynce DIN, co zapewni ochronę przed warunkami atmosferycznymi i zapewni modułowość rozwiązania.
- Dobór najlepszych nastaw regulatora PID przy pomocy pierwszej metody Zieglera-Nicholsa na podstawie badań fizycznego obiektu.
- Stworzenie estymatora stanu stężeń substancji w zbiorniku głównym w celu poprawy dokładności sterowania.

WYKAZ LITERATURY

- [01] Wikipedia - Układ regulacji (automatyka)
[https://pl.wikipedia.org/wiki/Uk%C5%82ad_regulacji_\(automatyka\)](https://pl.wikipedia.org/wiki/Uk%C5%82ad_regulacji_(automatyka)), odtworzono 20.01.2021
- [02] YoungWonks - Raspberry Pi 4 pinout <https://www.youngwonks.com/blog/Raspberry-Pi-4-Pinout>, odtworzono 2.11.2021
- [03] Margolis M., Jepson B., Weldin N. R., "Komunikacja za pomocą I2C i SPI" w *Arduino Przepisy na rozpoczęcie, rozszerzenie i udoskonalenie projektów*, wyd. 3, Gliwice: Helion, 2021, s. 457-486 (tłumaczenie)
- [04] Electronoobs, *PROTOCOLS: UART-I2C-SPI-Serial communications #001*, 11.08.2019 [dostęp 7.11.2021]. Dostępne w YouTube: <https://youtu.be/IyGwvGzrqp8>
- [05] Howtomechatronics - How I2C Communication Works? Arduino and I2C Tutorial
<https://howtomechatronics.com/tutorials/arduino/how-i2c-communication-works-and-how-to-use-it-with-arduino/>, odtworzono 7.11.2021
- [06] Gay W., *Raspberry Pi: Hardware Reference*, Berkeley: Apress, 2014
- [07] Horowitz P., Hill W., "The Art of Electronics", wyd. 3, New York: Cambridge University Press, 2015
- [08] Pratt C., "Encyklopedia elementów elektronicznych", Tom 1, Gliwice: Helion, 2021
- [09] Pratt C., "Encyklopedia elementów elektronicznych", Tom 2, Gliwice: Helion, 2021
- [10] Benton Jones J., Jr. , "Complete Guide for Growing Plants Hydroponically", Boca Raton: CRC Press, 2014
- [11] Al Tawaha A. R. M., Al-Karaki G., Al Tawaha A. R. M., Sirajuddin S. N., Makhadmeh I., Wahab P. E. M., Alsultan W., Massadeh A., "Effect of water flow rate on quality and quantity of lettuce (*Lactuca sativa L.*) in nutrient film technique (NFT) under hydroponics conditions", Sofia: Bulgarian Journal of Agricultural Science, 2018
- [12] Grigas A., Kemzūraitė A., Steponavičius D., Steponavičienė A., Domeika R., "Impact of Slope of Growing Trays on Productivity of Wheat green Fodder by a Nutrient Film Technique System", Kaunas: Institute of Agricultural Engineering and Safety, 2020
- [13] Valdez J., Becker J. , "Understanding the I2C Bus", Texas Instruments, 2015,
https://www.ti.com/lit/an/slva704/slva704.pdf?ts=1637568236827&ref_url=https%253A%252F%252Fwww.google.com%252F [dostęp 22.11.2021]
- [14] Houde F., "Why, When, and How to use I2C Buffers", Texas Instruments, 2018
https://www.ti.com/lit/an/scpa054/scpa054.pdf?ts=1637580618116&ref_url=https%253A%252F%252Fwww.google.com.ua%252F [dostęp 22.11.2021]
- [15] Alldatasheet.com "G3MB-202P Datasheet", OMRON Electronics LLC,
<https://pdf1.alldatasheet.com/datasheet-pdf/view/333500/OMRON/G3MB-202P.html> [dostęp 24.11.2021]



- [16] Botland - Moduł przekaźników Iduino 2 kanały z optoizolacją -styki 10A/250VAC - cewka 5V
https://botland.com.pl/moduly-przekaznikow/14266-modul-przekaznikow-iduino-2-kanal-y-z-optoizolacja-styki-10a250vac-cewka-5v-5903351242332.html?gclid=CjwKCAiAv_KMBhAzEiwAs-rX1DkpK3LmLWnQcxFW7a8ZNh1JP5RkvLB0ylqEUFfn9PLCiE_mYAeqzRoC-KUQAvD_BwE&sskey=074f1ea4f5ed47709b6816bdcb47cded [dostęp 24.11.2021]
- [17] Pickeringtest - Calculating MTBF Figures for Switching Systems
<https://www.pickeringtest.com/kb/hardware-topics/relay-reliability/calculating-mtbf-figures-for-a-switching-system> [dostęp 24.11.2021]
- [18] Kosmowski K, Materiały pomocnicze do zajęć z przedmiotu Niezawodność i Diagnostyka, Wydział Elektrotechniki i Automatyki, Politechnika Gdańska, 2020
- [19] Jaskulski A., "Autodesk Inventor Professional 2021 PL/ 2021+ / Fusion 360. Metodyka projektowania" . Gliwice, Helion, 2020
- [20] Ultimaker -Ultimaker Cura
<https://ultimaker.com/software/ultimaker-cura>, [dostęp 3.12.2021]
- [21] Jaskulski A., "Autodesk Inventor 2020 PL/2020+. Podstawy Metodyki Projektowania", Warszawa, PWN, 2019
- [22] L. Morgana "Hydroponic Lettuce Production: A Comprehensive, Practical and Scientific Guide to Commercial Hydroponic Lettuce Production", Casper Publications Pty Ltd, Melbourne,1999
- [23] Raspberry Pi - Documentation
<https://www.raspberrypi.com/documentation/> [dostęp 7.12.2021]
- [24] Github kizniche - Mycodo
<https://github.com/kizniche/Mycodo#install-mycodo> [dostęp 7.12.2021]
- [25] Github kizniche -Mycodo manual
<https://kizniche.github.io/Mycodo/mycodo-manual.pdf> [dostęp 7.12.2021]
- [26] Wikipedia - Hydroponics
<https://en.wikipedia.org/wiki/Hydroponics> [dostęp 10.12.2021]
- [27] Fresh Water System - What are Hydroponic Systems and How do They Work
<https://www.freshwatersystems.com/blogs/blog/what-are-hydroponic-systems> [dostęp 10.12.2021]

WYKAZ RYSUNKÓW

1.1. System hydroponiczny.....	7
2.1. Poglądowy przekrój pionowy szklarni wzdłuż szerokości.....	10
2.2. Rura do systemu przepływowego.....	11
2.3. Łącznik rur oraz montaż.....	12
2.4. Kolano 90° oraz trójnik.....	12
2.5. Złożenie obu stron piętra.....	13
2.6. Układu przepływowy bez podpór.....	13
2.7. Ideowy rzut boczny układu przepływowego bez podpór.....	14
2.8. Reprezentacja odchyłu rur poziomym.....	14
2.9. Elementy stelażu utrzymującego układ przepływowy.....	15
2.10. Gotowa konstrukcja przepływowa.....	16
3.1. Klasyczny układ regulacji.....	17
3.2. Zbiornik wody.....	18
3.3. Rozkład GPIO w mikrokontrolerze Raspberry PI 4 B.....	19
3.4. Struktura transmisji UART.....	21
3.5. Podłączenie urządzeń przez UART.....	21
3.6. Struktura komunikacji protokołu I2C.....	23
3.7. Schemat połączeń magistrali I2C.....	23
3.8. Struktura komunikacji protokołu SPI.....	25
3.9. Schemat połączeń układu magistrali SPI dla dwóch urządzeń podrzędnych.....	25
3.10. Poglądowy schemat sektora centralnego układu sterowania.....	27
3.11. Moduł przekaźnikowy mechaniczny.....	28
3.12. Moduł przekaźnikowy półprzewodnikowy.....	29
3.13. Schemat połączeń skrzynki kontroli zasilania.....	30
3.14. Pojemnik pomiarowy.....	31
3.15. Pokrywa pojemnika.....	32
3.16. Środowisko Ultimaker Cura 4.12.0.....	32



3.17.Ultimaker Cura - cięcie modelu.....	33
3.18.Uproszczony model czujnika.....	34
3.19.Model miernika przepływu.....	34
3.20.Złożenie pojemnika pomiarowego.....	35
3.21.Umiejszczenie pojemnika pomiarowego w zbiorniku głównym.....	35
3.22.Schemat połączeń podsystemu pomiarowego.....	36
3.23.Rura i jej złączka.....	37
3.24.Elementy przekładni planetarnej.....	37
3.25.Silnik napędzający pompę.....	38
3.26.Złożenie przekładni.....	38
3.27.Pompa perystaltyczna.....	39
3.28.Cała struktura pompująca.....	40
3.29.Schemat połączeń podsystemu dowartościowywanie wody.....	41
3.30.Przykładowa struktura równoległa.....	42
3.31.Błąd względny od niezawodności.....	45
4.1.Raspberry Pi Imager.....	48
4.2.Plik wpa_supplicant.conf.....	48
4.3.Połączenie przewodów do Raspberry Pi.....	49
4.4.Przykładowy system w Mycodo.....	50
4.5.Komenda instalacyjna.....	51
4.6.Tworzenie administratora.....	51
4.7.Wprowadzone do systemu wejścia.....	52
4.8.Kalibracja czujnika w środowisku Mycodo.....	52
4.9.Kod regulujący pH i elektroprzewodnictwo.....	54
4.10.Deklaracja urządzeń do funkcji.....	55
4.11.Kod sterujący przekaźnikami systemu zasialnia.....	56
4.12.Panel monitorujący na PC.....	58



WYKAZ TABEL

3.1. Podsumowanie cech protokołów komunikacyjnych.....	26
3.2. Parametry modułów przekaźnikowych.....	29
3.3. Parametry wydruku drukarki 3D.....	33
3.4. Parametry pompy perystaltycznej.....	40
3.5. Zależność błędu względnego od parametru niezawodności.....	44
4.1. Lista elementów do konfiguracji.....	47
4.2. Tabela parametrów na ekranach LCD.....	57

DODATEK A: Klasy wygenerowane przez program Mycodo (płyta CD)

Full Conditional Statement code:

```
1: import os
2: import sys
3: sys.path.append(os.path.abspath('/var/mycodo-root'))
4: from mycodo.controllers.base_conditional import AbstractConditional
5: from mycodo.mycodo_client import DaemonControl
6: control = DaemonControl(pyro_timeout=30.0)
7:
8: class ConditionalRun(AbstractConditional):
9:     def __init__(self, logger, function_id, message):
10:         super(ConditionalRun, self).__init__(logger, function_id, message, timeout=30.0)
11:
12:         self.logger = logger
13:         self.function_id = function_id
14:         self.variables = {}
15:         self.message = message
16:         self.running = True
17:
18:     def conditional_code_run(self):
19:         import time
20:
21:         #zalecane zakresy parametrów
22:         EcZakresGora = 1300
23:         EcZakresDol = 1000
24:         pHZakresDol = 5.5
25:         pHZakresGora = 6.2
26:         pHZakresKrytycznyGora = 7.0
27:         pHZakresKrytycznyDol = 5.0
28:
29:         #pomiar
30:         PomiarEc = self.condition("762d2b57-a463-4dad-9e9d-d43a47ee5332")#czujnik Ec
31:         PomiarPH = self.condition("04b00ae8-bdb7-441b-a960-3c8c01519ea6")#czujnik pH
32:
33:         #sprawdzanie wartosci krytycznych
34:         if PomiarPH < pHZakresKrytycznyDol:
35:             self.run_action("7ad4f6bf-6a94-4f33-a8d7-8ceae38f1b2b")#pompa2
36:
37:         elif PomiarPH > pHZakresKrytycznyGora:
38:             self.run_action("c438692a-5649-4649-acff-a63c5ffe6154")#pompa1
39:
40:         # regulacja parametrow
41:         elif PomiarEc < EcZakresDol:
42:             self.run_action("6ccf55f3-4888-409d-8d17-af40fc52d1f1")#pompa3
43:             self.run_action("e9d361e8-81e5-4c6f-b10f-efd6a3a7ee93")#pompa4
44:
45:         elif PomiarPH < pHZakresDol:
46:             self.run_action("7ad4f6bf-6a94-4f33-a8d7-8ceae38f1b2b")#pompa2
47:
48:         elif PomiarPH > pHZakresGora:
49:             self.run_action("c438692a-5649-4649-acff-a63c5ffe6154")#pompa1
50:
51:     def function_status(self):
52:         # Example code to provide a return status for other controllers and widgets.
53:         status_dict = {
54:             'string_status': "This is the demo status of the conditional controller. "
55:                             "The controller has looped {} times".format(self.loop_count),
56:             'loop_count': self.loop_count,
57:             'error': []
58:         }
59:         return status_dict
```

Conditional Statement code analysis:

Rys. A.1. Klasa wygenerowana do funkcji sterującej poziomem składników w zbiorniku



```

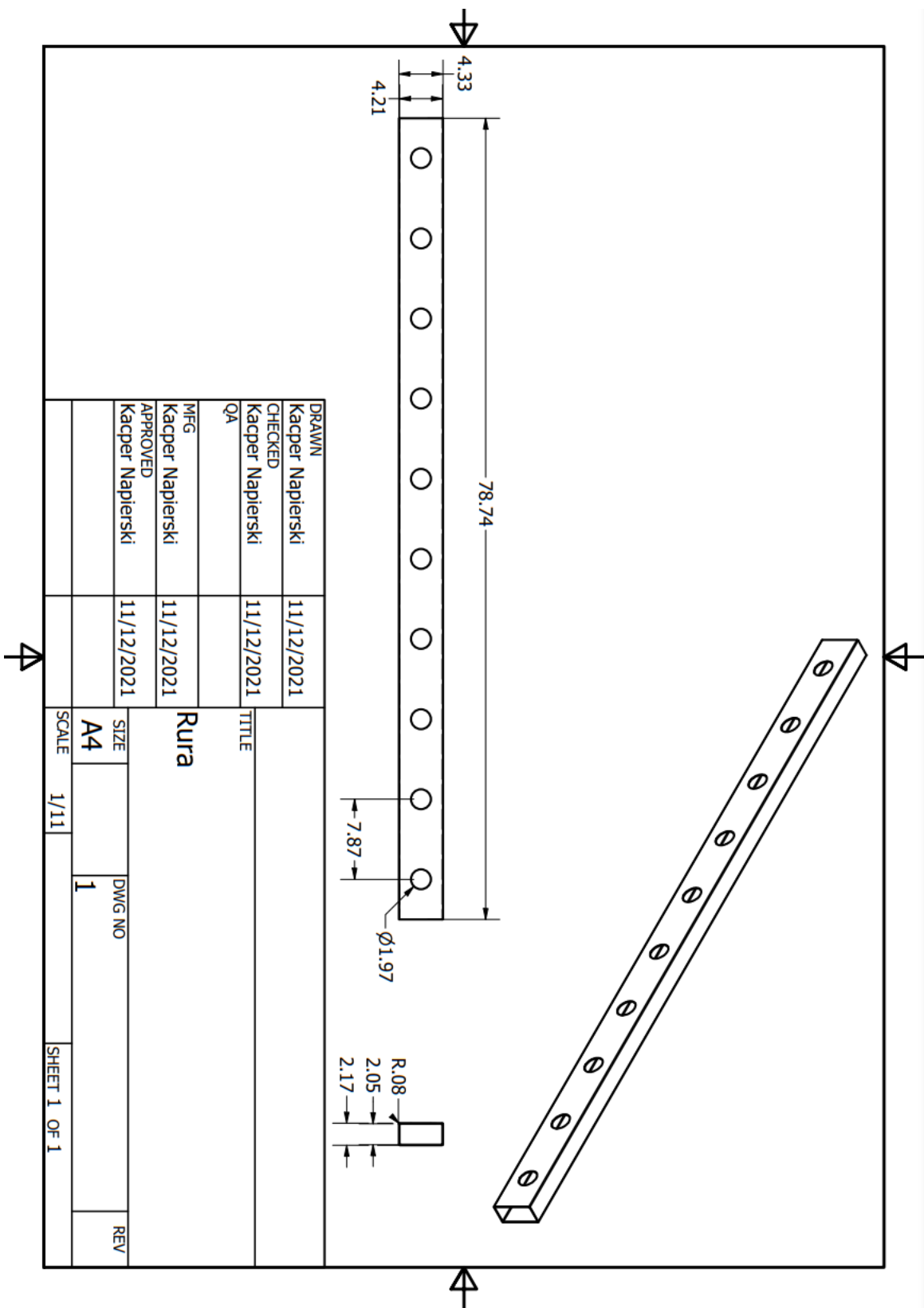
1 Full Conditional Statement code:
2
3 1: import os
4 2: import sys
5 3: sys.path.append(os.path.abspath('/var/mycodo-root'))
6 4: from mycodo.controllers.base_conditional import AbstractConditional
7 5: from mycodo.mycodo_client import DaemonControl
8 6: control = DaemonControl(pyro_timeout=80.0)
9 7:
10 8: class ConditionalRun(AbstractConditional):
11 9:     def __init__(self, logger, function_id, message):
12 10:         super(ConditionalRun, self).__init__(logger, function_id, message, timeout=80.0)
13 11:
14 12:         self.logger = logger
15 13:         self.function_id = function_id
16 14:         self.variables = {}
17 15:         self.message = message
18 16:         self.running = True
19 17:
20 18:     def conditional_code_run(self):
21 19:         PomiarPrzeplywuSuma = self.condition("3cdb3e39-2508-4fc0-98db-1f2b243e2b0a") #czujnik przeplywu suma pomiarow
22 20:         PomiarPrzeplywu = self.condition("db92fb77-2452-4a07-8ca4-c6afc79963d3") #czujnik przeplywu ostatni pomiar
23 21:         StanPoziomu = self.condition("e4290ece-3c01-4d79-a2b3-eb88eb7dfa7a") #czujnik poziomu
24 22:
25 23:         #zapobieganie suchobiegowi
26 24:         if StanPoziomu == 0: #brak wody
27 25:             self.run_action("3dfce204-a738-49a2-853d-e1168c933926") #przekaznik 1 wylaczony
28 26:             self.run_action("70893796-dblc-4e73-abea-49eef656671d") #przekaznik 2 wylaczony
29 27:
30 28:         elif PomiarPrzeplyw < 0.7*(PomiarPrzeplywuSuma - PomiarPrzeplywu):
31 29:             self.run_action("3dfce204-a738-49a2-853d-e1168c933926") #przekaznik 1 wylaczony
32 30:             self.run_action("21eced14-0fdf-4d20-b23a-clb4d3ea9cd3") #zalaczenie drugiej pompy
33 31:
34 32:         elif self.condition("b4ba677b-4e14-4212-9adf-a430a58aclba") == "off" and PomiarPrzeplyw < 0.7*(PomiarPrzeplywuSuma - PomiarPrzeplywu):
35 33:             self.run_action("70893796-dblc-4e73-abea-49eef656671d") #przekaznik 2 wylaczony
36 34:             self.run_action("d7a35229-a6d3-42b4-9375-3d792d589792") #przekaznik 1 wlaczony
37 35:
38 36:     def function_status(self):
39 37:         # Example code to provide a return status for other controllers and widgets.
40 38:         status_dict = {
41 39:             'string_status': "This is the demo status of the conditional controller. "
42 40:                             "The controller has looped {} times".format(self.loop_count),
43 41:             'loop_count': self.loop_count,
44 42:             'error': []
45 43:         }
46 44:         return status_dict

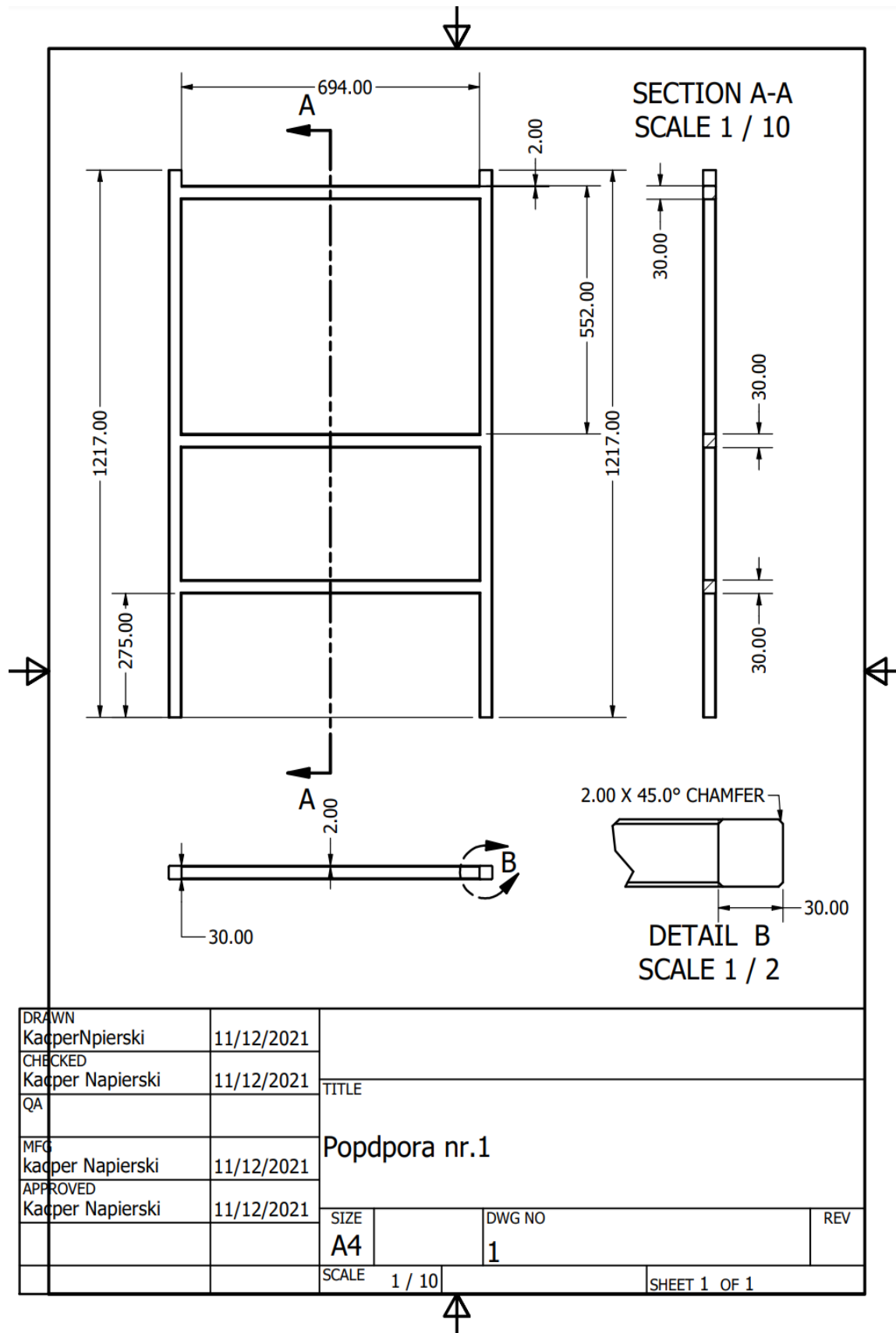
```

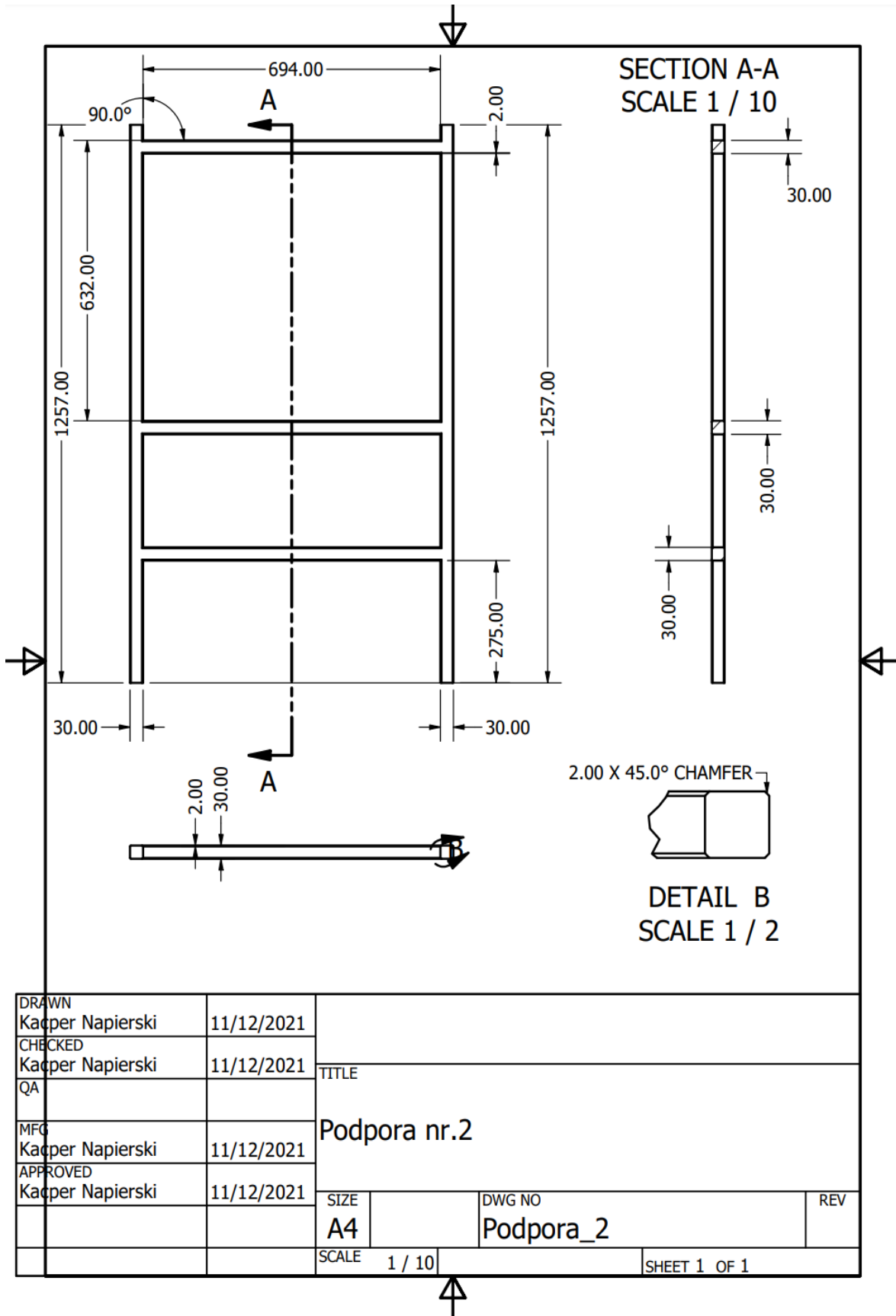
Rys.A.2. Klasa wygenerowana do funkcji zabezpieczenia pomp



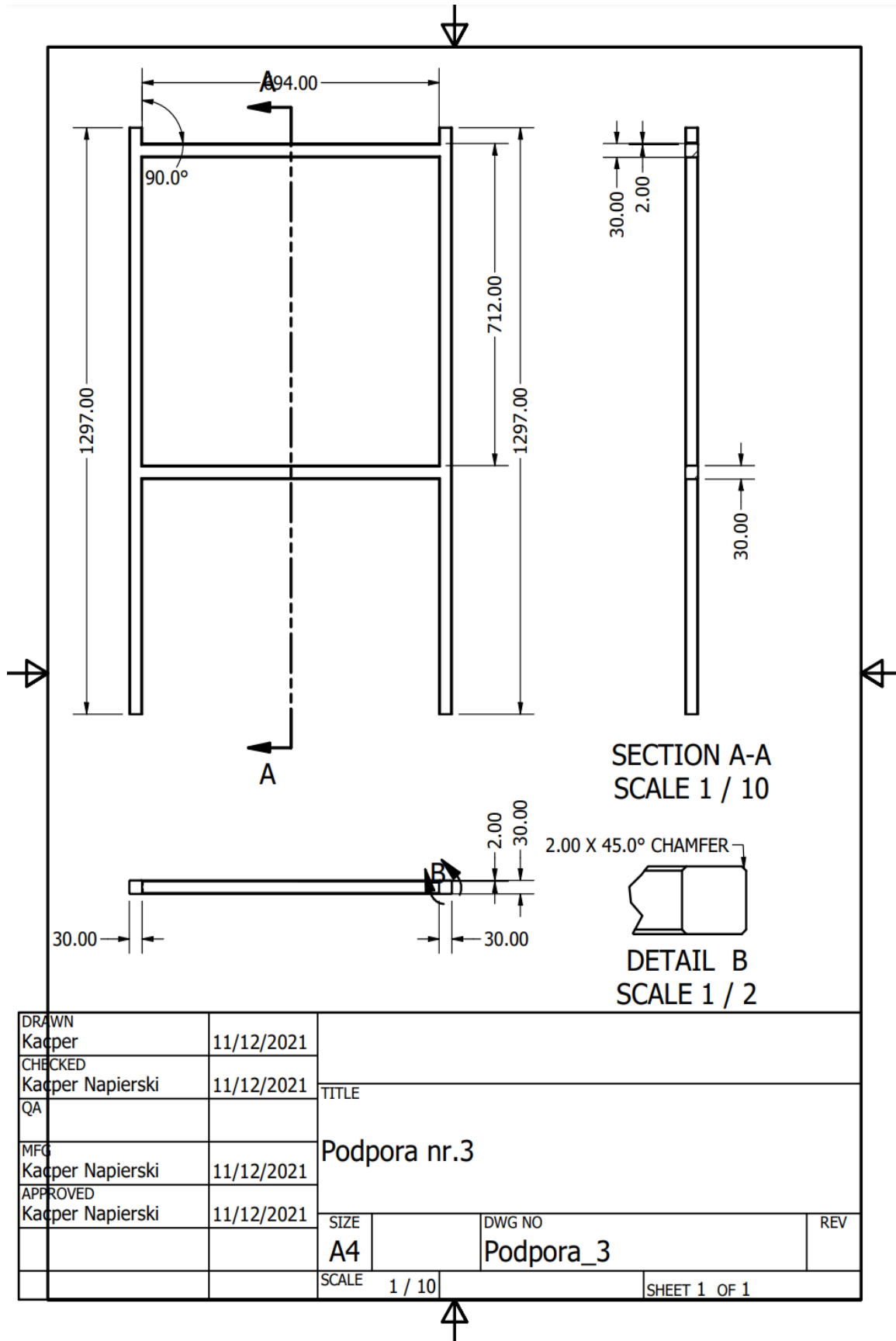
DODATEK B: Rysunki techniczne (płyta CD)

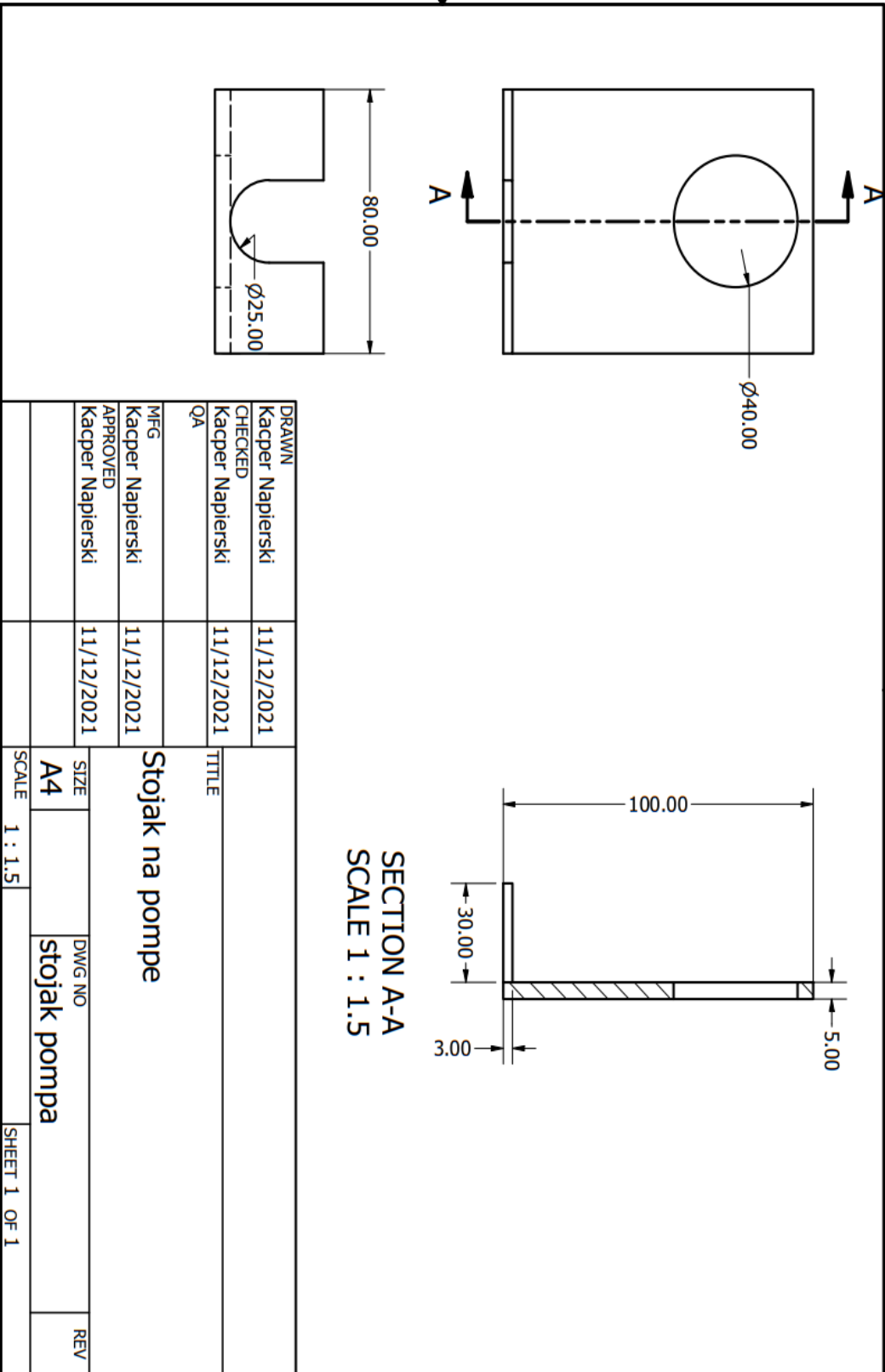






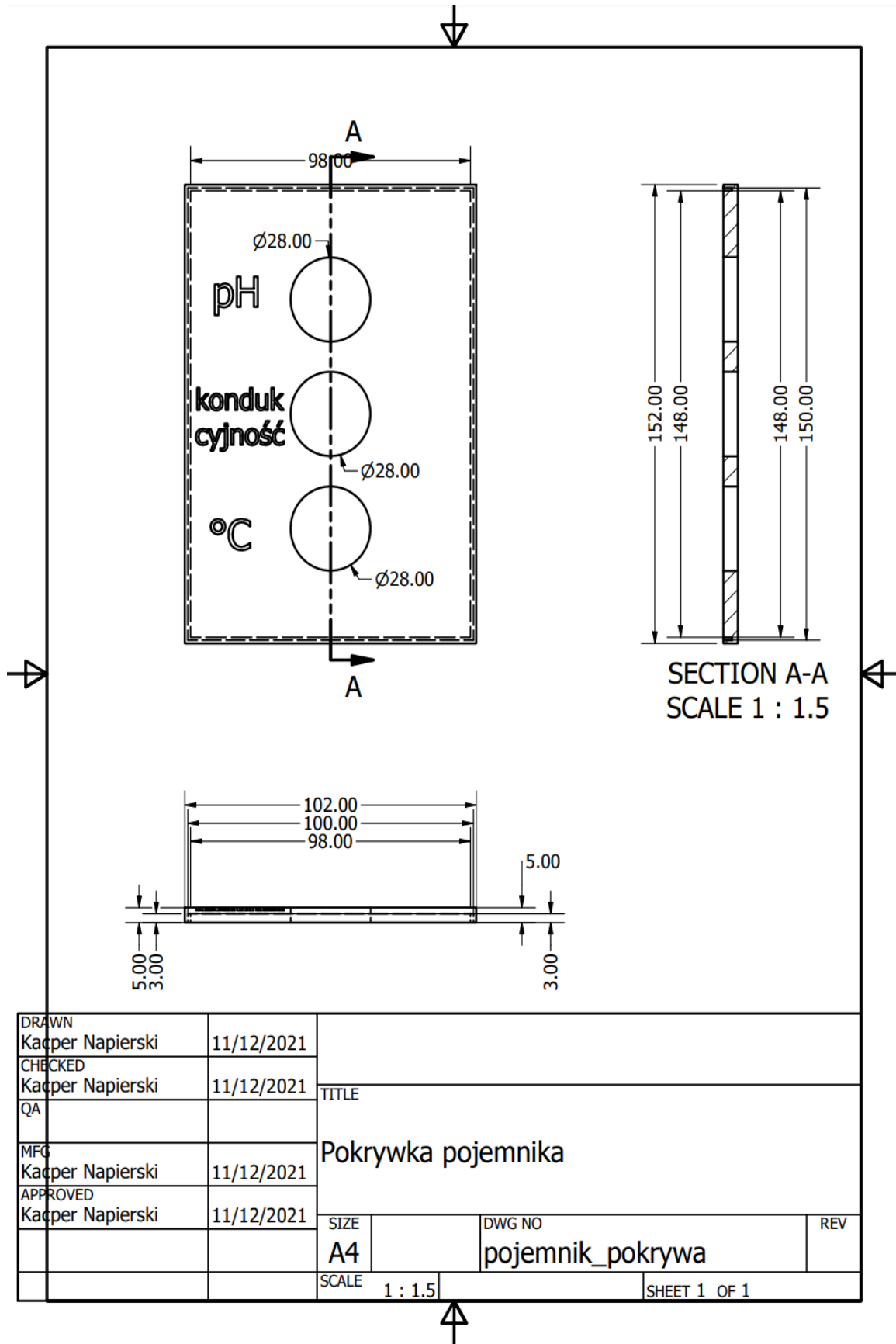
DRAWN	Kacper Napierski	11/12/2021	TITLE					
CHECKED	Kacper Napierski	11/12/2021				Podpora nr.2		
QA								
MFG	Kacper Napierski	11/12/2021						
APPROVED	Kacper Napierski	11/12/2021	SIZE	DWG NO	REV			
			A4	Podpora_2				
			SCALE	1 / 10	SHEET 1 OF 1			



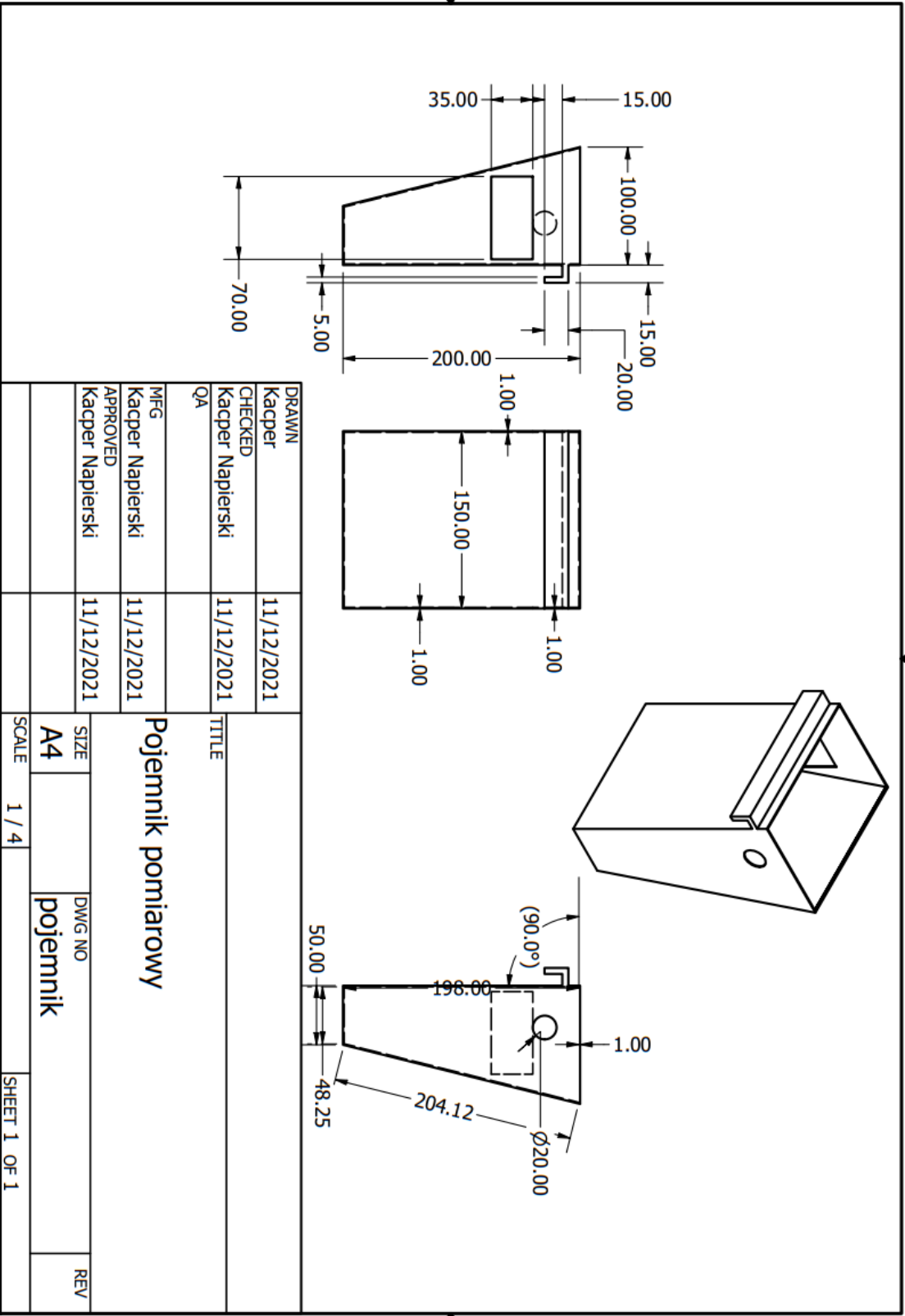


SECTION A-A
SCALE 1 : 1.5

DRAWN	Kacper Napierski	11/12/2021	TITLE	Stojak na pompe	SIZE	A4	DWG NO	stojak pompa	REV
CHECKED	Kacper Napierski	11/12/2021							
QA									
MFG	Kacper Napierski	11/12/2021	SCALE		1 : 1.5	SHEET 1 OF 1			
APPROVED	Kacper Napierski	11/12/2021							

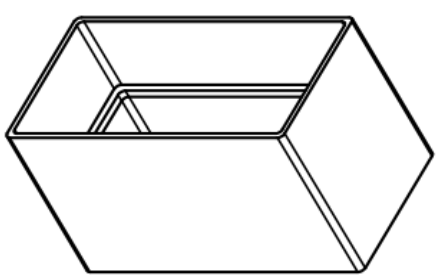
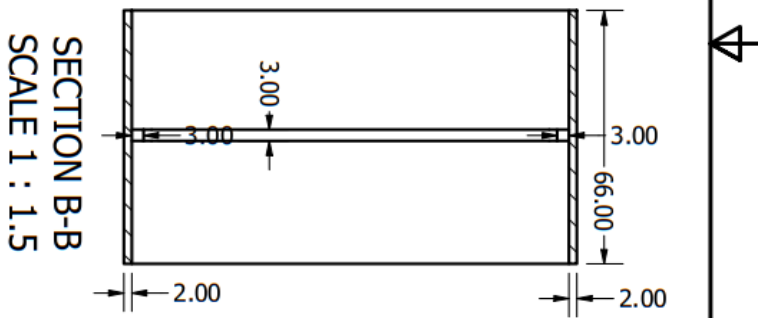
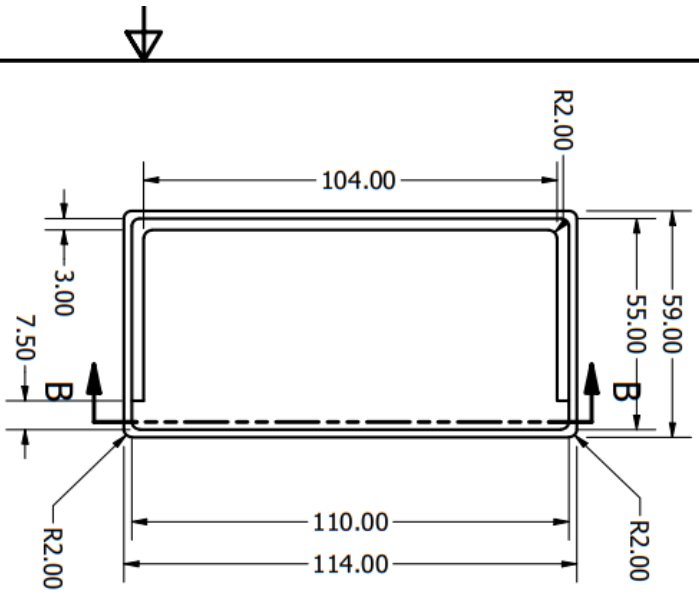


DRAWN	Kacper Napierski	11/12/2021	TITLE Pokrywka pojemnika		
CHECKED	Kacper Napierski	11/12/2021			
QA					
MFG	Kacper Napierski	11/12/2021			
APPROVED	Kacper Napierski	11/12/2021	SIZE	DWG NO	REV
			A4	pojemnik_pokrywa	
			SCALE	SHEET 1 OF 1	
			1 : 1.5		



DRAWN	Kacper	11/12/2021
CHECKED	Kacper Napierski	11/12/2021
QA		
MFG	Kacper Napierski	11/12/2021
APPROVED	Kacper Napierski	11/12/2021

TITLE		Pojemnik pomiarowy	
SIZE	A4	DWG NO	pojemnik
SCALE	1 / 4	SHEET	1 OF 1
REV			



DRAWN	Kacper Napierski	11/12/2021
CHECKED	Kacper Napierski	11/12/2021
QA		
MFG	Kacper Napierski	11/12/2021
APPROVED	Kacper Napierski	11/12/2021

TITLE		SCALE	SIZE	DWG NO	REV
		1 : 1.5	A4	Lacznik	
		SHEET 1 OF 1			

DODATEK C: Pliki modeli 3D (płyta CD)

Załączona płyta CD zawiera pliki modeli 3D w formatach ipt. I iam.. Stworzone przy pomocy oprogramowania CAD Autodesk Inventor PRO w wersjach 2018 i 2022. Wewnątrz folderu o nazwie "Modele" są pliki:

- Rura.ipt
- Podpora_1.ipt
- Podpora_2.ipt
- Podpora_3.ipt
- Stojak pompa.ipt
- Pojemnik_pokrywa.ipt
- Pojemnik.ipt
- Lacznik.ipt
- Rurka2.ipt
- Rurka_pompka.ipt
- Butla.ipt
- Silnik.ipt
- Pompa_gerar_2.ipt
- Pompa_zewnatrz.ipt
- Pompa_gear.ipt
- Końcówka_rurka_pompka.ipt
- Przepływowy.ipt
- Czujnik.ipt
- Zbiornik.ipt
- Usieta_rura.ipt
- Trójnik_2.ipt
- Pion.ipt
- Kolano.ipt
- Pompa perystaltyczna.iam
- Pojemnik_zbiornik.iam
- Pojemnik_zlozenie.iam
- model_2.iam

