



POLSKO-JAPOŃSKA AKADEMIA
TECHNIK KOMPUTEROWYCH

WYDZIAŁ INFORMATYKI
KATEDRA SIECI KOMPUTEROWYCH
SIECI URZĄDZEŃ MOBILNYCH

Piotr Werczyński
s17093

**System nawadniania i ochrony roślin przed
niekorzystnymi warunkami klimatycznymi**

Praca inżynierska
dr inż. Michał Tomaszewski

Warszawa, Luty 2021

Spis treści

1 Modelowanie 3D	2
1.1 Model pierwszy z wykorzystaniem lin	2
1.1.1 Kij	3
1.1.2 Druty	4
1.1.3 Wałek	5
1.1.4 Szpula	5
1.1.5 Złożenie	6
1.1.6 Wnioski na temat pierwszego podejścia	8
1.2 Model drugi z wykorzystaniem przekładni	9
1.2.1 Przekładnia planetarna	10
1.2.2 Przekładnia ślimakowa	11
1.2.3 Słońce	11
1.2.4 Zębatka	13
1.2.5 Pierścień	14
1.2.6 Drut	15
1.2.7 Złożenie	16
1.3 Ostateczny prototyp	18
2 Wykorzystane elementy elektryczne	21
2.1 Czujniki	21
2.1.1 Czujnik wilgotności gleby: HL-69	22
2.1.2 Czujnik natężenia światła: Fotorezystor	23
2.1.3 Czujnik temperatury i wilgotności powietrza: DHT11 - moduł Iduino SE052	24
2.2 Silnik krokowy	25
2.2.1 28BYJ-48 5V	25
2.2.2 Sterownik silnika krokowego ULN2003	27
2.3 Mikrokontroler	28
2.3.1 Arduino Uno	28
2.4 Zasilanie	29
2.5 Układ	30

3 Oprogramowanie	32
3.1 Przepływ sygnałów	32
3.2 Stany urządzenia i badanych elementów	33
3.3 Obsługa silników krokowych	35
3.4 Obsługa czujnika wilgotności i temperatury powietrza	37
3.5 Obsługa fotorezystora	39
3.6 Obsługa czujnika wilgotności gleby	41
3.7 Algorytm sterowania	43
4 Podsumowanie Projektu	46

Streszczenie

Celem niniejszej pracy dyplomowej było utworzenie urządzenia do magazynowania wody i ochrony roślin przed niskimi temperaturami. Praca została podzielona na trzy rozdziały w których zostały opisane etapy realizacji projektu.

W pierwszym rozdziale zostały opisane zaprojektowane modele przygotowane do wydrukowania w technologii druku 3D. Kolejny rozdział został poświęcony opisowi wykorzystanych części elektrycznych oraz mikrokontrolera. W ostatniej części pracy opisano algorytm sterowania urządzeniem oraz sposób komunikacji mikrokontrolera z podłączonymi urządzeniami.

Wprowadzenie

Postępujące w bardzo szybkim tempie zmiany klimatu są niepodważalnym faktem. W tym kontekście coraz częściej podnoszony jest temat rosnącego zapotrzebowania na wodę oraz tego, że w niedalekiej przyszłości może jej zabraknąć. W ostatnich latach można zaobserwować częstsze występowanie suszy na terenach krajów europejskich. Długie tygodnie bez deszczu przestają już kogokolwiek dziwić. Deficyt wód opadowych doprowadza do obumierania roślin oraz konieczności przeznaczania wielu litrów wody na nawadnianie terenów zielonych w miastach.

Kolejnym problemem związanym z próbą ochrony roślin przed niekorzystnymi warunkami klimatycznymi są zimy podczas których należy je osłonić przed niskimi temperaturami. Okrycie roślin materiałami pozwalającymi przetrwać mrozy jest zajęciem pracochłonnym, szczególnie jeśli mówimy o terenach z dużą ilością nasadzeń. Opisane powyżej sytuacje stanowią wyzwanie dla miast. Konieczność zapewnienia roślinom dodatkowej ochrony przed niekorzystnymi warunkami atmosferycznymi jest dużym przedsięwzięciem logistycznym angażującym nie tylko duże zasoby finansowe ale także osobowe.

Wspomniane powyżej problemy skłoniły do próby stworzenie urządzenia pozwalającego magazynować wodę oraz osłaniać rośliny przed niekorzystnymi warunkami panującymi zimą. Ma ono sprawić, że pielęgnacja roślin dzięki wykorzystaniu innowacyjnych rozwiązań będzie prostsza i mniej czasochłonna.

Rozdział 1

Modelowanie 3D

W tym rozdziale został opisany proces konstrukcji modelu parasola. Do realizacji tego zadania zostało wykorzystane narzędzie o nazwie "Solidworks". Solidworks jest narzędziem typu CAD (z angielskiego Computer Aided Design). Są to narzędzia do wspomagania procesu projektowania na komputerze. Wykorzystanie tych narzędzi pozwala na projektowanie w dużo szybszy i bardziej przyjazny sposób obiektów 2D i 3D niż na papierze. Oprogramowania tego typu udostępniają również narzędzia pozwalające na przeprowadzenie symulacji ruchu na obiekcie oraz zobrazowanie projektu w przestrzeni 3D i szybkie wprowadzanie zmian odnośnie wymiarów oraz kształtu komponentów.

Na potrzeby tego projektu zostały stworzone modele w przestrzeni trójwymiarowej aby umożliwić wykonanie druku w drukarce 3D.

1.1 Model pierwszy z wykorzystaniem lin

W tym podejściu została podjęta decyzja zaprojektowania prototypu parasola chroniącego mniejsze drzewa, krzewy i rośliny. Parasol korzysta z dwóch rodzajów materiałów: agrowłókniny i tkaniny wodoodpornej. Agrowłóknina została zastosowana aby utrzymywać temperaturę w otoczeniu rośliny i chronić ją przed zimnem. Zastosowano właśnie ten materiał ponieważ pozwala on przepuszczać

powietrze do rośliny a co za tym idzie dostarczać tlen dzieki temu roślina nie zwróciłnie. Zastosowanie innego typu materiału który nie przepuszczał by tlenu do rośliny mogło by doprowadzić do jej wymarcia. Tkanina wodooodporna została zastosowana aby chronić roślinę przed opadami oraz aby magazynować wodę we-wnętrz kielicha utworzonego podczas jednego ze stanów parasola który zostanie opisany w dalszej części rozdziału. W modelu zostały wyróżnione następujące części:

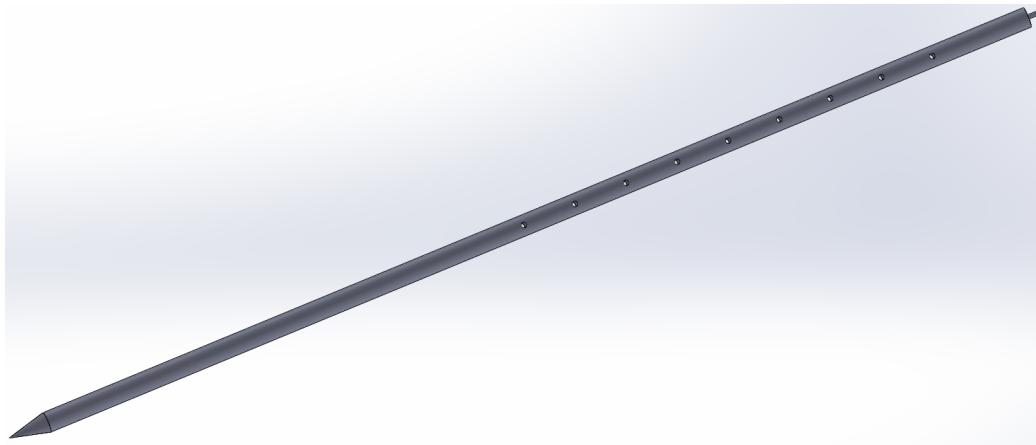
- Kij,
- 8 drutów,
- Wałek,
- Szpula

W dalszej części sekcji zostanie opisana konstrukcja poszczególnych części, rola jaką odgrywają w modelu oraz opis całego złożenia. Na sam koniec sekcji zostaną wyciągnięte wnioski odnośnie zaproponowanej konstrukcji.

1.1.1 Kij

Kij jest główną częścią na której opiera się cała konstrukcja parasola. Jest on zakończony ostrym końcem w kształcie stożka który umożliwia wbicie kija w ziemię obok rośliny. Na drugim końcu kija pozostawiono miejsce na silnik który ma pozwolić na obrót szpulą. Jak widać na ilustracji 1.1, na długości kija zostało utworzone 9 otworów umożliwiających zablokowanie konstrukcji parasola oraz uregulowanie jego wysokości. Taki efekt można uzyskać łącząc część kija i wałka w całość. Nie zdecydowano się jednak na takie połączenie ponieważ zastosowane podejście zwiększa elastyczność urządzenia. Dzięki takiemu rozwiązaniu

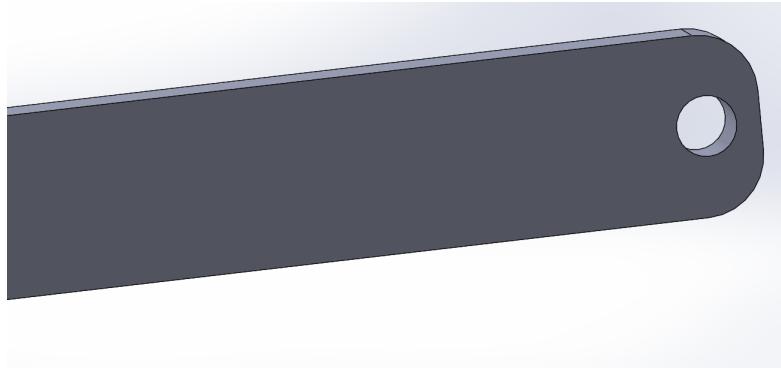
użytkownik może w prosty sposób regulować wysokość parasola bez dużej interwencji w działanie całego urządzenia. Odstępy między otworami wynoszą 5 cm co pozwala dostosować konstrukcję na potrzeby konkretnej rośliny.



Rysunek 1.1: Kij na którym opiera się konstrukcja

1.1.2 Druty

Drut jest częścią odpowiedzialną za utrzymanie materiałów. W konstrukcji znajdują się 8 drutów. Drut na każdym końcu posiada otwór. Na rysunku 1.2 przedstawiono jeden z końców części. Otwór z jednej strony został utworzony na potrzeby umieszczenia części w specjalnie przygotowanych do tego celu zawiasach. Z drugiej strony otwory zostały wykorzystane do mocowania materiałów oraz przełożenia sznurów. Otwory po obu stronach są identycznej wielkości i identycznie umiejscowione przez co podczas montażu nie powinny sprawiać większej trudności użytkownikowi. Drut jest częścią wprawianą w ruch za pomocą lin przywiązywanych na jednym z końców. To przez ten fakt jego budowa jest dość cienka aby genetrować jak najmniejsze ciężary do podnoszenia za pomocą lin przez silnik krokowy.



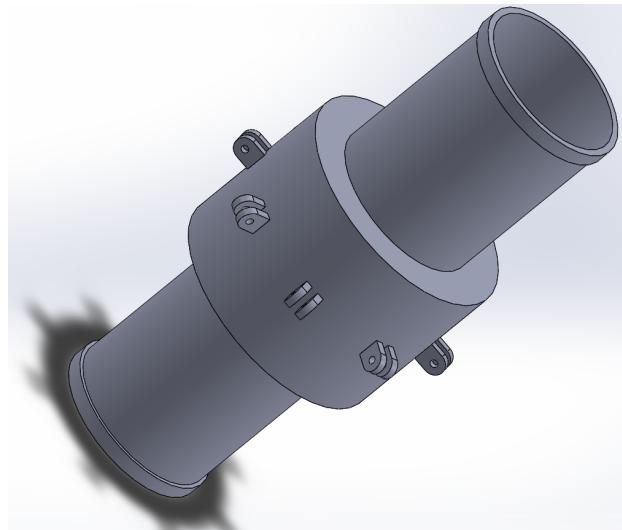
Rysunek 1.2: Otwór w drucie

1.1.3 Wałek

Wałek jest częścią do której zostały umocowane druty. Jak można zaobserwować na rysunku 1.3 składa się on z 16 wystających elementów które w parze tworzą zawiasy do mocowania drutów. Odległość miedzy parami wystających elementów jest równa szerokości drutu. Zawiasy mają wydrążone otwory przez które przekłada się element mocujący do nich druty dzięki czemu zostają one zablokowane w miejscach mocowań. Cały wałek posiada duży otwór o średnicy równej średnicy kija który został wykorzystany do przełożenia kija przez otwór i umocowania całej konstrukcji na nim. Otwory utworzone w kiju pozwalają na wstawienie elementu blokującego przemieszczanie się wałka po kiju oraz zablokowaniu jego pozycji. Dzięki takiemu rozwiązaniu użytkownik uzyskuje wysoką elastyczność w dostosowywaniu wysokości konstrukcji.

1.1.4 Szpula

Szpula jest częścią wokół której oplatane są sznury. Konstrukcja tej części została zainspirowana kobestanem który jest wykorzystywany na łodziach żaglowych. Część montowana jest na niezastrzonym końcu kija w specjalnie przygotowanym do tego miejscu. Budowa elementu została przedstawiona na rysunku

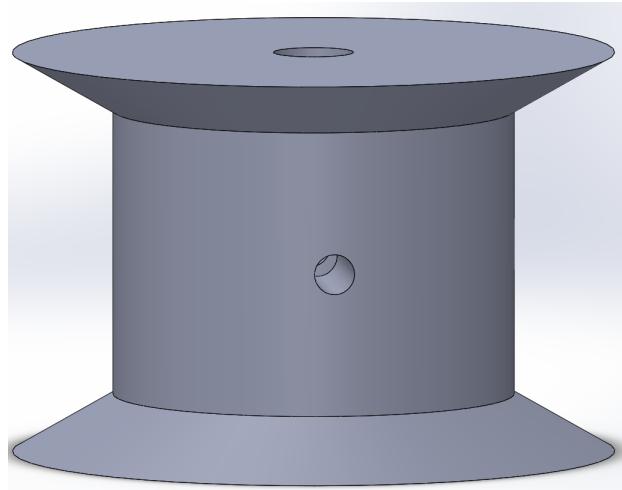


Rysunek 1.3: Wałek

1.4. Jak można zaobserwować na ilustracji część posiada w sobie dwa otwory, jeden pionowy pozwalający na zamocowanie jej na wale silnika krokkowego oraz drugi pozwalający na przeplecenie przez niego lin i zablokowanie ich grubym węzłem. Tak zablokowane sznury są następnie opłatane wokół szpuli oraz mocowane na specjalnie utworzonych do tego otworach w drutach. W momencie obrotu szpuli liny będą zmieniały swoją długość co umożliwi wprawienie w ruch drutów i zmianę stanu parasola. Szpula mogłaby być tylko częścią w kształcie walca na który zostaje nawijana lina. Jednak dzięki zastosowaniu stożkowego kształtu na każdym z końców wału lina zostaje zablokowana i utrudnia to jej ześlizgnięcie się z mechanizmu.

1.1.5 Złożenie

Ostateczne złożenie wszystkich komponentów utworzy parasol przedstawiony na rysunku 1.5. Parasol może znajdować się w 4 stanach w zależności od kąta druta do osi kija:

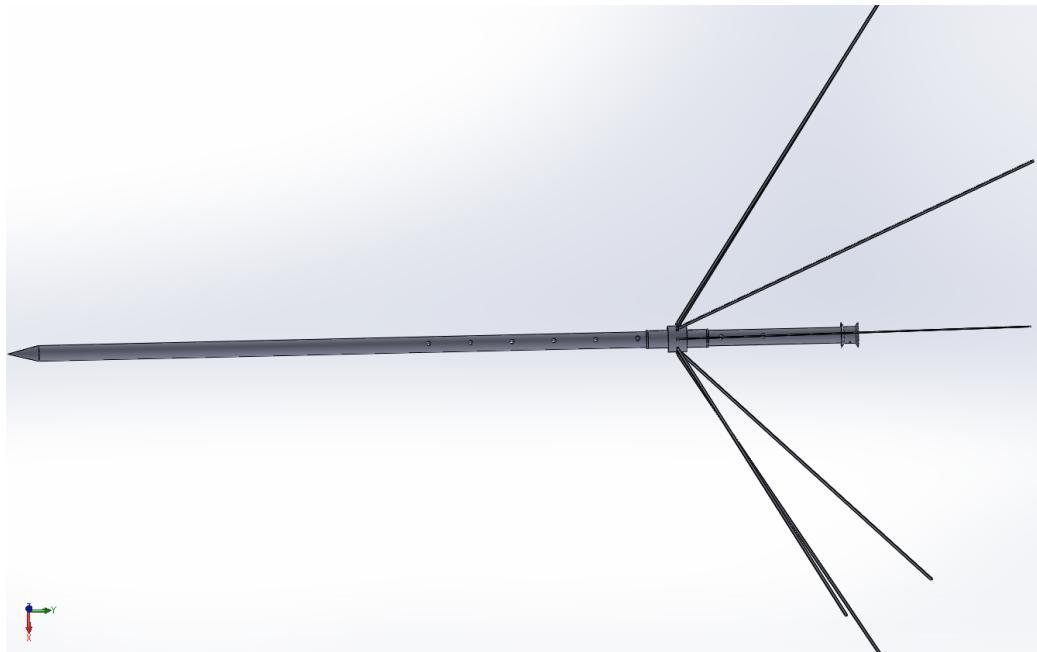


Rysunek 1.4: Szpula

- Kielicha- gdzie kąt $< 45^\circ$ (*tak jak na rysunku 1.5*)
- Osłona słoneczna- gdzie kąt = 90°
- Osłona Cieplna- gdzie kąt $> 90^\circ$ i kąt $< 180^\circ$
- Złożony- gdzie kąt = 180°

W momencie gdy mikrokontroler nadaje sygnał zmiany stanu wysyła do silnika krokkowego sygnał aby ten zaczął się obracać. Obrót wału silnika krokkowego wprawia w ruch obrotowy szpulę która nawija lub rozwija liny zmieniając kąt drutów między kijem i zmieniając stan parasola. Kielich jest stanem w którym parasol zaczyna magazynować w sobie wodę podczas opadów dzięki zamontowanej od góry wodoodpornej tkaninie. Osłona słoneczna jest stanem w którym parasol pełni rolę ochrony przed słońcem przy nadmiernym oświetleniu i wysokich temperaturach aby chronić roślinę przed namierną ekspozycją na słońce. Osłona cieplna pełni funkcję ochrony rośliny przed za niskimi temperaturami. W tym stanie zamontowana na końcach agrowłóknina tworzy kopułę trzymającą temperaturę wokół rośliny. Stan złożony sprawia że cały parasol zostaje zwinięty co

pozwala na odsłonięcie rośliny w pełni na zewnętrzne warunki atmosferyczne.



Rysunek 1.5: Złożenie podejście 1

1.1.6 Wnioski na temat pierwszego podejścia

Analiza powyższego podejścia pozwoliła na wyciągnięcie wniosków oraz problemów wynikających ze specyfiki konstrukcji. Wyróżnione poniżej wnioski doprowadziły do zaprzestania rozwoju powyższego modelu i w efekcie niedoprowadzenia go do ostatecznej wersji.

Pierwszym problemem jest stabilność konstrukcji. Cały parasol opiera się na długim kiju. Duża powierzchnia materiałów jest narażona na silne podmuchy wiatru co może doprowadzić do przewrócenia i w efekcie uszkodzenia konstrukcji lub rośliny. Nie można jednak zrezygnować z zastosowania materiałów gdyż są one nieodłączną częścią projektu bez których nie można by było uzyskać zamierzonych efektów. Innym sposobem zmniejszenia wpływu wiatru na stabilność konstrukcji byłoby stworzenie otworów w materiałach pozwalających

na częściową ucieczkę powietrza podczas podmucha wiatru. Niestety tak jak i w poprzednim przypadku takie podejście spowodowałoby pogorszenie efektywności działania takiego urządzenia. Przez zastosowanie takich otworów urządzenie nie trzymałoby odpowiedniej temperatury wokół rośliny oraz magazynowanie wody stanowiłoby większy problem przez zastosowane otwory.

Kolejnym problemem jest siła jaką trzeba przyłożyć aby wprawić w ruch druty. Jeden silnik krokowy musiałby potencjalnie wprawiać w ruch 8 drutów obciążonych ciężarem materiałów oraz wody znajdującej się w utworzonym kielichu. Samo obciążenie duża ilość wody mogłoby wpłynąć również na uszkodzenie materiałów takich jak druty czy liny. Niestety powyższe podejście pozwala zwiększać moc tylko poprzez zwiększanie mocy pojedynczego silnika co dużo bardziej zwiększa koszty niż zakup kilku słabszych silników. Niestety model nie został zaprojektowany w sposób który umożliwia montaż większej liczby silników.

1.2 Model drugi z wykorzystaniem przekładni

Przez problemy zdefiniowane podczas pierwszego podejścia została podjęta decyzja utworzenia nowego modelu. Pierwsze podejście pozwoliło jednak określić dalszy kierunek rozwoju projektu oraz problemy które należy wyeliminować w nowo utworzonym modelu. Jednym z wyciągniętych wniosków wpływających na stworzenie nowej konstrukcji była potrzeba przyłożenia większej siły w celu wprawienia w ruch parasola. W wyniku tego wniosku, w nowej konstrukcji wykorzystano trzy silniki krokowe zamiast jednego. Zmiana podejścia doprowadziła również do usunięcia koncepcji lin z projektu. Zamiast tego do przełożenia siły zostały wykorzystane dwa układy maszyn służące do przeniesienia ruchu. Są nimi przekładnia ślimakowa i przekładnia planetarna. W przeciwieństwie do poprzed-

niego projektu materiały otrzymały swoje oddzielne uchwyty i w momencie gdy jeden jest rozwijany drugi się zwija. Sam wygląd i działanie modelu się zmienił i zostały wyróżnione w nim następujące części:

- Słońce,
- 3 druty,
- 3 zębatki,
- Pierścień
- 3 mocowania silników

Aby zrozumieć działanie konstrukcji w poniższych sekcjach zostały wytłumaczone takie zagadnienia jak przekładnia planetarna, przekładnia ślimakowa i opis każdej części i roli jaką odgrywa. Na samym końcu opisaną konstrukcję oraz jej stany wynikające ze przemieszczenia się części.

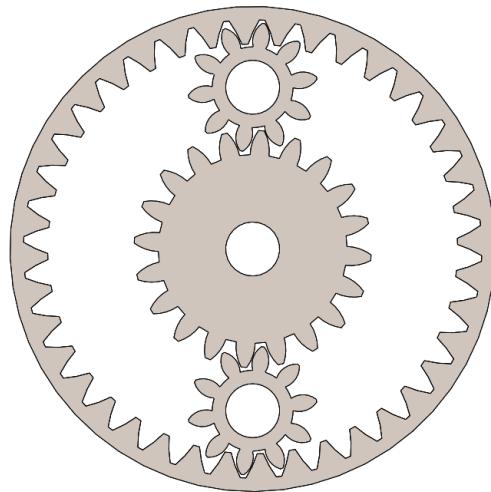
1.2.1 Przekładnia planetarna

Przekładnia planetarna inaczej przekładnia obiegowa jest to rodzaj przekładni przenoszący moment obrotowy. W przekładni planetarnej wyróżniamy następujące części:

- Koło zebate wewnętrzne (centralne) z zębami na zewnątrz,
- Koło pierścieniowe zewnętrzne z zębami wewnętrz,
- Koła obiegowe z zębami na zewnątrz,

Każda z powyżej przedstawionych części można zaobserwować na rysunku 1.6. W momencie wprawienia w ruch jednego koła inne koła zaczynają się obracać

w wyniku przeniesienia energii. W zależności od tego które koła są zablokowane koła zębate obracają się w inną stronę. Zaletą tego rozwiązania jest to że przekładnia pozwala przenosić dużo większe przełożenia.



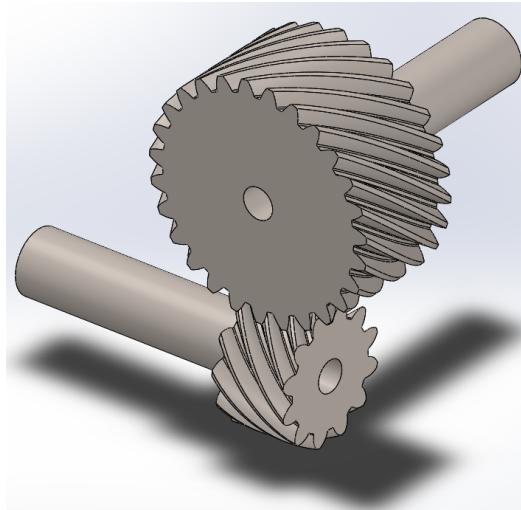
Rysunek 1.6: Przekładnia planetarna

1.2.2 Przekładnia ślimakowa

Przekładnia ślimakowa składa się z dwóch części o prostopadłych do siebie osiach: ślimacznicy i ślimaka. Ślimaczница jest kołem zębatym z zebami śrubowymi. Ślimak jest walcem z wykończeniem śrubowym. W momencie obrotu jednej części wzdłuż swojej osi dochodzi do przeniesienia momentu obrotowego na inną część. Przykładowa konstrukcja tego typu została przedstawiona na ilustracji 1.7 .

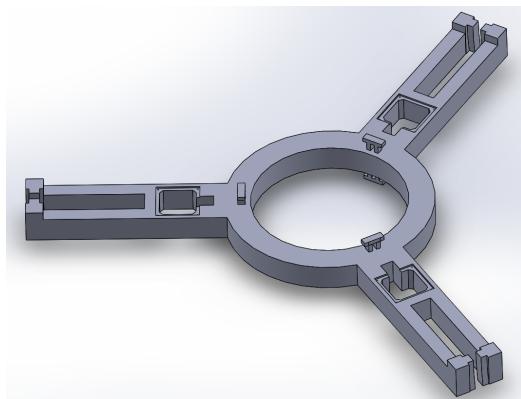
1.2.3 Słońce

Słońce jest podstawą całej konstrukcji. Konstrukcja składa się z pierścienia i 3 ramion co można zaobszerwować na rysunku 1.8. Część posiada na środku otwór który został stworzony w celu umieszczenia konstrukcji na pniu drzewa. Na kon-



Rysunek 1.7: Przekładnia ślimakowa

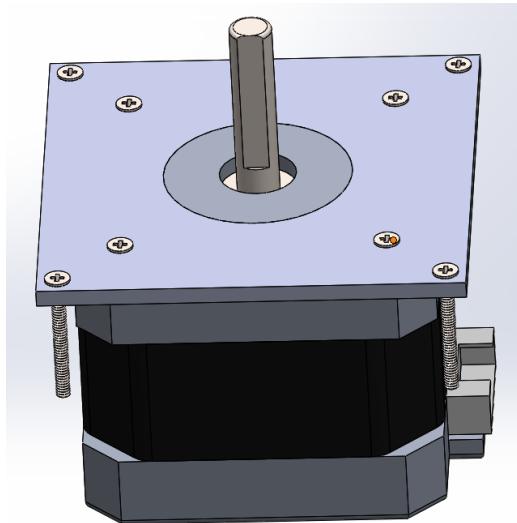
strukcji umieszczone zostało 6 uchwytów na materiał z czego 3 od góry i 3 symetrycznie po drugiej stronie.



Rysunek 1.8: Słońce

Na każdym ramieniu słońca powstały otwory do umieszczenia silników krokowych. Do umocowania silników krokowych potrzeba mocowań które sprawią że silniki będą wisiały na konstrukcji i zostaną umieszczone w otworach słońca. Mocowanie silników zostało przedstawione na rysunku 1.9. Za pomocą śrub silnik przykręcany jest do elementu mocującego a element mocujący do słońca.

Otwory zostały zaprojektowane pod obudowe silników krokowych NEMA 17 które są przedstawione na rysunku 1.9 razem z mocowaniem. Koniec każdego ramienia zostało zakończony szynami z obu stron które mają umożliwić przemieszczanie się drutów po konstrukcji. Stworzenie otworów na silniki i szyny umożliwiła operacja gniazda[5] w programie solidworks. Gniazdo pozwala stworzyć wycięcie w części na podstawie bryły innej części. Operacja gniazda szczególnie była pomocna w przypadku tworzenia szyn ponieważ dokładnie pozwoliło to odtworzyć kształt wydrążonych otworów w drutach to pozwoliło na zmianę pozycji drutów.

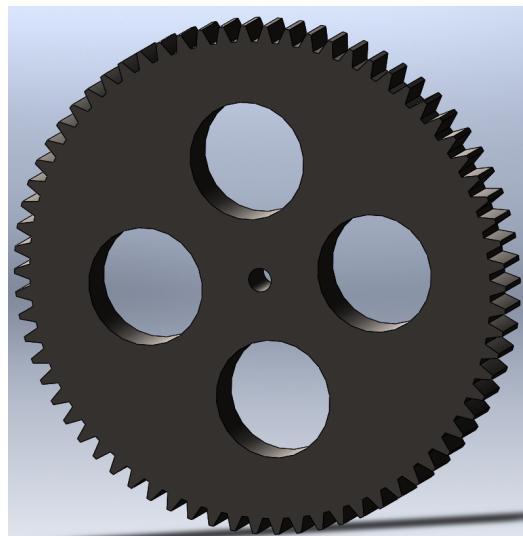


Rysunek 1.9: Mocowanie na silnik krokowy

1.2.4 Zębatka

Zębatka jest częścią wprawiającą w ruch reszte konstrukcji. Koło zębata jest rodzaju walcowego prostego. W przypadku tej części zastosowano zęby proste ponieważ nie było potrzeby stosowania zębów skośnych które niepotrzebnie skomplikowały by konstrukcję a co za tym idzie utrudniłoby to druk. Część znajduje się na wale silnika krokowego. W momencie obrotu zęby koła przekazują energię

zębom wewnątrz pierścienia wprawiając go w ruch. Do utworzenia kół zębatych użyto biblioteki toolbox[6] w narzędziu solidworks. Pozwala ono na utworzenie wcześniej zdefiniowanych części określając jej parametry. W przypadku tego projektu wykorzystano koło zębate o 70 zębach i module 1.5. Jak można zaobserwować na Rysunku 1.10 Komponent posiada również 4 otwory utworzone w celu zmniejszenia ciężaru konstrukcji. Otwory zostały umieszczone w różnych odległościach oraz są różnych rozmiarów. Na środku koła zębnego został również umieszczony otwór pozwalający na montaż części na wale silnika.



Rysunek 1.10: Zębatka

1.2.5 Pierścień

Pierścień jest częścią ruchomą stworzoną w celu przenoszenia siły na druty i wprawiania ich w ruch. Jak można zaobserwować na rysunku 1.11 na całej konstrukcji umieszczono 10 równo oddalonych od siebie otworów w kształcie elipsy w celu zmniejszenia ciężaru. Część posiada dwa rodzaje zębatek:

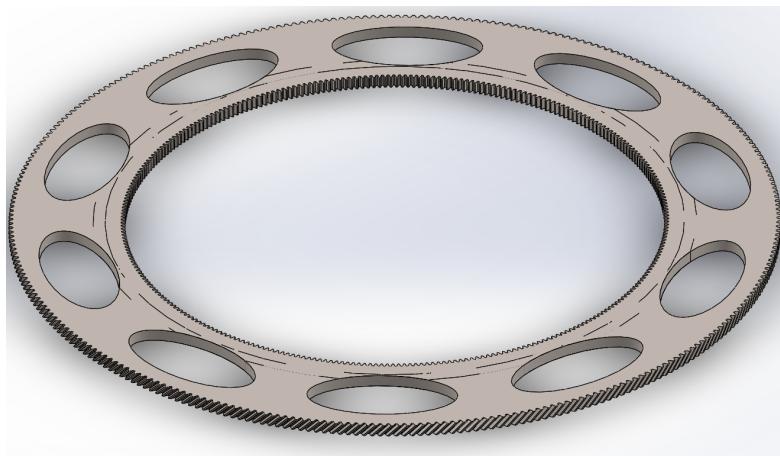
- proste w wewnętrznej części pierścienia

-
- skośne w zewnętrznej części pierścienia

Zęby proste są wykorzystywane aby zębatki mogły wprawić pierścień w ruch. Moduł jest równy modułowi kół zębatych czyli 1.5 a ilość zębów jest równa 300.

Zęby skośne są wykorzystywane aby pierścień mógł wprawić w ruch Druty. Druty są podnoszone albo upuszczane w zależności od kierunku obrotu pierścienia. Kąt zębów wynosi 45° . Ilość zębów i moduł są równe wewnętrznej części.

Do stworzenia całej części wykorzystano narzędzie solidworks toolbox[6]. Ponieważ moduł toolbox nie posiada wymaganej części do złożenia tego obiektu wykorzystano dwie różne części. Pierwszą częścią był pierścień z zewnętrznymi zębami skośnymi a drugą pierścień z wewnętrznymi zębami prostymi. Połączenie obu elementów pozwoliło na utworzenie jednego obiektu i dokonanie w nim odpowiednich zmian.

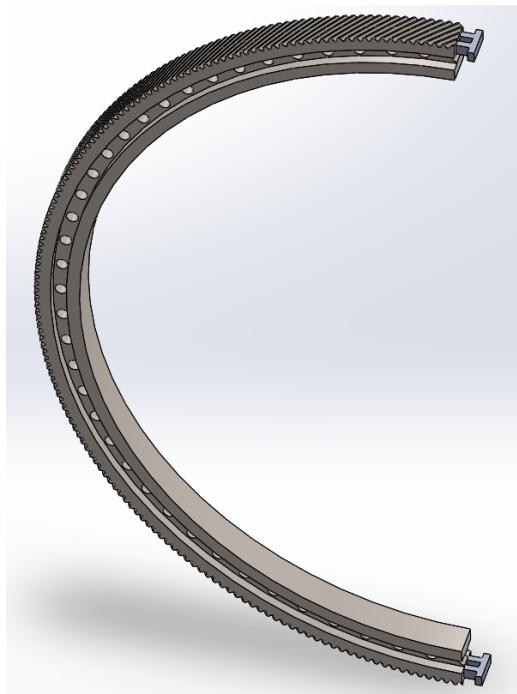


Rysunek 1.11: Pierścień

1.2.6 Drut

Drut jest częścią w kształcie półpierścienia. Został on utworzony z jednej z części składającej się na pierścień. Następnie część została przecięta na pół. Na

całej długości utworzono otwory aby zmniejszyć ciężar konstrukcji. Na każdym końcu umieszczono zaczepy na materiał co pozwala na zaczepienie na każdym z końców odpowiedniego materiału i manipulowanie jego rozwinięciem poprzez ruch. Na zewnętrznej części półpierścienia znajdują się zęby skośne. Zostały one umieszczone w celu wprawiania drutu w ruch przez obracający się pierścień i przenoszenie energii z jednego zęba na drugi. Moduł półpierścienia jest równy 1.5 a ilość zębów wynosi 300. Jak można zaoferować na rysunku 1.12 cały drut posiada w środku wcięcie umożliwiające przemieszczanie się części po szynach umieszczonych w ramionach słońca.



Rysunek 1.12: Drut

1.2.7 Złożenie

Ostateczne złożenie wykorzystuje połączenie dwóch koncepcji przekładni ślimakowej i zmodyfikowanej przekładni planetarnej. W przeciwieństwie do prze-

kładni planetarnej zastosowana konstrukcja nie zawiera koła centralnego. Koła planetarne czyli zębatki w konstrukcji są nie ruchome i obracają się w miejscu wprawiając w obrót pierścień czyli pierścień zewnętrzny i ślimaka jednocześnie. Ślimak wprawiony w ruch przekłada siłę na ślimacznice czyli drut. Przez przekazaną siłę drut zmienia swoją pozycję a zmieniając pozycje zmienia stan. Jesteśmy w stanie wyróżnić takie stany jak:

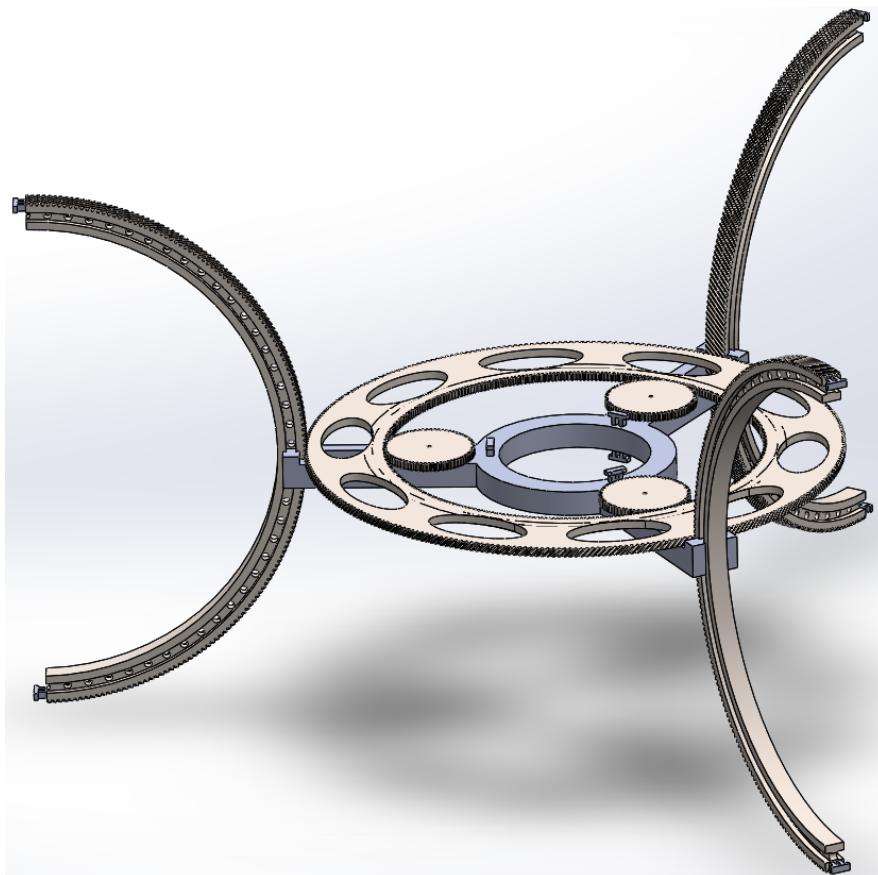
- Kielich
- Osłona

Kielich jest stanem w którym dłuższa część drutu znajduje się w górnej części konstrukcji. W tym momencie wodooodpornej część jest rozwinięta co pozwala na zebranie dużej ilości wody podczas opadu. Osłona jest stanem przeciwnym do stanu kielicha czyli dłuższa część drutu znajduje się w dolnej części konstrukcji. W tym momencie część z agrowłókniną jest rozwinięta co pozwala ochronić roślinę przed mrozami i zachować bardziej przyjazną temperaturę wokół rośliny.

Całą konstrukcję można zaobserwować na rysunku 1.13. Tak skonstruowane urządzenie posiada jednak parę wad. W porównaniu do pierwszego modelu takie podejście zmniejsza elastyczność dostosowywania wysokości parasola. W tym przypadku część jest na stałe montowana na pniu drzewa przez co jego modyfikacja wysokości może być utrudniona. Takie podejście posiada jednak również pozytywną stronę ponieważ zamontowanie urządzenia na pniu drzewa zwiększa znacząco jego stabilności.

Kolejną wadą tego rozwiązania jest jego żywotność. W porównaniu do pierwszego podejścia w tym rozwiążaniu podczas pracy urządzenia może przy długim użytkowaniu dojść do starcia się kół zębatych a co za tym idzie konieczności ich wymiany.

Pomimo kilku wymienionych wad do stworzenia prototypu ostatecznie użyto tego rozwiązania które zostało delikatnie zmodyfikowane. O ostatecznych modyfikacjach w modelu można wyczytać w kolejnym podrozdziale.



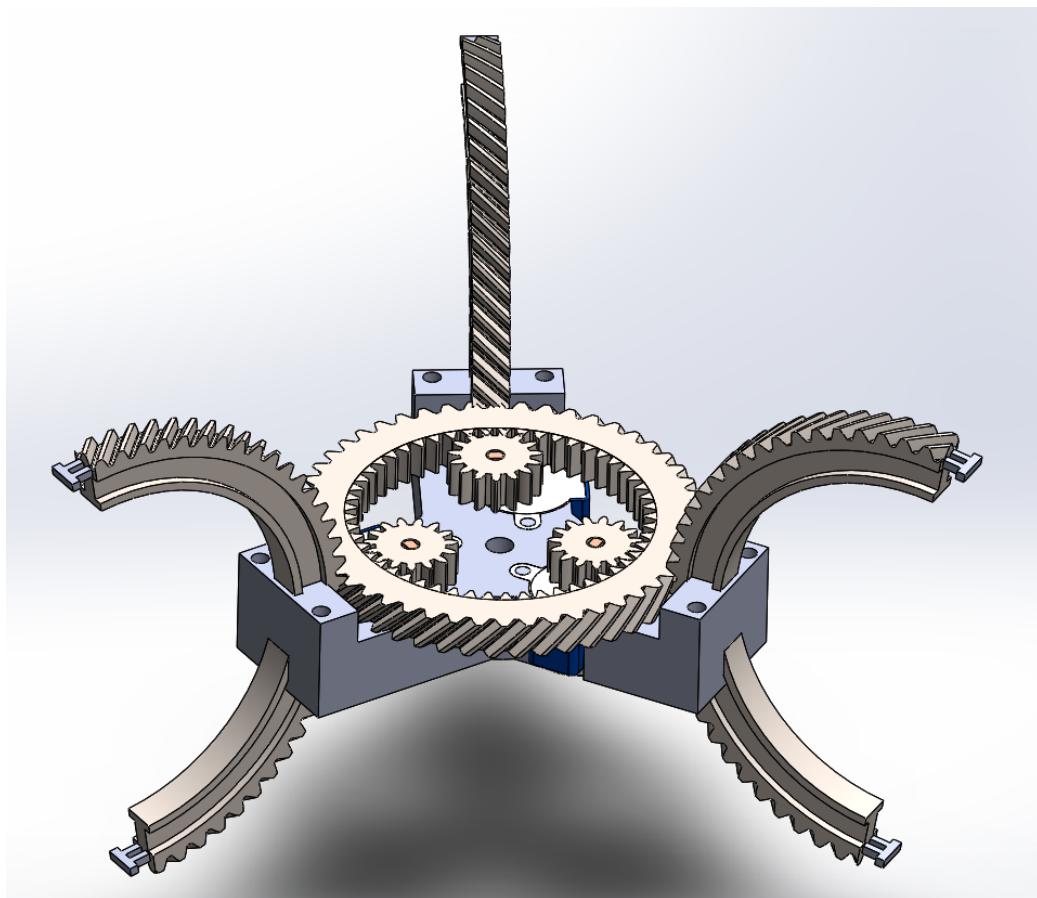
Rysunek 1.13: Złożenie

1.3 Ostateczny prototyp

Na potrzeby przygotowania prototypu części modelu zostały zmodyfikowane w celu uzyskania mniejszego rozmiaru. Zmniejszony model został oparty na konstrukcji modelu z drugiego podejścia. Poszczególne części zostały zmniejszone do rozmiarów mniejszych niż 15cm na 15cm w celu wygodnego wydrukowania

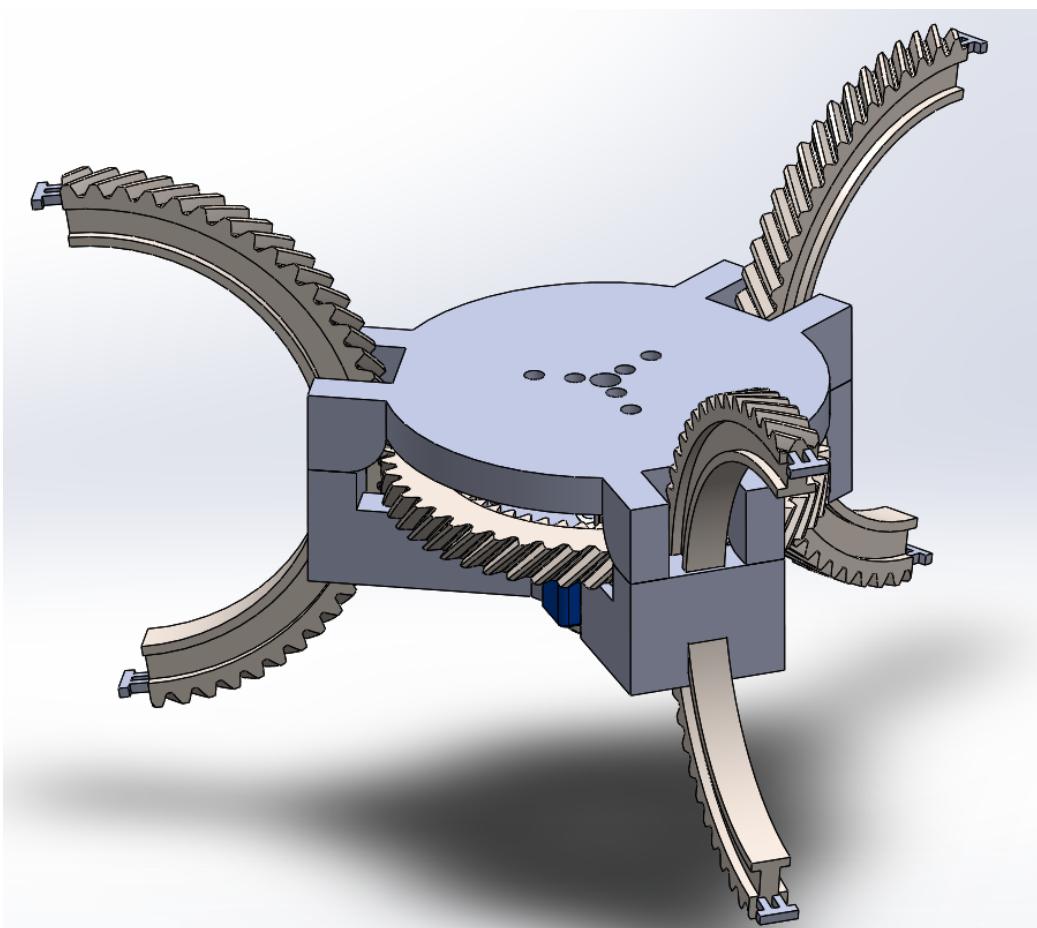
na drukarce 3D oraz zminimalizowania kosztów.

Otwory na silniki w komponencie nazywanym słońcem zostały zmodyfikowane i dostosowane do mniejszego silnika krokowego o nazwie 28BYJ-48. Wyjścia na kable w silnikach krokowych zostały ustawione równolegle do jednej ze ścian ramion w celu prostego wyprowadzenia i podłączenia kabli. Mocowania na silniki krokowe zostały wbudowane w słońce w formie 'słupków' przez które przekłada się obudowę silnika. Ze słońca usunięto również mocowania na materiał i zamieniono je na otwory umożliwiające mocowanie. Zmodyfikowany model można zaobserwować na rysunku 1.14.



Rysunek 1.14: Złożenie bez obudowy

Nową częścią dodaną do złożenia jest obudowa. Została ona dodana w celach niemożliwienia wkręcania się materiału w mechanizm kół zębatych. Jest ona mocowana w specjalnie przygotowanych otworach w słońcu. Ma ona również przygotowane 6 otworów które mają umożliwić zaczepienie wodooodpornego materiału do górnej części konstrukcji. Cała konstrukcja z nałożoną obudową została przedstawiona na rysunku 1.15.



Rysunek 1.15: Złożenie z obudową

Rozdział 2

Wykorzystane elementy elektryczne

Niniejszy rozdział został poświęcony działaniu układu elektrycznego wykorzystanego w projekcie. Przedstawiono w nim jakie części wykorzystano, sposób działania każdej z części oraz funkcje jaką pełnią w projekcie. Na koniec podsumowano budowę całego układu oraz sposób w jaki komunikują się ze sobą poszczególne komponenty.

2.1 Czujniki

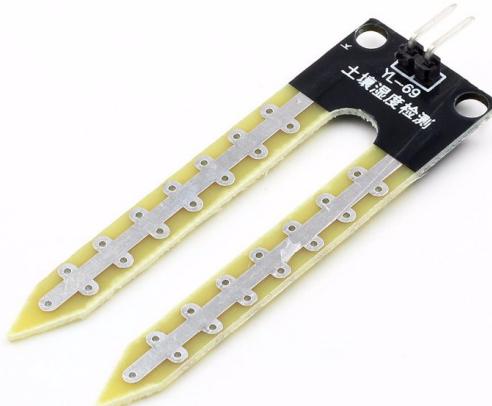
W przedstawionym projekcie czujniki są urządzeniami mającymi na celu wychwytywać sygnały wynikające z panujących warunków atmosferycznych wokół rośliny ochronianej przez parasol. W projekcie zastosowano 3 typy czujników takich jak:

- Czujnik wilgotności gleby,
- Czujnik natężenia światła,
- Czujnik temperatury i wilgotności powietrza,

W dalszej części opisano działanie poszczególnych elementów oraz w jaki sposób odebrano z nich sygnał.

2.1.1 Czujnik wilgotności gleby: HL-69

W projekcie do wykrywania zmian związanych ze stanem gleby wykorzystano czujnik wilgotności gleby. Użyta sądą w projekcie jest sonda o nazwie HL-69 która została przedstawiona na rysunku 2.1. Aby dokonać pomiarów należy sonde wprowadzić w glebe. Sonda wykrywa zmiany wilgotności dzięki właściwości zmiany rezystancji na skutek wilgoci. Większe wartości napięcia uzyskujemy przy wzroście suchości gleby a mniejsze przy wyższym współczynniku nawodnienia gleby.



Rysunek 2.1: Sonda czujnika wilgotności gleby HL-69¹

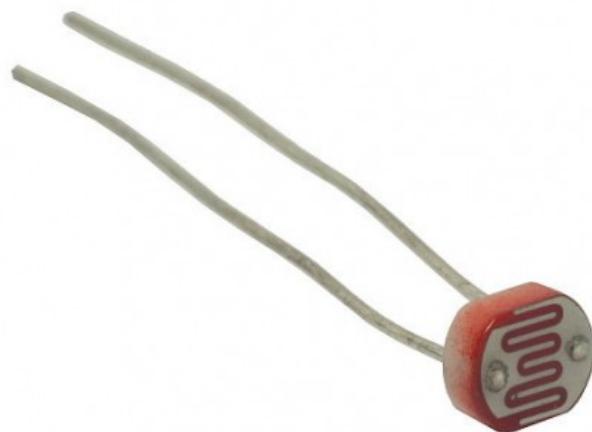
Problemem związanym z wykorzystaniem tego komponentu jest jego żywotność. Podczas użytkowania sonda podczas kontaktu z wodą może ulegać korozji i po czasie może zostać w pełni uszkodzona przez co przestanie działać a co za tym idzie bedzie wymagała wymiany. Na szczęście sam koszt elementu nie jest duży

¹<https://cf.shoppee.com.br/file/2daaf03d23eaa020b573f85861bc192f>

co jest zaletą tej części.

2.1.2 Czujnik natężenia światła: Fotorezystor

Do rejestracji zmian w natężeniu światła został wykorzystany fotorezystor. Fotorezystor składa się z dwóch elektród oraz powierzchni światłoczułej. Budowa elementu została przedstawiona na rysunku 2.2. Rezystancja w fotorezystorach jest zależna od natężenia światła padającego na światłoczułą powierzchnię. W momencie zwiększania natężenia światła opór w fotorezystorze będzie się zmniejszał natomiast w całkowicie nieoświetlonej przestrzeni będzie on osiągał najwyższą rezystancję.



Rysunek 2.2: Fotorezystor²

Innym elementem który mógłby zostać wykorzystany do pomiaru natężenia światła jest fotodioda. W fotodiodach fotony padające na urządzenie są absorbo-

²https://mageek.com.pl/601-large_default/fotorezystor-g15537-2m-dark-20-50k-10lx.jpg

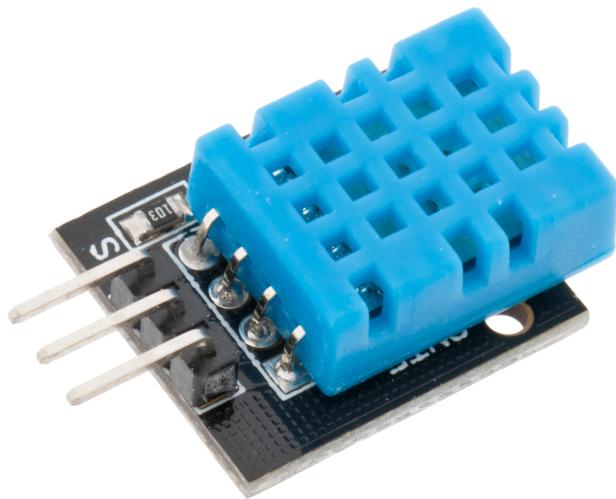
wane. W momencie padania światła na element zwiększa się prąd przepływający przez układ. Dzięki jej zastosowaniu można badać zmianę natężenia światła poprzez zmianę natężenia prądu wychodzącego. W powyższym projekcie ostatecznie skorzystano z fotodiony z powodu dużo niższych kosztów oraz wystarczającej precyzji wymaganej do realizacji projektu

Aby dołączyć fotorezystor do układu z jednej strony powinien on zostać podłączony do napięcia a z drugiej do uziemienia. Aby dokonać odczytu fotorezystor powinien zostać podpięty do wejścia analogowego w mikrokontrolerze. Pozwoli to na odczyt zmiany natężenia prądu a co za tym idzie odczyt zmiany natężenia światła.

2.1.3 Czujnik temperatury i wilgotności powietrza: DHT11 - moduł Iduino SE052

Kolejną częścią której użyto w projekcie jest czujnik DHT11. Jego wykorzystanie umożliwia uzyskanie danych o temperaturze oraz o względnej wilgotności powietrza. Jak można wyczytać z noty katalogowej[4] czujnik składa się z 4 pinów. Aby zasilić urządzenie należy pierwszy pin podłączyć do zasilania od 3,3V do 5V a czwarty pin do uziemienia. Drugi pin jest odpowiedzialny za wysyłanie danych cyfrowych. Czujnik został podłączony do 3 pinowego modułu który został przedstawiony na rysunku 2.3 wraz z czujnikiem. Pierwszy pin jest odpowiedzialny za dodatnie napięcie a 3 za ujemne napięcie. Pin nr. 2 został wykorzystany do wysyłania cyfrowych danych o temperaturze i wilgotności. Aby móc odczytać dane należy zasilić czujnik i podłączyć drugi pin do cyfrowego wyjścia w mikrokontrolerze.

³<https://www.electrokit.com/uploads/productimage/41015/41015728-1.jpg>



Rysunek 2.3: Czujnik temperatury i wilgotności powietrza DHT11³

2.2 Silnik krokowy

Silnik krokowy jest silnikiem którego wał dokonuje obrotu o określony kąt. Aby dokonać obrotu należy wysyłać impulsy elektryczne w określonej kolejności. Z tego powodu silniki krokowe pozwalają na dokonanie dokładniejszych operacji od serwomechanizmów. To właśnie ta cecha czyli przeważająca dokładność zadecytoała o wyborze mechanizmu obracającego koła zębate. Kolejną cechą która zaważyła o wyborze silnika krokowego jest możliwość utrzymania obciążenia podczas postoju bez potrzeby zasilania. W przypadku serwomechanizmu też można osiągnąć ten efekt ale tylko poprzez zasilenie mechanizmu. Jednak potrzeba dokładnych obrotów oraz ich dokładność ostatecznie zadecydowała o wyborze silnika krokowego.

2.2.1 28BYJ-48 5V

W projekcie zdecydowano się wykorzystać silnik krokowy 28BYJ-48 który został przedstawiony na rysunku 2.4. O wykorzystaniu tego komponentu zadecydowała jego wielkość oraz niskie koszty zakupu. Rozmiary silnika wynoszą 42mm na 31

mm czyli wystarczająco małe wymiary by wpasować się do zminimalizowanego rozmiaru modelu prototypu. Wał posiada odpowiednie ścięcia ułatwiające montaż komponentów a jego średnica wynosi 5mm. Na wale silnika będą montowane specjalnie przygotowane do tego celu zębatki.

Do zasilenia silnika potrzeba napiecia 5V co umożliwia zasilanie bezpośrednio z płytki Arduino Uno. Przełożenie przekładni tego urządzenia wynosi 64:1. Silnik składa się z 5 przewodów o różnych kolorach z których każdy reprezentuje swoją funkcję. Czerwony przewód służy do wyprowadzenia zasilania 5V do silnika. Pozostałe przewody zostały użyte do sterowania impulsami przechodzącymi przez cewki co umożliwia wprawienie wału w ruch przy wprowadzeniu odpowiedniej sekwencji sygnałów. Wszystkie przewody są zakończone końcówką umożliwiającą podłączenie komponentu do sterownika co można zaobserwować na końcach kalbi na rysunku 2.4.

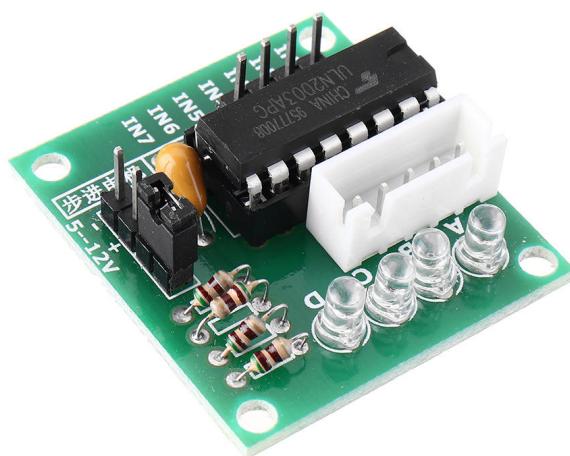


Rysunek 2.4: Silnik krokowy 28BYJ-48 5V⁴

⁴https://www.makerfabs.com/image/cache/makerfabs/28BYJ-48\%20Stepper\%20motor-5V/28BYJ-48\%20Stepper\%20motor-5V_1-1000x750.JPG

2.2.2 Sterownik silnika krokowego ULN2003

Do sterowania impulsami w silniku krokowym wykorzystano sterownik silnika krokowego ULN2003 który został przedstawiony na rysunku 2.5. Napięcie zasilania wynosi do 12V a części logicznej 5V. Sterownik składa się z 4 pinów wejściowych IN, złącza pozwalającego na proste połaczenie silnika, 4 diod led sygnalizacyjnych oraz 2 pinów do zasilania.



Rysunek 2.5: Sterownik silnika krokowego ULN2003⁵

Aby zasilić sterownik należy na pinach zasilania podłączyć dodatnie napięcie do pinu oznaczonego znakiem plusa a uziemienie do pinu oznaczonego minusem. Do tak zasilonego sterownika należy podłączyć przewody silnika krokowego do przygotowanego złącza. Na koniec pozostaje podłączyć przewody do czterech pinów IN które będą wysyłały sygnały wskazujące na kolejność przekazywania

⁵<https://imgaz3.staticbg.com/thumb/large/oaupload/ser1/banggood/images/6A/09/028eb2d8-c64f-4411-ae11-953fda62e7af.jpeg>

impulsów do cewek. W momencie przekazywania sygnału diody led będą komunikowały odebrany sygnał za pomocą podświetlenia. Sygnały przekazywane w prawidłowy sposób wprowadzą w ruch wał silnika w ustalonym kierunku.

2.3 Mikrokontroler

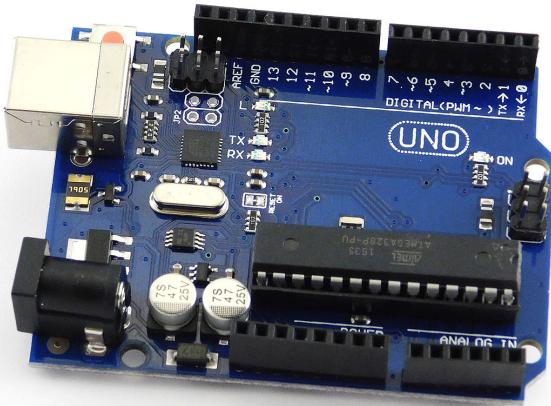
Mikrokontrolery to urządzenia dające możliwość sterowania urządzeniami podłączonymi do niego dzięki określonej logice. Dzięki wyprowadzeniom goldpin w urządzeniu za pomocą przewodów podłączeniowych możemy podłączyć inne urządzenia a dzięki połączeniu odbierać i wysyłać sygnały.

W projekcie zostało użyte urządzenie zgodne z platformą Arduino. Arduino jest platformą typu open-source. Zaletą urządzeń zgodnych z arduino jest to że nie wymagają one zewnętrznego programatora co oznacza że aby wprowadzić program, urządzenie należy podłączyć przewodem USB do komputera i wgrać program. Programy na tego typu urządzenia pisane są w języku programowania Arduino. Jest to język oparty o C++ ale rozwinięty o rozwiązania umożliwiające komunikacje z urządzeniami.

2.3.1 Arduino Uno

Mikrokontrolerem który został użyty w prototypie jest Arduino Uno przedstawiony na rysunku 2.6. Jest to jeden z najpopularniejszych modułów do nauki elektroniki i programowania na rynku. Płytki składa się z dwóch 8 bitowych mikrokontrolerów, sześciu złączy analogowych i czternastu złączy cyfrowych. Płytki posiada również przycisk RESET umożliwiający ponowne uruchomienie płytka. Do programowania płytka wykorzystano złącze USB typu B. Podłączenie do komputera na płytce sygnalizowane jest miganiem stworzonej do tego diody. Napięcia na pinach wynoszą 5V.

⁶<https://nettigo.pl/system/images/1995/original.jpg>



Rysunek 2.6: Mikrokontroler Arduino UNO⁶

Z powodu konieczności pracy z dużą ilością czujników w celu odbioru wartości z sygnałów korzysta się ze złączy analogowych. Złącza analogowe pozwalają mierzyć napięcie. Rozdzielcość pomiaru wynosi 10 bitów. Dzięki takiej rozdzielcości jesteśmy w stanie określić 1024 poziomów. Znając wartość maksymalnego napięcia jakie może przyjąć złącze czyli 5V i ilość poziomów pozwala określić że dokładność pomiarów jest bliska 0.005V. Duża ilość złączy i możliwość wczytywania z każdego złącza wartości pozwoliło na wygodny odczyt danych z każdego czujnika.

2.4 Zasilanie

W układzie należy zasilić moduł arduino, czujniki oraz silniki krokowe. Do zasilenia płytka arduino można wykorzystać przewód USB typu B lub zasilacz o napięciu od 7V do 12V. Do zasilenia można również użyć baterii.

Czujniki podłączone do mikrokontrolera można zasilić bezpośrednio z urządze-

dzenia podłączając zasilanie czujników do odpowiednich pinów. W przypadku zasilania silników krokowych porządane jest podłączenie ich do zewnętrznego zasilania takiego jak bateria czy akumulator ponieważ jesteśmy w stanie wyprowadzić większe napięcia. W przypadku silnika krokoowego jesteśmy w stanie wyróżnić zasilanie oraz część odpowiedzialną za logike. Sterownik silnika krokoowego będzie przekazywał odpowiednie napięcie potrzebne do zasilenia silnika oraz będzie wysyłał sygnały logiczne odpowiedzialne za wprowadzenie walu silnika w ruch.

Podsumowując do zasilenia całego obwodu będą potrzebne dwa źródła zasilania:

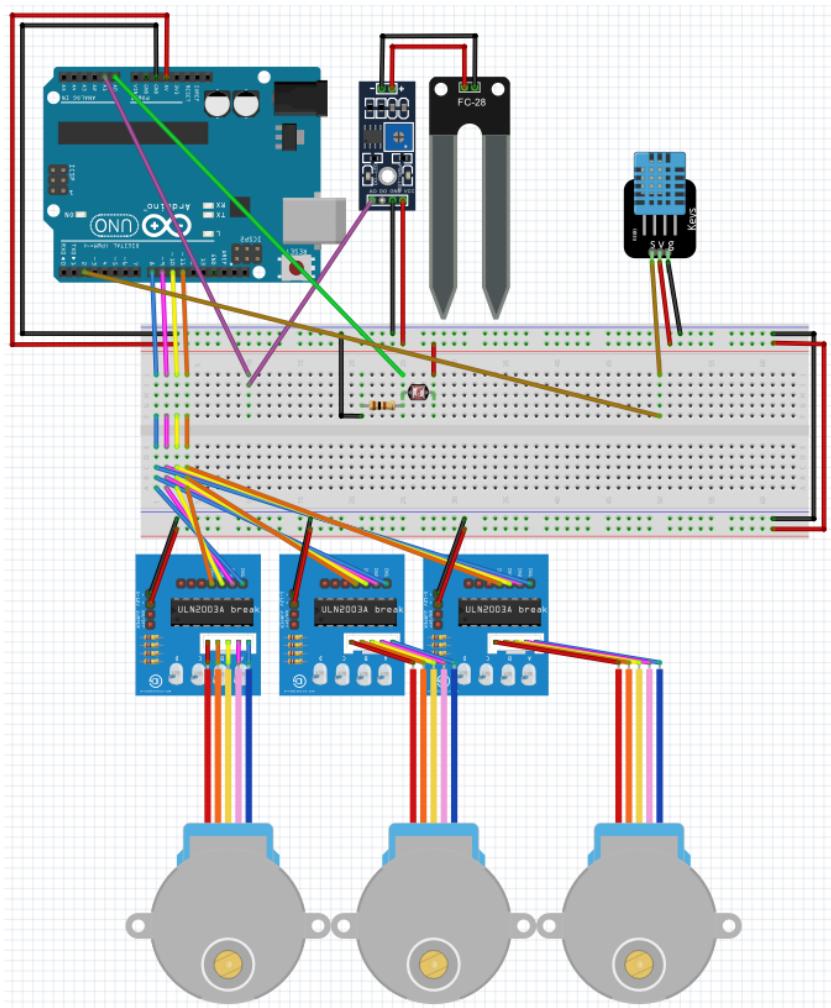
- Zasilanie modułu i czujników,
- Zasilanie sterowników i silników krokowych,

2.5 Układ

Ostatecznie na całe złożenie układu składają się takie komponenty jak:

- Mikrokontroler: Arduino UNO,
- Czujnik wilgotności gleby: HL-69,
- Czujnik wilgoci i temperatury powietrza: DHT11,
- Czujnik natężenia światła: fotorezystor,
- 3 sterowniki silników krokowych: ULN2003,
- 3 silniki krokoowe 28BYJ-48,
- Zasilanie (pewnie bateria)

Na rysunku 2.7 zaprezentowano przykładowe podłączenie komponentów za pomocą mikrokontrolera Arduino Uno. W kolejnym rozdziale zostało opisane jak wygląda przepływ sygnałów między komponentami oraz jak urządzenia obsługiwane są w kodzie arduino.



Rysunek 2.7: Przykładowe podłączenie komponentów do Arduino Uno

Rozdział 3

Oprogramowanie

Mikrokontrolery oparte na standardzie arduino pozwalają na wysyłanie sygnałów oraz ich obsługę podczas odbierania za pomocą języka programowania Arduino. Jest to język oparty na języku programowania C++. Zawiera on również wbudowaną bibliotekę pozwalającą na obsługę wejścia i wyjścia na pinach w płytce. Takie rozwiązanie umożliwia w prosty sposób obsługę odbieranych sygnałów z czujników i wyprowadzanie sygnałów na urządzenia według zdefiniowanej w kodzie logiki.

W powyższym projekcie wykorzystano urządzenia zgodne ze standardem arduino a co za tym idzie przygotowano oprogramowanie w arduino. Rozdział ten został poświęcony wykorzystanemu oprogramowaniu w projekcie. Poniżej przedstawiono informacje o tym jaki algorytm został wykorzystany do sterowania urządzeniem, wykorzystane biblioteki oraz logika związana z komunikacją między czujnikami, mikrokontrolerem i urządzeniami.

3.1 Przepływ sygnałów

W opisie przepływu sygnałów należy uwzględnić mikrokontroler który jest odpowiedzialny za logikę działania urządzenia oraz urządzenia do niego podłączone. W procesie komunikacji między mikrokontrolerem i podłączonymi do niego urządze-

dzeniami można wyróżnić 2 typy urządzeń:

- Urządzenia wyjściowe czyli urządzenia do których mikrokontroler wysyła sygnał
- Czujniki czyli urządzenia z których mikrokontroler odczytuje sygnał,

Wśród urządzeń wyjściowych w projekcie można wyróżnić 3 silniki krokowe które będą wprawiały komponenty w ruch oraz 3 sterowniki silników krokowych które będą odpowiedzialne za sterowanie ruchem wałów silnika.

Wśród czujników które będą wysyłały sygnały do mikrokontrolera znajdują się 3 urządzenia takie jak:

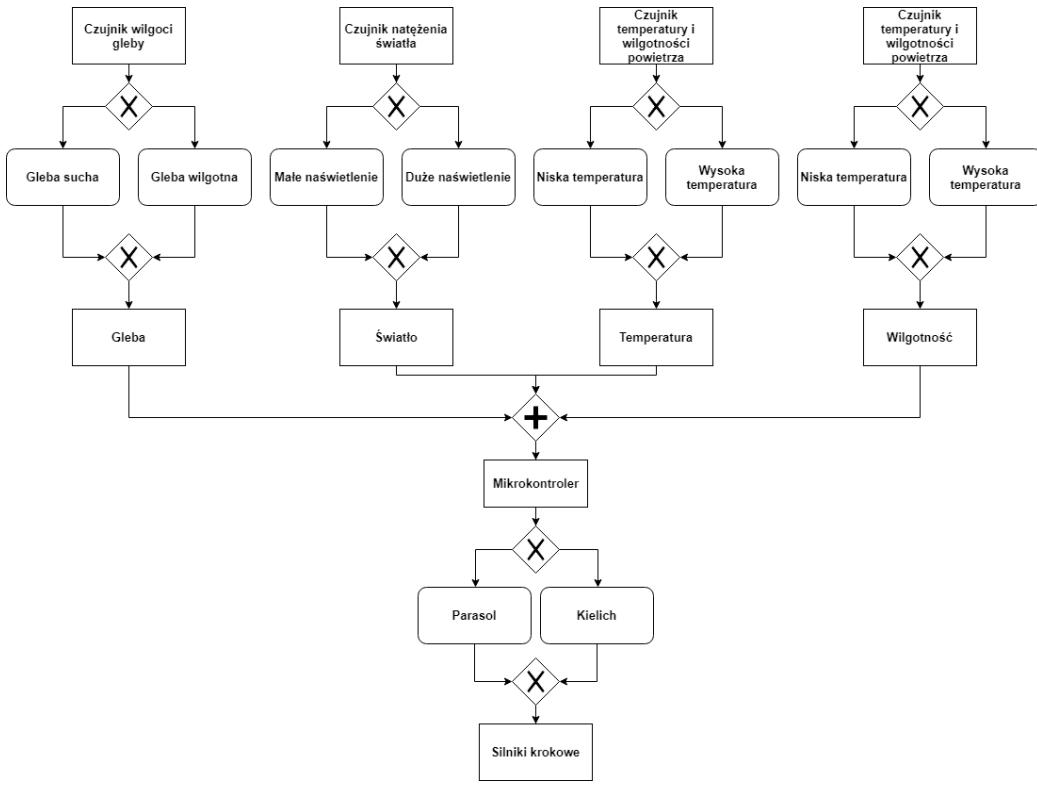
- Czujnik wilgotności gleby,
- Czujnik natężenia światła,
- Czujnik temperatury i wilgotności powietrza,

Na podstawie informacji uzyskanych z czujników, mikrokontroler podejmuje decyzję w jakim stanie ma znajdować się urządzenie i wysyła sygnał do sterowników silników krokowych aby wykonały obrót o określony kąt. Uproszczony schemat komunikacji został przedstawiony na rysunku 3.1.

3.2 Stany urządzenia i badanych elementów

Urządzenie może znajdować się w jednym z dwóch stanów to jest parasola i kielicha. Stan urządzenia jest zależny od odczytów jakie mikrokontroler uzyska z czujników.

Każdy z czujników wskazuje na stan odpowiednio badanego przez niego elementu. Odczyt z każdego czujnika został podzielony na dwie części za pomocą z



Rysunek 3.1: Uproszczony diagram ukazujący przepływ sygnałów

góry ustalonej wartości granicznej. Dzięki takiemu rozwiązaniu z każdego urządzenia uzyskujemy dwa stany badanego elementu.

Ponieważ czujnik temperatury i wilgotności powietrza dokonuje pomiaru temperatury i powietrza bada on 2 elementy. Takie podejście daje 4 badane elementy z czego każdy element może znajdować się w 2 stanach.

Aby sprawdzić liczbę możliwych kombinacji należy użyć równania s^e gdzie liczba s jest liczbą stanów w jakich może znajdować się urządzenie a liczba e jest liczbą badanych elementów. Po podłożeniu odpowiednich wartości pod symbole otrzymano równanie 2^4 co w wyniku daje 16 możliwych kombinacji odczytów z czujników. Kombinacje wszystkich możliwych odczytów z czujników oraz wynikające z nich stany urządzenia zostały przedstawione na rysunku 3.2 z tabelą kombinacji odczytów i stanów urządzenia.

Gleba	Naświetlenie	Temperatura	Wilgotność powietrza	Stan
Sucha	Niskie	Niska	Niska	Parasol
Sucha	Niskie	Niska	Wysoka	Parasol
Sucha	Niskie	Wysoka	Niska	Kielich
Sucha	Niskie	Wysoka	Wysoka	Kielich
Sucha	Wysokie	Niska	Niska	Parasol
Sucha	Wysokie	Niska	Wysoka	Kielich
Sucha	Wysokie	Wysoka	Niska	Kielich
Sucha	Wysokie	Wysoka	Wysoka	Kielich
Mokra	Niskie	Niska	Niska	Parasol
Mokra	Niskie	Niska	Wysoka	Parasol
Mokra	Niskie	Wysoka	Niska	Kielich
Mokra	Niskie	Wysoka	Wysoka	Kielich
Mokra	Wysokie	Niska	Niska	Parasol
Mokra	Wysokie	Niska	Wysoka	Parasol
Mokra	Wysokie	Wysoka	Niska	Parasol
Mokra	Wysokie	Wysoka	Wysoka	Kielich

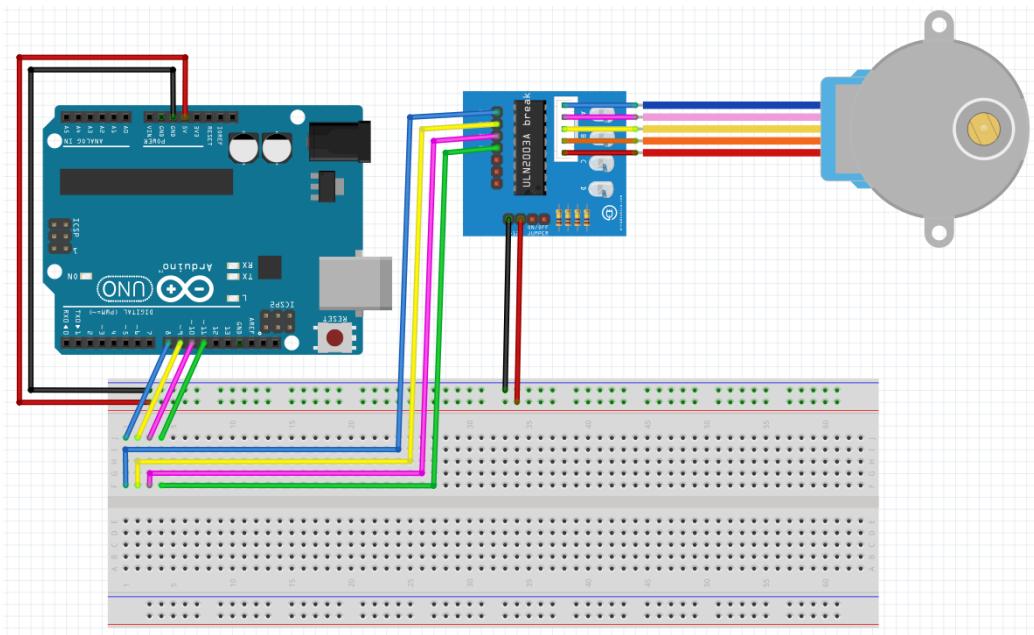
Rysunek 3.2: Tabela kombinacji stanów elementów i urządzenia

3.3 Obsługa silników krokowych

Do obsługi silników krokowych wykorzystano bibliotekę "Stepper[2]" w wersji 1.1.3 stworzoną przez firmę Arduino. Biblioteka ta umożliwia obsługę jednogebiernowych i dwubiegowych silników krokowych. Aby użyć biblioteki należy użyć nagłówka #include <Stepper.h>.

Do zainicjowania silnika krokkowego w kodzie wykorzystano funkcję Stepper(steps, pin1, pin2, pin3, pin4). Funkcja ta tworzy instancję klasy Stepper która jest reprezentacją podłączonych do urządzenia silników krokowych i zwraca ją po wywołaniu. Istnieją dwie funkcje Stepper jedna z 3 parametrami oraz druga z 5 parametrami. W powyższym przypadku wykorzystano funkcję z pięcioma parametrami ponieważ sterownik silnika został podłączony za pomocą 4 pinów do

Arduino. Pierwszym parametrem w funkcji jest liczba całkowita reprezentująca liczbę kroków przypadających na jeden obrót silnika. W przypadku wykorzystanych silników jest to 2048 kroków na obrót. Pozostałe parametry są to liczby całkowite reprezentujące piny na mikrokontrolerze. Piny te są wykorzystywane do sterowania silnikami krokowymi i są łączone ze sterownikiem do wejść podpisany jako IN1, IN2, IN3, IN4. W przypadku schematu zaprezentowanego na rysunku 3.3 piny sterujące 8, 9, 10, 11 na mikrokontrolerze zostały odpowiednio podłączone do wejść IN1, IN2, IN3, IN4.



Rysunek 3.3: Przykładowy schemat podłączenia silnika krokowego 28BYJ-48 ze sterownikiem ULN2003A

Po uzyskaniu instancji silnika przed wprawieniem go w ruch należy ustawić jego prędkość. Aby tego dokonać należy na instancji obiektu wywołać funkcję setSpeed(rpms) przyjmującą jeden parametr. Jako parametr należy wprowadzić dodatnią liczbę zmiennoprzecinkową reprezentującą liczbę obrotów na minutę czyli RPM (z ang. Rotations Per Minute). Funkcja ta jest wywoływana w funkcji

setup() która jest uruchamiana przed funkcją loop() na początku po uruchomieniu lub zresetowaniu mikrokontrolera.

Do wprowadzenia silnika w ruch wykorzystywana jest funkcja step(steps). Funkcja ta jest wywoływana na instancji reprezentującej silniki krokowe. Jako parametr przyjmowana jest liczba całkowita reprezentująca liczbę kroków którą silnik powinien wykonać po wywołaniu funkcji. Jeśli liczba przyjęta w parametrze jest dodatnia silnik obróci się w jedną stronę a gdy przyjęta wartość jest ujemna silnik dokona obrotu w przeciwnym kierunku. Funkcja ta jest blokująca. Oznacza to że urządzenie podczas wykonania kodu nie przejdzie do kolejnej linii do momentu gdy silnik nie skończy się obracać.

Aby dokonać obrotu o określony kąt utworzono funkcję obrocSilnik(kat) przyjmującą jeden parametr. Parametrem jest liczba zmienoprzecinkowa reprezentująca kąt w stopniach. Aby zmienić kierunek obrotu podana liczba powinna mieć ujemną wartość. Aby dokonać obrotu o określony kąt funkcja wywołuje metodę step(steps) na instancji reprezentującej silniki. Parametr funkcji step odpowiada liczbie kroków jaką należy dokonać aby silnik obrócił się o określony kąt podany w parametrze metody obrocSilnik. Aby obliczyć liczbę kroków przypadającą na określony kąt należy obliczyć następujące równanie: $l = r/(360/k)$ gdzie l to liczba kroków przypadająca na określony kąt, r to liczba kroków na jeden pełny obrót silnika a k to kąt podany w parametrze funkcji.

3.4 Obsługa czujnika wilgotności i temperatury powietrza

Do pobrania pomiarów z czujnika DHT11 wykorzystano bibliotekę "DHT sensor library[1]" w wersji 1.3.10 stworzoną przez firmę Adafruit. Jest to biblioteka umożliwiająca dokonanie pomiarów z czujników typu DHT11, DHT21 oraz

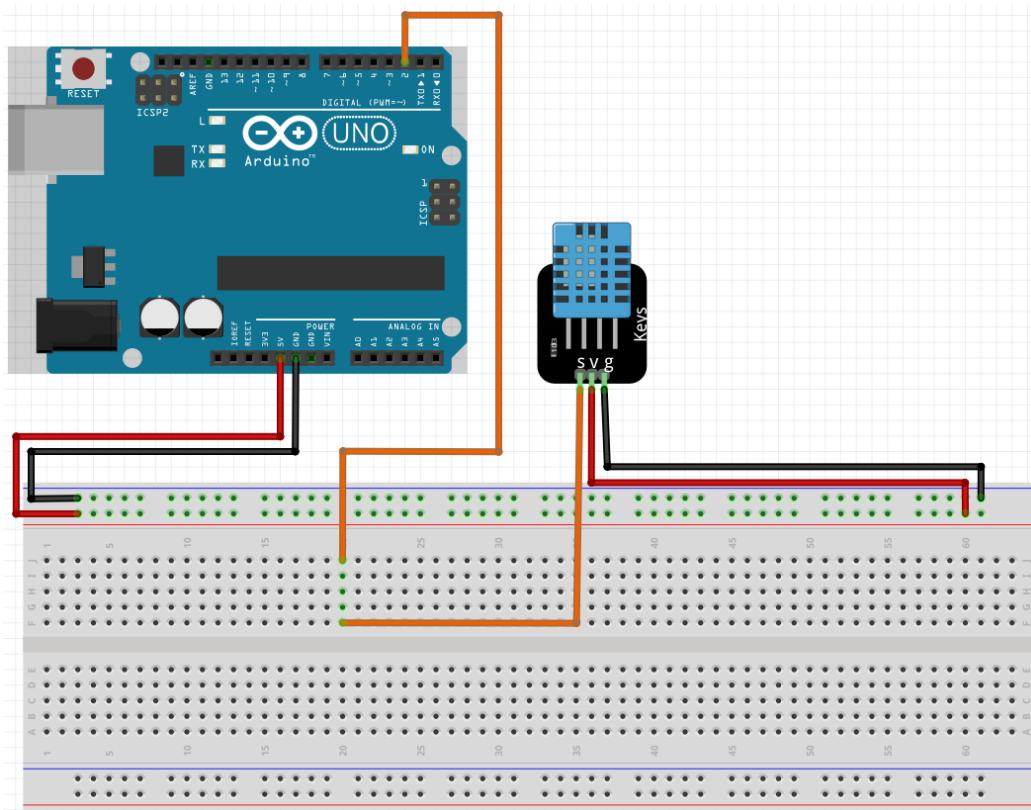
DHT22. Aby móc korzystać z wspomnianej biblioteki w projekcie należy do projektu dodać bibliotekę o nazwie Adafruit Unified Sensor"która również została stworzona przez firmę Adafruit i jest pewnego rodzaju warstwą abstrakcją używaną przez wiele bibliotek czujników Adafruit. Aby użyć biblioteki w kodzie należy wprowadzić nagłówek #include "DHT.h".

Aby rozpocząć prace z biblioteką należy wywołać metodę DHT(pin, type) przyjmującą dwa parametry. Pierwszym parametrem jest pin na mikrokontrolerze umożliwiający na odczyt wartości. W przedstawionym przykładzie na rysunku numer 3.3 pinem podłączonym do urządzenia i podawanym w pierwszym parametrze jest pin numer 2. Drugim parametrem jest typ urządzenia, w powyższym przypadku DHT11. Metoda zwraca instancje reprezentującą czujnik DHT. Aby zainicjować urządzenie należy wywołać metodę begin() na instancji zwrotnej przez metodę DHT.

Aby uzyskać informacje o temperaturze należy wywołać funkcję readTemperature() która zwróci przybliżoną temperaturę w miejscu pomiaru podaną w stopniach celsjusza. Błąd pomiaru może jednak wynosić $\pm 2^{\circ}C$ wiec uzyskane wartości są tylko przybliżeniem rzeczywistego stanu temperatury.

Do odczytania wartości wilgotności powietrza wykorzystano metodę readHumidity(). Funkcja ta zwraca przybliżoną wartość względnej wilgotności powietrza w miejscu pomiaru podaną w procentach gdzie im wartość bliższa 100% tym nasycenie powietrza parą wodną jest wyższe. Błąd pomiaru w czujniku może wynosić $\pm 5\%RH$ (RH – z ang. Relative Humidity) wiec uzyskany pomiar jest tylko przybliżeniem rzeczywistej względnej wilgotności powietrza.

Ostatnim parametrem jaki można uzyskać przy pomocy czujnika oraz opisywanej biblioteki jest indeks ciepła. Indeks ciepła potocznie znany jest również pod nazwą temperatura odczuwalna. Aby dokonać odczytu wartości należy wywołać funkcję computeHeatIndex(t, h, true) która przyjmuje trzy parametry. Pierw-



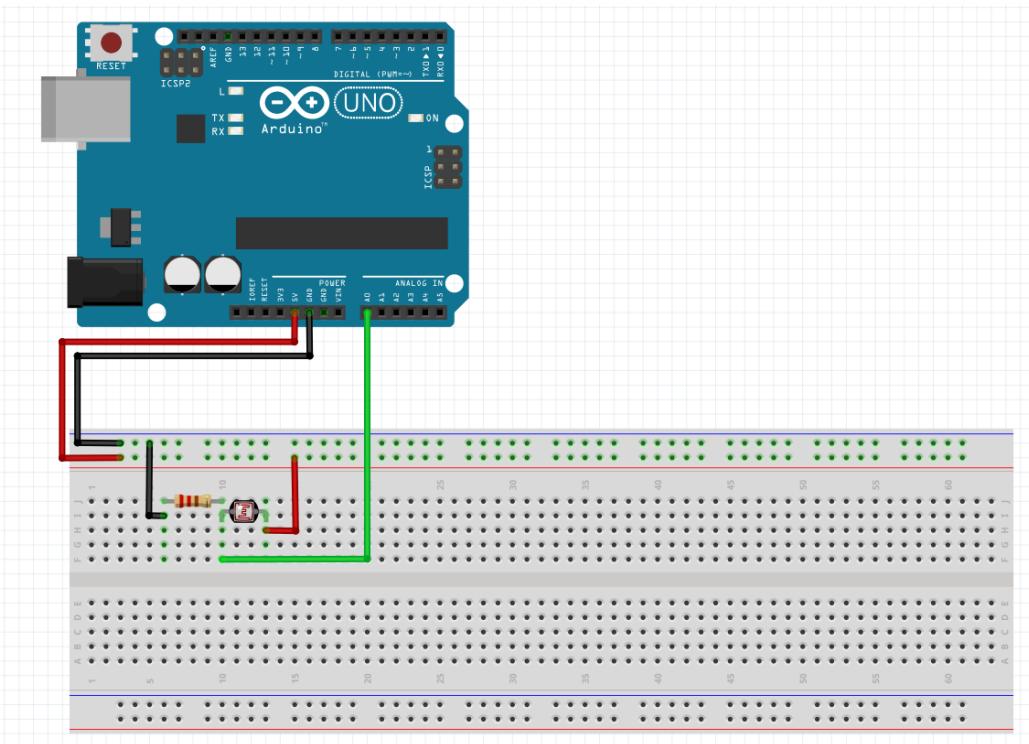
Rysunek 3.4: Przykładowy schemat podłączenia czujnika temperatury i wilgotności powietrza typu DHT11

szym parametrem jest temperatura. Drugim parametrem jest wilgotność powietrza. Ostatni parametr jest parametrem opcjonalnym i przyjmuje wartość logiczną true (prawda) lub false (fałsz). Domyślnie wartość ustawiona jest na true i oznacza że wynik będzie zwrócony w stopniach fahrenheitita. W przypadku parametru ustawionego na false zwracana wartość będzie w stopniach celsjusza.

3.5 Obsługa fotorezystora

Do obsługi pomiaru sygnałów z fotorezystora wykorzystano standardowe funkcje wbudowane w język arduino[3]. Aby dokonać odczytu z fotorezystora musi być on podłączony do pinu na mikrokontrolerze umożliwiającego odczyt wartości.

ści analogowych. W przedstawionym przykładzie podłączenia fotorezystora na rysunku numer 3.5 dane są odbierane przez kabel na pinie A0.



Rysunek 3.5: Przykładowy schemat podłączenia fotorezystora w celu pomiaru natężenia światła

Przed rozpoczęciem pomiaru należy skonfigurować odpowiednio pin. Służy do tego metoda `pinMode(pin, mode)` przyjmującą dwa parametry. Pierwszym parametrem jest numer pinu. Drugim parametrem jest tryb działania pinu. W powyższym przypadku tryb działania pinu został ustawiony na "INPUT" w celu odczytu wartości na pinie. Funkcja ta jest wywoływana w metodzie `setup()` która wykonuje się przed funkcją `loop()` i służy konfiguracji urządzenia przed rozpoczęciem działania.

Aby odczytać wartość należy wywołać funkcję `analogRead(pin)` przyjmującą jeden parametr. Parametrem jest pin z którego odczytywana jest wartość. Od-

czytując wartość z fotorezystora można określić natężenie światła w miejscu pomiaru. Im niższa odczytana wartość na pinie tym mniejsze natężenie światła padającego na fotorezystor.

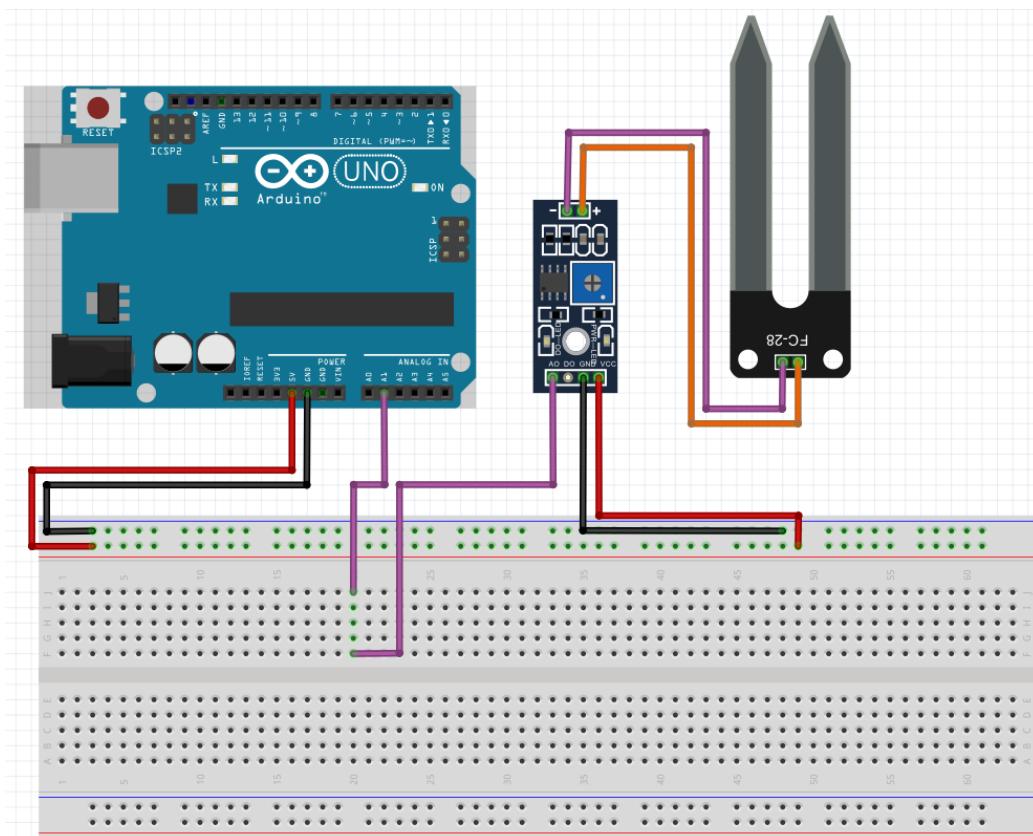
3.6 Obsługa czujnika wilgotności gleby

Do obsługi pomiaru sygnałów z czujnika wilgotności gleby wykorzystano standardowe funkcje wbudowane w język arduino[3]. W celu odczytania wartości z czujnika hl-69 należy płytke podłączyć do czujnika oraz czujnik podłączyć do sondy. Czujnik można podłączyć na dwa sposoby:

- Pin cyfrowy umożliwiający odczyt jednego z 2 stanów
- Pin analogowy umożliwiający odczyt reprezentującej wilgotność gleby (jak przedstawiono na rysunku numer 3.6)

Przed odczytem należy dokonać konfiguracji pinu z którego odczytywana jest wartość. W tym celu w metodzie `setup()` konfigurującej mikrokontroler przed uruchomieniem należy wywołać funkcje `pinMode(pin, mode)` przyjmującą dwa parametry. W pierwszym parametrze należy wprowadzić numer pinu z którego odczytywana jest wartość a w drugim tryb działania czyli w tym przypadku INPUT.

W celu odczytu wartości z czujnika należy wywołać jedną z dwóch funkcji w zależności od sposobu pomiaru z czujnika. W przypadku pomiaru z pinu cyfrowego należy wywołać funkcje `digitalRead(pin)` przyjmującą jeden paramet będący numerem pinu z którego dokonywany jest pomiar. Funkcja w odpowiedzi zwróci jeden z dwóch stanów: LOW (stan niski) lub HIGH (stan wysoki). Stan pomiaru jest zależny od wartości granicznej ustawionej na czujniku. Jeśli odczytana wartość wilgotności gleby będzie większa od wartości granicznej w wyniku



Rysunek 3.6: Przykładowy schemat podłączenia czujnika wilgotności gleby HL-69

czujnik zwróci stan wysoki. W przeciwnym przypadku odczytana wartość na pinie jest równa stanu niskiemu. Aby zmienić wartość graniczną należy obrócić potencjometr znajdujący się na czujniku w odpowiednią stronę.

W przypadku pomiaru wartości z czujnika na pinie analogowym należy wywołać funkcję `analogRead(pin)` przyjmującą jeden parametr reprezentujący numer pinu z którego odczytywana jest wartość. W momencie gdy wilgotność gleby się zwiększa odczytana wartość z czujnika zmalaże. W przeciwnym przypadku wartość na czujniku jest wyższa.

3.7 Algorytm sterowania

Kod arduino[3] składa się z dwóch najważniejszych funkcji czyli setup() oraz loop(). Funkcja setup jest wywoływana przy uruchomieniu lub zresetowaniu urządzenia. Jest to pierwsza wywoływana funkcja i jest ona wywoływana tylko raz w celu konfiguracji urządzenia. W powyższym projekcie została ona wykorzystana do ustawienia trybu działania pinów na urządzeniu oraz zainicjowania czujnika dht i ustawienia prędkości silników krokowych.

Funkcja loop jest funkcją która wykonuje swój kod w pętli. W projekcie została ona wykorzystana do zebrania pomiarów z czujników i zmianiania stanów urządzenia poprzez wywołanie funkcji rozpocznijPomiary(). Funkcja ta została wykorzystana do wywołania w ustalonej kolejności funkcji:

- rozpocznijDHT11()
- rozpocznijHL69()
- rozpocznijFotorezystor()
- podejmijDecyzje()

Trzy pierwsze funkcje zostały wykorzystane do zbierania wartości z czujników oraz wyświetlenia tych wartości na konsoli. Sposób odczytywania danych z czujników został opisany w podrozdziałach: 3.4, 3.5, 3.6.

Metoda podejmijDecyzje() jest odpowiedzialna za ustawienie stanu urządzenia na podstawie wartości odczytanych z czujników. Urządzenie może znajdować się w jednym z 2 stanów:

- Parasol
- Kielich

Stany te w kodzie zostały przechowane w zmiennej logicznej o nazwie "czySilnikWysoko". Zmienna ta może przyjąć jeden z dwóch stanów to jest true (prawda) lub false (fałsz).

Funkcja podejmijDecyzje() na samym początku zapisuje poprzedni stan zmiennej "czySilnikWysoko" w zmiennej "poprzedniStan". Następną czynnością jest wywołanie funkcji dokonajPorownanWartosciZCzujnikow(). Funkcja ta odpowiedzialna jest za porównanie parametrów odczytanych czujników z ustalonimi dla nich wartościami granicznymi. Na potrzeby funkcji utworzono czteroelementową tablicę w której pod konkretnym indeksem przechowywany jest wynik każdego z porównania odpowiednio:

- 0 - wilgotność gleby
- 1 - natężenie światła
- 2 - temperatura powietrza
- 3 - wilgotność powietrza

Kolejnym krokiem wykonywanym w funkcji podejmijDecyzje() jest wywołanie funkcji ustawStan(wynik) z jednym parametrem. Przyjmowany parametr jest wcześniej utworzoną cztero elementową tablicą której wartości zostały przypisane w poprzednio omawianej funkcji. Zadaniem funkcji "ustawStan" jest sprawdzenie wartości na konkretnych indeksach i ustawienie stanu maszyny według logiki przedstawionej na rysunku numer 3.2. Aby ograniczyć liczbę porównań wyliczono że najmocniej wpływającym parametrem na stan urządzenia jest temperatura. Zwróciło uwagę że gdy temperatura jest niska to w 7 na 8 przypadkach stan urządzenia będzie ustawiony na "parasol". Z tego powodu w przypadku gdy temperatura jest niska sprawdzany jest tylko jeden przypadek który sprawdzi czy stan urządzenia to "kielich" w przeciwnym przypadku ustawią stan urządzenia na

”parasol”. W przypadku gdy temperatura jest wysoka również zaobserwowano że w 7 na 8 przypadków stan urządzenia to będzie ”Kielich”. Z tego powodu analogicznie jak w poprzednim przypadku sprawdzana jest tylko jedna kombinacja stanów odczytanych z czujników która powinna zwrócić stan ”Parasol”. Jesli wynik nie zgadza się ze sprawdzaną kombinacją ustawiany jest stan ”Kielich”.

W kolejnym kroku wywoływana jest funkcja obrocZgodnieZDecyzja(czy-SilnikWysoko, poprzedniStan) przyjmująca 2 parametry. Pierwszym parametrem jest nowy stan urządzenia obliczony na podstawie wartości odczytanych z czujników. Drugi parametr to poprzedni stan urządzenia który został zapisany na początku funkcji podejmijDecyzje().

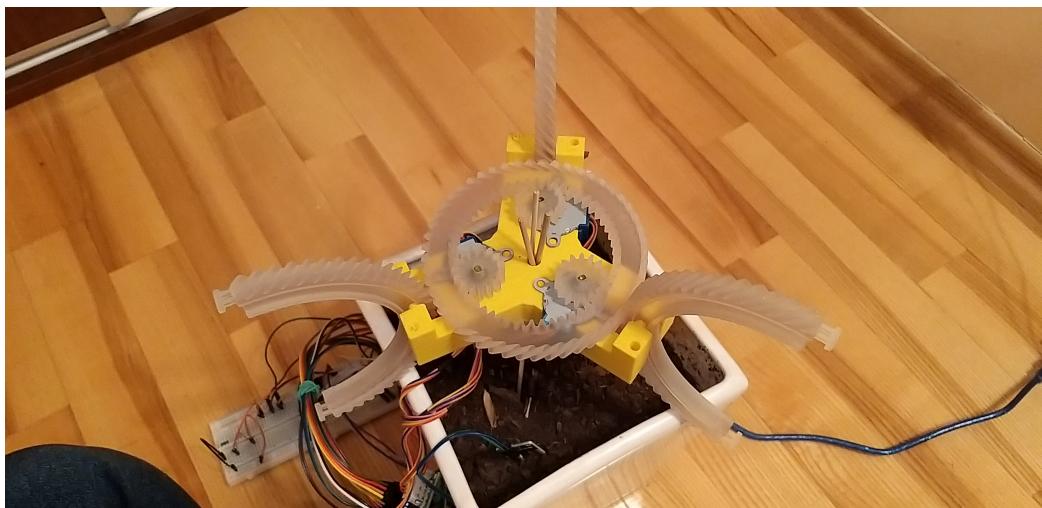
Pierwszą czynnością dokonywaną w funkcji ”obrocZgodnieZDecyzja” jest porównanie nowego stanu urządzenia z poprzednim stanem. Jeśli stany są takie same funkcja kończy swoje działanie. Jeśli jednak stany się między sobą różnią, w kolejnej instrukcji warunkowej sprawdzany jest nowy stan i na podstawie tego stanu wywoływana jest metoda ”obrocSilnik” z dodatnim albo ujemnym parametrem reprezentującym kąt obrotu. Metoda obróć silnik dokonuje obrotu silnika o określony kąt zgodnie z formułą opisaną w podrozdziale 3.3.

Rozdział 4

Podsumowanie Projektu

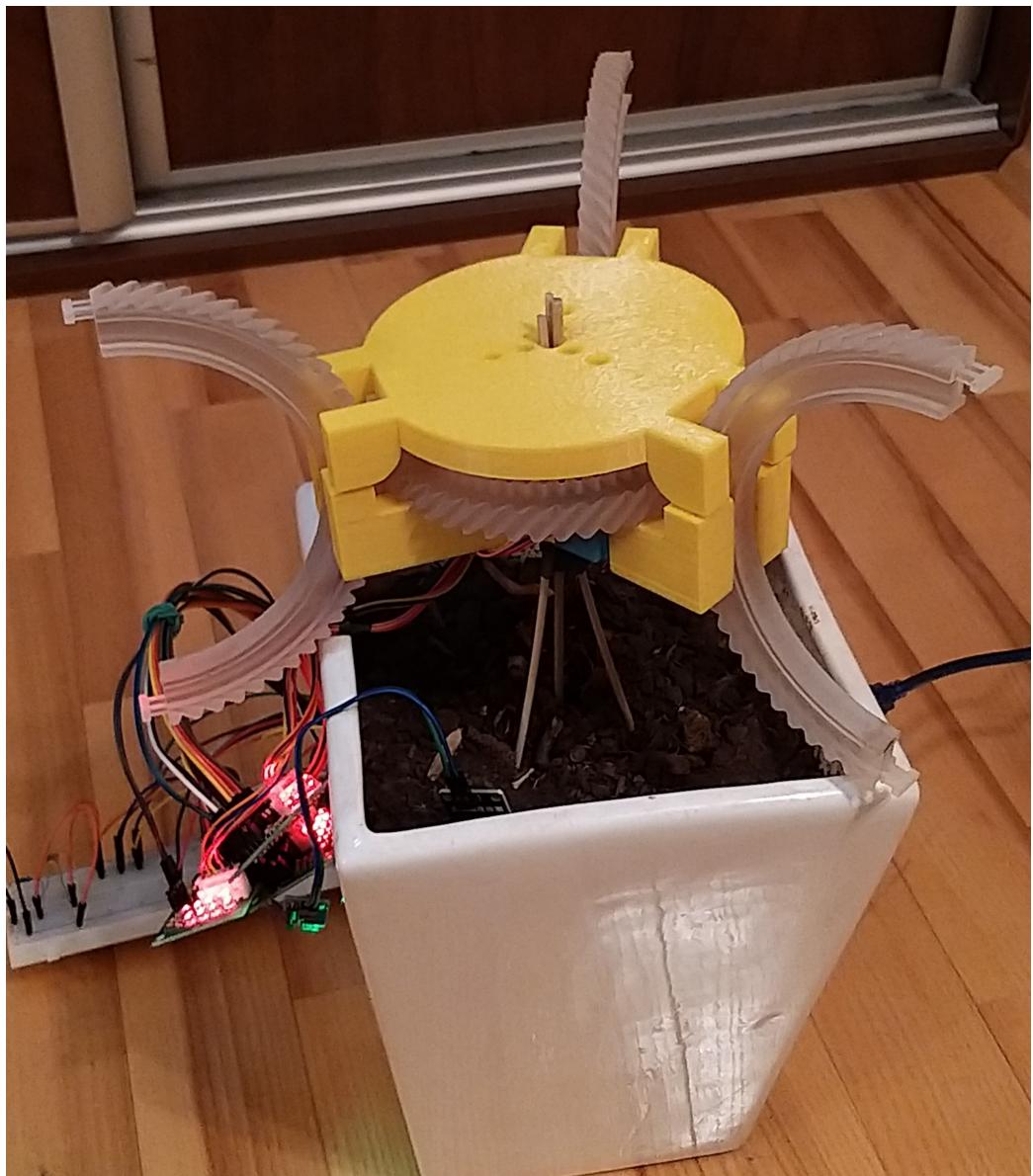
W powyższej pracy zostały zaprezentowane utworzone modele części przygotowanych na potrzeby utworzenia urządzenia. Modele zostały przygotowane w programie Solidworks. Przygotowane złożenie zostało przetestowane za pomocą modułu Solidworks motion. Po wyborze jednego z dwóch przygotowanych projektów urządzenia, dokonano ponownego zaprojektowania części aby zminimalizować rozmiar części a co za tym idzie zmniejszyć koszty produkcji prototypu. Model wymagał ponownego przeprojektowania z powodu konieczności dostosowania go do silników krokowych o mniejszym rozmiarze. Przygotowane modele części zostały przekonwertowane do formatu STL. Tak przygotowane pliki umożliwiły wydruk części w technologii druku 3D. Na rysunku numer 4.1 przedstawiono wydrukowane części znajdujące się na prototypie.

Na potrzeby projektu dobrano odpowiednie czujniki do pomiaru warunków atmosferycznych. Projekt wymagał również wykorzystania silników krokowych oraz sterowników w celu wprowadzenia w ruch urządzenia. Ostatnim dobranym do projektu komponentem jest mikrokontroler który został wykorzystany do odczytywania sygnałów z czujników oraz sterowania silnikami krokowymi. W rozdziale 2, opisano metodę działania każdej z wykorzystanych części oraz sposób połączenia ich ze sobą.



Rysunek 4.1: Prototyp bez założonej osłony

W ostatniej części pracy opisano przepływy sygnałów oraz logikę działania urządzenia. Logika działania została opisana w tekście oraz przygotowanych tabelach i schematach. Na potrzeby realizacji projektu, przygotowano oprogramowanie realizujące założenia opisanej logiki. W kolejnej części pracy opisano sposób komunikacji mikrokontrolera z podłączonymi urządzeniami oraz wykorzystane do tego biblioteki i fragmenty kodu w języku Arduino. W ostatnim podrozdziale o numerze 3.7 opisano zasadę działania całego algorytmu sterowania który został zrealizowany na potrzeby działania prototypu. Opisany algorytm został przetestowany na mikrokontrolerze wspierającym standard Arduino oraz uruchomiony wykorzystując wydrukowane części za pomocą technologii druku 3D. Na rysunku numer 4.2 przedstawiono całosciowo złożony prototyp z umieszczoną na górnej części osłoną.



Rysunek 4.2: Prototyp z założoną osłoną

Bibliografia

- [1] Adafruit. *Github-DHT-sensor-library.* [https://github.com/adafruit/DHT-sensor-library.](https://github.com/adafruit/DHT-sensor-library)
- [2] Arduino. *Arduino- Stepper Library.* [https://www.arduino.cc/en/reference/stepper.](https://www.arduino.cc/en/reference/stepper)
- [3] Arduino. *Arduino documentation.* [https://www.arduino.cc/reference/en/.](https://www.arduino.cc/reference/en/)
- [4] D-Robotics UK. *DHT11 Humidity and Temperature Sensor.* [http://secom.nazwa.pl/shopgold/pobieranie/dht11.pdf.](http://secom.nazwa.pl/shopgold/pobieranie/dht11.pdf)
- [5] Dassault Systèmes. *SolidWorks- Gniazdo.* [http://help.solidworks.com/2018/Polish/SolidWorks/sldworks/c_Mold_Tools_Cavity.htm.](http://help.solidworks.com/2018/Polish/SolidWorks/sldworks/c_Mold_Tools_Cavity.htm)
- [6] Dassault Systèmes. *SolidWorks- Toolbox.* [http://help.solidworks.com/2020/polish/SolidWorks/toolbox/c_toolbox_overview.htm.](http://help.solidworks.com/2020/polish/SolidWorks/toolbox/c_toolbox_overview.htm)