

Sprawozdanie
Interdyscyplinarny projekt zespołowy

Elektroniczna doniczka

Wykonali:

Patryk Mulik

Nr. albumu: 230247

Bartłomiej Polkowski

Nr. Albumu: 230241

Wojciech Sałdan

Nr. Albumu: 236277

Spis treści

1. Założenia projektowe

.....

2. Realizacja projektu

.....

3. Opis działania urządzenia

.....

4. Projekt i wykonane płytek drukowanych

.....

5. Oprogramowanie mikrokontrolera

.....

6. Montaż i uruchomienie urządzenia

.....

7. Podsumowanie

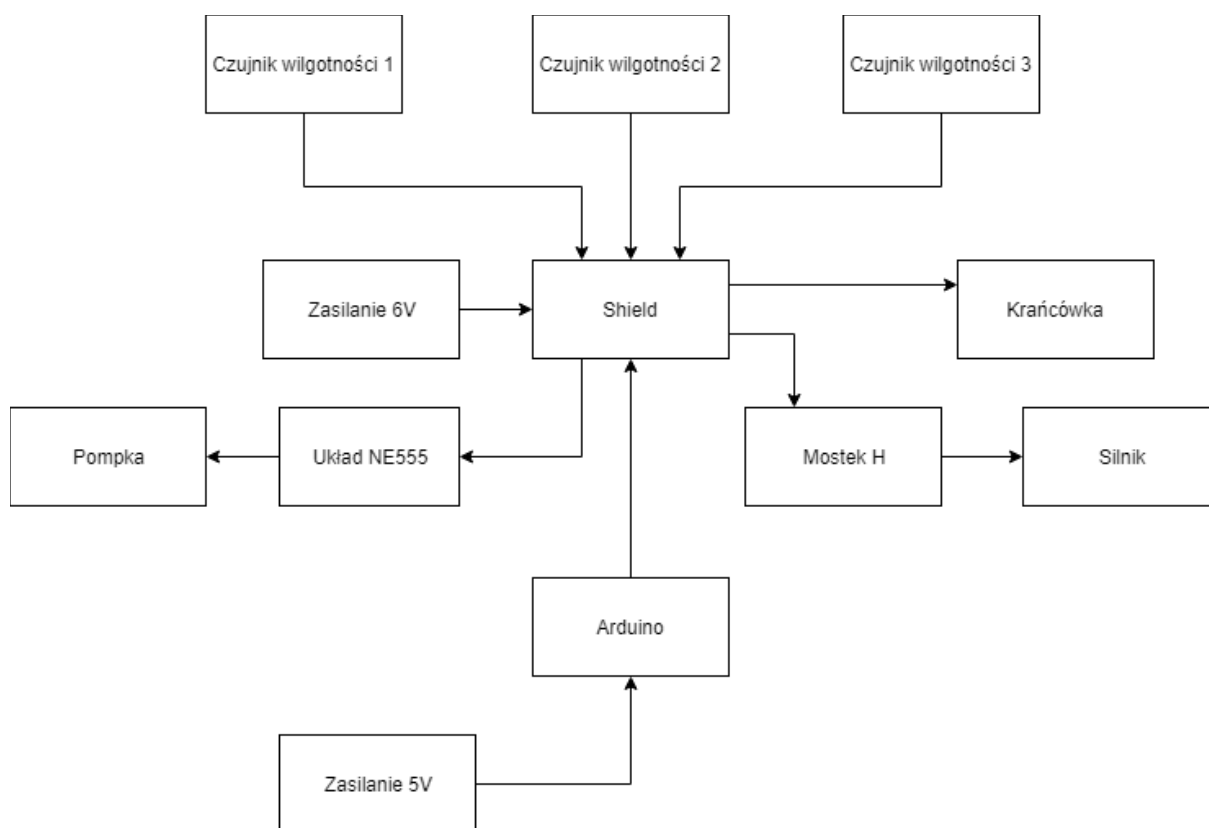
.....

1. Założenia projektowe

Założeniem projektu było wykonanie elektronicznej doniczki, której zadaniem będzie automatyczne podlewanie trzech roślin posadzonych w doniczce, w chwilach gdy będą one potrzebowały wody (na podstawie zmiany rezystancji gleby). O zmianach rezystancji gleby informować będą czujniki wilgotności gleby umieszczone w niej.

2. Realizacja projektu

Ogólny zamysł i schemat blokowy całego projektu został przedstawiony na rysunku 1, a w dalszej części zostaną opisane poszczególne elementy, ich zadania oraz zastosowanie.



Rysunek 1 . Schemat blokowy działania całego urządzenia

Cały układ składa się z wielu elementów:

- Mikrokontrolera Arduino zasilanego z zasilacza napięciem 5 V
- Silnika elektrycznego DC prądu stałego
- Pompki cieczy
- Układów elektronicznych
 - a) Mostka H – przeznaczonego do sterowania silnikiem
 - b) Układu na bazie timera NE555 – służącego do zmniejszenia napięcia zasilania pompki do 2.5 V z 6 V
- Trzech czujników wilgotności gleby
- Wyłącznika krańcowego magnetycznego (nazwanego na schemacie „Krańcówką”)
- Shielda nakładanego na płytkę Arduino w celu zmniejszenia ilości prowadzonych przewodów
- Drewnianej konstrukcji mechanicznej

3. Opis działania urządzenia

Urządzenie, które zaprojektowaliśmy służy do automatycznego podlewania 3-ech roślin posadzonych w jednej doniczce. Za informację o tym, czy gleba ma odpowiednią zawartość wody odpowiedzialne są czujniki wilgotności gleby. W momencie, gdy czujniki poinformują system oprogramowania układu o zmianie rezystancji gleby, rozpoczyna się proces podlewania danej rośliny. Uruchomiony zostaje mostek H, służący do sterowania silnikiem elektrycznym. Do silnika elektrycznego zamontowany został łożyskowany pręt gwintowany z nakrętką, do której zamontowany został wężyk pompki. Na podstawie nawigacji zliczeniowej, silnik kręci odpowiednio prętem, tak aby nakrętka wykonująca ruch posuwisty przemieściła się na pręcie na odpowiednie stanowisko danej rośliny. Gdy nakrętka wraz z zamocowanym wężykiem („układ podlewaczki”) znajduje się w odpowiedniej pozycji, uruchomiony zostaje układ elektroniczny oparty na timerze NE555, służący do sterowania pompką który zmniejsza napięcie zasilania z 6 V do 2.5 V by strumień wody nie był za mocny i nie wylewał wody poza obrzys doniczki. Pompka zostaje zasilana, dzięki czemu rozpoczyna się procedura podlewania danej rośliny. Po wykonaniu zadania silnik kręcąc prętem w przeciwnym kierunku, powoduje powrót nakrętki z wężykiem do pozycji zerowej. O tym, że „podlewaczka” znajduje się w pozycji początkowej informuje wyłącznik krańcowy. Sygnał odbierany jest przez wyprowadzenie cyfrowe pod które jest podciągnięty pull up. Gdy magnes neodymowy znajdujący się na nakrętce dotrze w pobliże odbiornika czujnik magnetyczny zwiera obwód co pozwala odczytać logiczne 0 na wejściu innego cyfrowego wyprowadzenia. Po powrocie do tej pozycji, urządzenie oczekuje odpowiedni czas na ustabilizowanie ponownego pomiaru następnie pozostaje w stanie gotowości, w razie zmiany rezystancji gleby na danym stanowisku nasza elektroniczna doniczka powtarza procedurę opisaną wcześniej.

Opis poszczególnych elementów:

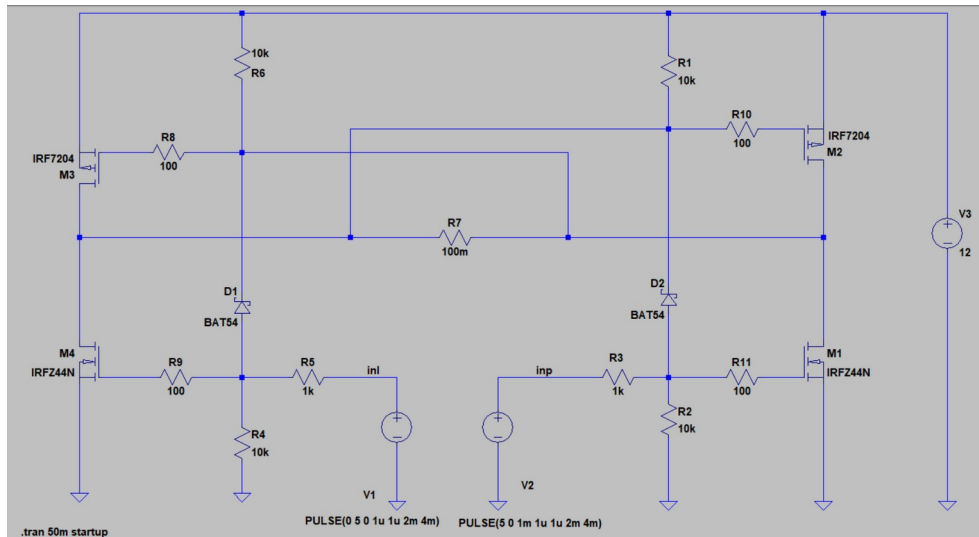
a) Silnik elektryczny

Do ruchu obrotowego pręta gwintowanego użyty został silnik elektryczny DC zasilanego od 3.3 V do 12 V. Silnik ten sterowany jest za pośrednictwem mostka H. Jako sprzęgło między prętem a silnikiem została użyta kawałek teflonowego wężyka dzięki czemu nasze sprzęgło jest bardziej elastyczne i pozwala na dłuższy czas działania. Dzięki obracaniu się pręta, ruch posuwisty wykonuje „układ podlewaczki” tzn. nakrętka z zamocowanym wężykiem pompki. Układ ten został odpowiednio wyważony, aby zmniejszyć bicie pręta (zlikwidować fakt, bycia pręta w rezonansie).

b) Mostek H

Mostek H został zastosowany w celu ułatwienia sterowania silnikiem ponieważ zasilanie silnika jest większe niż wychodzące z mikrokontrolera oraz umożliwia zmianę kierunku obrotu

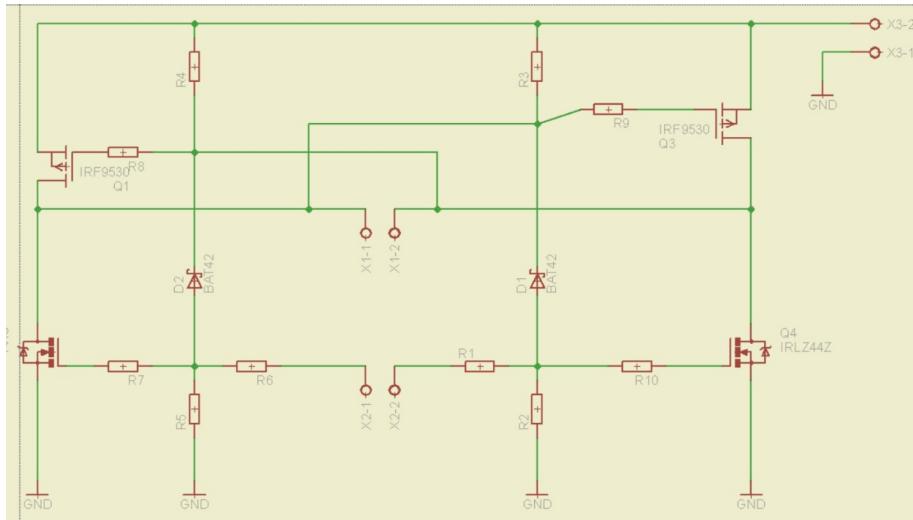
Pierwszym krokiem po ustaleniu potrzeby zastosowania mostka H do sterowania silnikiem, było przeprowadzenie symulacji w programie LTspice w celu sprawdzenia poprawności doboru elementów oraz działania układu.



Rysunek 2 Schemat mostka H w programie LTspice

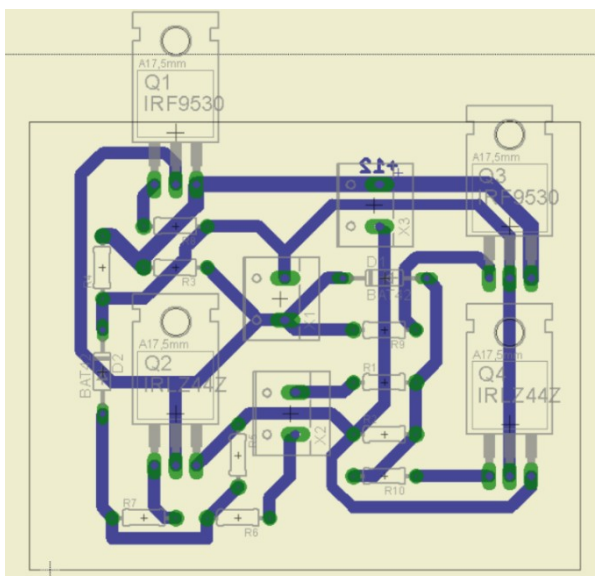
Po przeprowadzeniu symulacji możemy zaobserwować że układ dobrze zabezpiecza przed zwarcie sterowany silnik co było jego głównym zadaniem.

Gdy została sprawdzona poprawność działania układu został on ponownie zaprojektowany lecz tym razem w programie EAGLE 5.12.0 zwracając uwagę na rodzaje obudów elementów.

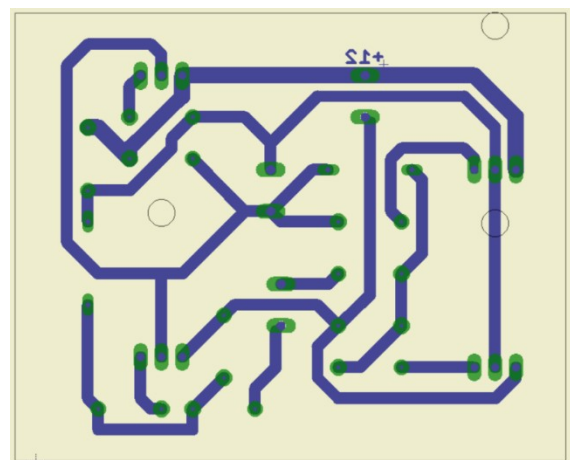


Rysunek 3 Schemat ideowy mostka H w programie EAGLE 5.12.0

Mając gotowy schemat ideowy układu przystąpiono do wykonania schematu płytki pcb. W tym przypadku uwaga zwrócona została nie tylko na obudowy elementów ale również na rozmiar pól lutowniczych czy grubość ścieżek. Znacznie pogrubiona w stosunku do innych została ścieżka zasilania ze względu na przepływający nią większy prąd.



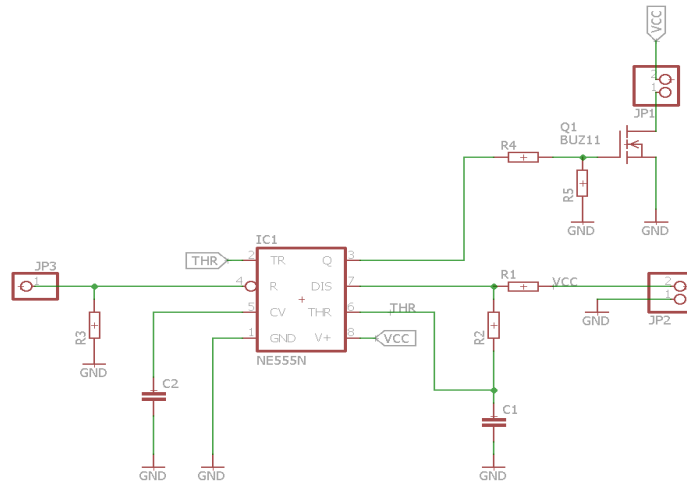
Rysunek 5 Schemat rozmieszczenia elementów płytki drukowanej



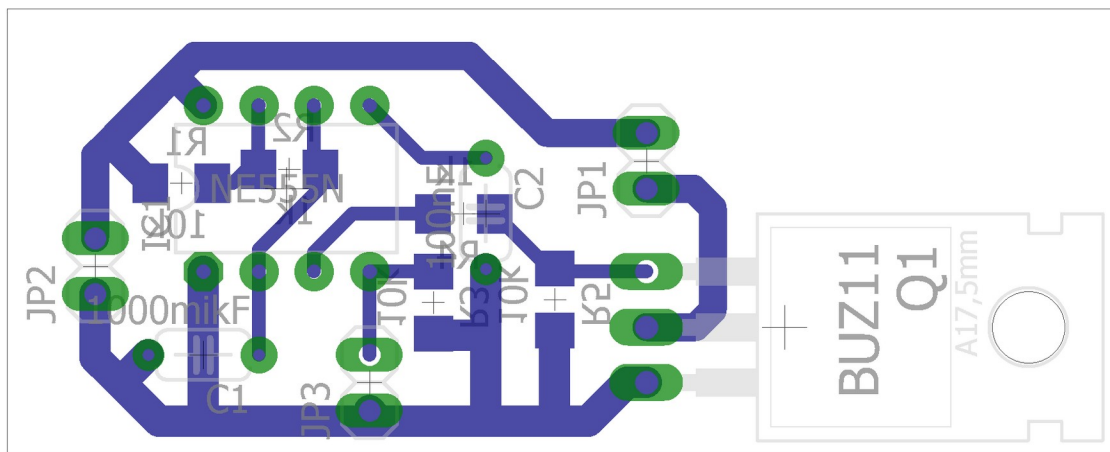
Rysunek 4 Schemat rozmieszczenia ścieżek płytki drukowanej

c) Układ pompki (na bazie układu timera NE555)

Układ ten służy do obniżania napięcia z 6 V do 2.5 V. Został zastosowany ponieważ pompka przy zasilaniu 6 V podlewa nasze stanowiska ze zbyt dużym ciśnieniem i woda nie jest dozowana precyzyjnie na stanowiska w których umieszczona jest roślina. Po serii prób z różnymi napięciami zasilania pompki zdecydowaliśmy się zasilać ją napięciem 2,5 V dzięki czemu woda dozowana jest precyzyjnie.



Rysunek 6 Schemat układu z NE555



Rysunek 7 Schemat płytki z układem scalonym NE555

d) Czujniki wilgotności gleby

Urządzenie składa się z trzech części: sondy pomiarowej, modułu detektora oraz przewodów. Sondy należy połączyć z modułem głównym przy pomocy przewodów i umieścić w glebie, której wilgotność będzie mierzona. Czujnik posiada wyjście cyfrowe D0 sygnalizujące przekroczenie ustawionej za

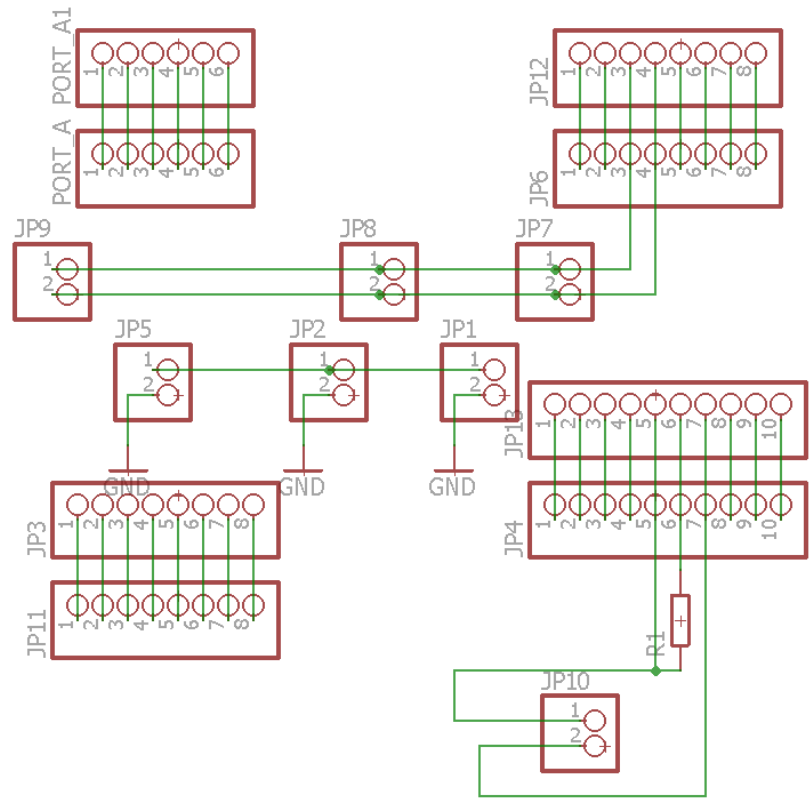
pomocą potencjometru wartości oraz analogowe A0 przy pomocy którego uzyskuje się dokładną wartość wilgotności.

- e) Wyłącznik krańcowy (Czujnik magnetyczny otwarcia drzwi/okien - kontaktron CMD19)

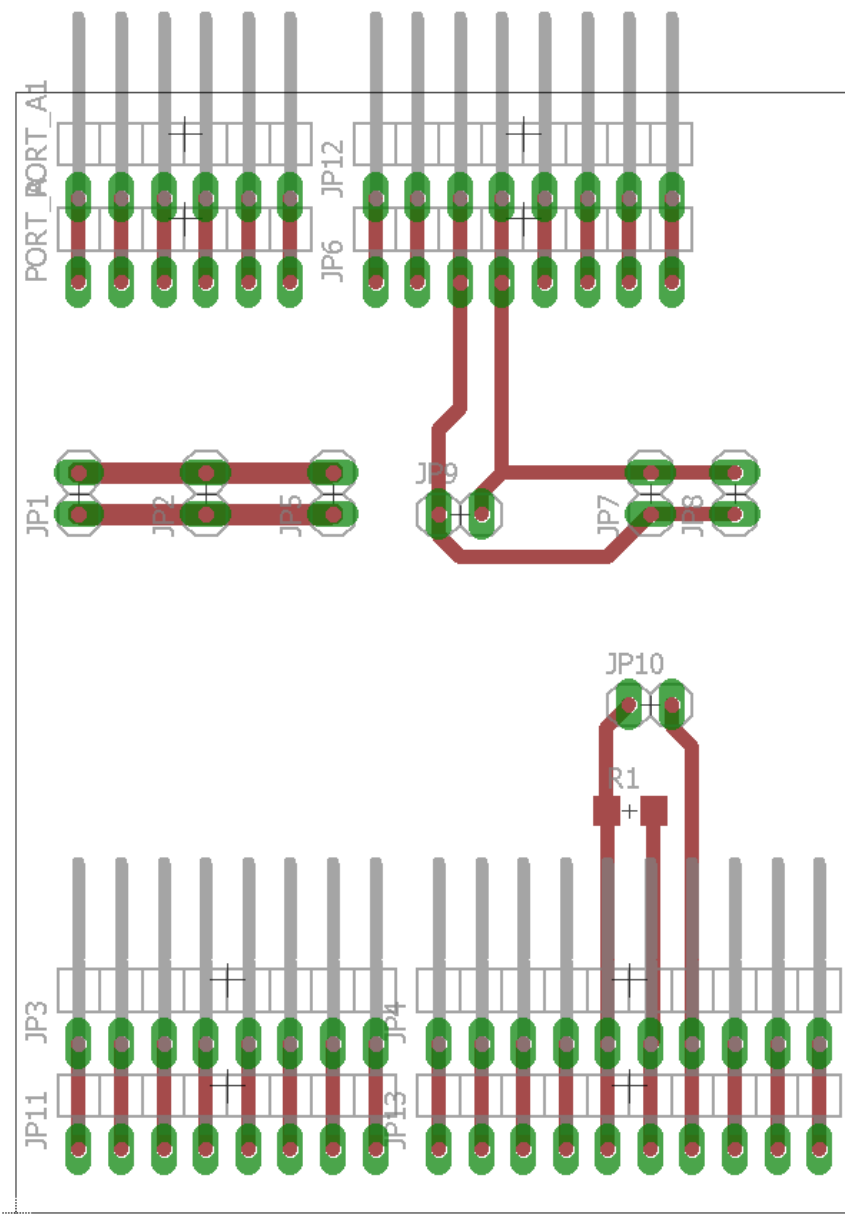
Przewodowy czujnik zbliżeniowy załączany magnetycznie. Urządzenie stosowane jest głównie do określenia pozycji drzwi i okien. Zasada działania jest podobna jak w tradycyjnym przekaźniku. Obwód jest domyślnie rozarty, po zbliżeniu do magnesu obwód zostaje zamknięty, co oznacza, że przez przewody zaczyna płynąć prąd.

- f) Shield

Został zastosowany w naszym układzie by ograniczyć ilość przewodów oraz ewentualnego rozłączania się przewodów podczas przemieszczania urządzenia. Shield nakładany na płytkę Arduino pozwala na zachowanie porządku oraz szybką diagnostykę w razie ewentualnych awarii układu. Na płytce znajdują się trzy zespoły wyprowadzeń. Pierwszy odpowiada za obsługę wejść i wyjść cyfrowych. Jest to odczyt stanów czujników, sterowanie mostkiem H, sterowanie układem z NE555, obsługa czujnika krańcowego. Drugim zespołem jest zasilanie zewnętrzne na 6 V, które zasila pompkę oraz silnik. Trzecim zespołem jest zasilanie niskoprądowe z mikrokontrolera Arduino do zasilania czujników i układu NE555. Układ płytki jest tak zaprojektowany by rozstaw goldpinów odpowiadał rozstawowi rastrowemu to znaczy 2,54 mm między wyprowadzeniami.



Rysunek 8 Schemat płytki Shield



Rysunek 9 Układ poprowadzenia ścieżek na płycie Shield

4. Projekt i wykonanie płytek drukowanych

Po zakończeniu projektu płytki w programie EAGLE 5.12.0 oraz dodania jej obrysu który zapewnia bezpieczną odległość elementów od krawędzi płytki. Następnie przystąpiono do wykonania wydruku schematu na papierze kredowym oraz wycięcia w laminacie kształtu odpowiadającego wymiarom o jeden do dwóch milimetrów obrysów, przy pomocy gilotyny. W przyciętym laminacie oszlifowano krawędzie papierem ściernym. Mając laminat o odpowiednich wymiarach został

on oczyszczony przy pomocy papieru ściernego. Od momentu oczyszczenia laminatu do naniesienia toneru, ostrożnie obchodzono się z laminatem, żeby nie pozostawić na nim śladów mogących powodować pogorszenie przylegania tonera. W celu nanoszenia toneru, ułożono wydruk schematem do góry i przyłożono do niego laminat od strony oczyszczonej miedzi. Następnie dopasowane do siebie elementy ułożono na płycie grzewczej podgrzewacza i dociskano przez około 30 sekund. Po tym procesie odklejamy papier od laminatu. Zauważono braki w naniesieniu tonera w hipotetycznym miejscu mocowania do laminatu (zrezygnowano z tego rozwiązania ze względu na brak konieczności większego odprowadzania ciepła).

Do korekty braków użyto niezmywalnego mazaka. Po dokonaniu korekt przystąpiono do wytrawiania. Płytkę została umieszczona w kuwecie wypełnionej roztworem nadsiarczanem sodu, w kuwecie znajdował się również podgrzewacz wody. Oczekiwanie zajęło ponad 1,5 godziny. Tak długi czas oczekiwania mógł być spowodowany dużym stopniem zużycia wytrawiacza. Wytrawiona płytka została umyta bieżącą wodą i wyczyszczono drucianym czyścikiem. Na wyczyszczonej płytce precyzyjną wiertarką wykonano otwory. W miejscach lutowań elementów naniesiono małą warstwę topnika (kalafonia rozpuszczona w spirytusie) o kleistej konsystencji, który zmniejszy utlenienie miedzi i ułatwi lutowanie. Gdy topnik był naniesiony umieszczono płytkę w uchwycie do lutowania i zamontowane metodą przewlekana z użyciem cyny bezołowiowej. Jako ostatni etap płytka została dokładnie umyta acetonem z pozostałości topnika aby wyeliminować możliwość zwarcia.

5. Oprogramowanie mikrokontrolera (fragmenty kodu)

Program został napisany w języku programowania C w środowisku Arduino IDE. W pierwszej kolejności zostały zadeklarowane wyjścia w taki sposób by wyprowadzenia były nazwami funkcji przez nich wykonywanych. Przykładem tego jest zadeklarowanie wyprowadzenia cyfrowego numer 3 jako czujnik1. Pozwala to na łatwiejsze pisanie kodu oraz ewentualną diagnozę i korektę w przypadku błędów. Następną częścią kodu jest zadeklarowanie które wyprowadzenia są wejściami a które wyjściami. Kod składa się z nie obiektowych instrukcji głównie „switch case” oraz „if else”. Oprogramowanie jest tak skonstruowane by mogło obsługiwać urządzenie w dowolnym położeniu początkowym naszej nakrętki. W pierwszej kolejności działania programu jest sprawdzana pozycja nakrętki oraz obsługa jej powrotu do miejsca zerowego. Dzięki temu nasza nawigacja zliczeniowa będzie działać niezawodnie nawet w przypadku chwilowego braku zasilania. Nasz układ pomiarowy może znaleźć się w jednym z 8 stanów poprzez stany logiczne na poszczególnych czujnikach opisanych poniżej w tabelce:

Tabela 1 Tabela możliwości odczytów czujników

| l.p | Czujnik 1 | Czujnik 2 | Czujnik 3 |
|-----|-----------|-----------|-----------|
| | | | |

| | | | |
|---|---|---|---|
| 1 | 0 | 0 | 0 |
| 2 | 1 | 0 | 0 |
| 3 | 1 | 1 | 0 |
| 4 | 1 | 1 | 1 |
| 5 | 0 | 0 | 1 |
| 6 | 0 | 1 | 1 |
| 7 | 1 | 0 | 1 |
| 8 | 0 | 1 | 0 |

Aby móc zastosować instrukcję „switch case” należało opracować odpowiedni algorytm obsługi. Suma stanów logicznych na poszczególnych wyprowadzeniach dała by nierzeczywisty obraz który czujnik wymaga obsługi. Np. suma równa 1 może oznaczać że czujnik pierwszy lub drugi lub trzeci wymaga obsługi. Z tego powodu został opracowany algorytm oparty na wzorze matematycznym poniżej gdzie:

Czujnik 1 = x

Czujnik 2 = y

Czujnik 3 = z

$$Suma = x + 2y + 4z$$

Dzięki temu każdy z odczytów da inną sumę i można obsłużyć poprawnie stanowiska. Poniżej w tabeli zostały umieszczone możliwe rozwiązania.

Tabela 2 Możliwości otrzymanych sum po zastosowaniu algorytmu

| Suma | Czujnik 1 | Czujnik 2 | Czujnik 3 |
|------|-----------|-----------|-----------|
| | x | y | z |
| 0 | 0 | 0 | 0 |
| 1 | 1 | 0 | 0 |
| 2 | 0 | 1 | 0 |
| 3 | 1 | 1 | 0 |
| 4 | 0 | 0 | 1 |
| 5 | 1 | 0 | 1 |
| 6 | 0 | 1 | 1 |
| 7 | 1 | 1 | 1 |

W ten sposób algorytm rozpoczyna obsługę odpowiedniego stanowiska. Poniżej został pokazany fragment kodu z obsługą stanowiska pierwszego. W takim przypadku suma równa jest jeden warunek jest obsługiwany dla sumy równej jeden. W tym przypadku nakrętka jest przemieszczana na stanowisko pierwsze kręcąc silnikiem w lewą stronę przez 5 sekund. Po sekundowej przerwie następuje podlewanie przez pompkę która jest włączona przez 3 sekundy. Po podlaniu nakrętka wraca na swoje stanowisko i czeka odpowiednią ilość czasu by sprawdzić czy rezultat podlania jest dał wynik który zmienił odczyt. Jeśli nie to ponownie rozpoczyna obsługę jeśli tak to czeka aż pojawi się kolejny odczyt który będzie wymagał obsługi kolejnego stanowiska.

```
int czujnik1 = 3; // czujnik 1 na wyprowadzeniu cyfrowym nr 3
```

```
int czujnik2 = 4; // czujnik 2 na wyprowadzeniu cyfrowym nr 4
```

```

int czujnik3 = 5; // czujnik 3 na wyprowadzeniu cyfrowym nr 5
int silnikl = 6; // sterowanie silnikiem włączenie lewych obrotów na wyjściu cyfrowym 6
int silnikp = 7; // sterowanie silnikiem włączenie prawych obrotów na wyjściu cyfrowym 7
int pompka = 8; //włączanie pompki za pomocą wyprowadzenia 8
int krancowkai = 13; //odczyt stanu krańcówki z 12 wejścia cyfrowego
int krancowkao = 12; //wyjście cyfrowe nr 13 podciągnięty pull up do krańcówki
void setup() {
  pinMode(czujnik1, INPUT); //ustalenie wyprowadzenia 3 jako wejście
  pinMode(czujnik2, INPUT); //ustalenie wyprowadzenia 4 jako wejście
  pinMode(czujnik3, INPUT); //ustalenie wyprowadzenia 5 jako wejście
  pinMode(krancowkai, INPUT); //ustalenie wyprowadzenia 13 jako wejście
  pinMode(silnikl, OUTPUT); //ustalenie wyprowadzenia 6 jako wyjście
  pinMode(silnikp, OUTPUT); //ustalenie wyprowadzenia 7 jako wyjście
  pinMode(krancowkao, OUTPUT); //ustalenie wyprowadzenia 12 jako wyjście
  pinMode(pompka, OUTPUT); //ustalenie wyprowadzenia 6 jako wyjście
  Serial.begin(9600); //włączenia monitora portu szeregowego by monitorować stan
  wyprowadzeń
}
void loop() {

  int sum = digitalRead(czujnik1) + digitalRead(czujnik2)*2 + digitalRead(czujnik3)*4; // wzór na
  sumę odczytów z czujników wilgotności

  digitalWrite(krancowkao, HIGH); // włączenie pull up
  int val = digitalRead(krancowkao); // włączenie sprawdzania stanu krańcówki

  Serial.println(val); // odczyt na monitorze portu szeregowego sumy odczytów czujników

  if(val==1) // sprawdzenie położenia nakrętki
  {

```

```
digitalWrite(silnikp, HIGH); // jeśli nakrętka nie znajduje się w pozycji zerowej kręć silnikiem w prawo aż się tam znajdzie
```

```
}
```

```
Else // w przeciwnym wypadku zacznij obsługę programu
```

```
{
```

```
digitalWrite(silnikp, LOW);
```

```
switch(sum){ // wybór spośród opcji obsługi poszczególnych czujników
```

```
case 0: // jeśli suma odczytów jest równa 0 to poczekaj 5 sekund i ponów odczyt
```

```
delay(5000);
```

```
break;
```

```
case 1: // jeśli suma jest równa 1 rozpocznij obsługę czujnika pierwszego
```

```
Serial.println("czujnik 1"); // poinformuj na monitorze portu szeregowego o tym którego czujnika obsługę zaczynasz
```

```
Serial.println(sum); //wyświetl sumę stanów czujnika na monitorze portu szeregowego
```

```
delay(1000);
```

```
digitalWrite(silnikl, HIGH); // włącz silnik w lewo tak by dojechać do pierwszego stanowiska w tym przypadku 5 sekund
```

```
delay(5000);
```

```
digitalWrite(silnikl, LOW);
```

```
delay(1000);
```

```
digitalWrite(pompka, HIGH); włącz nalewanie na 3 sekundy
```

```
delay(3000);
```

```
digitalWrite(pompka, LOW);
```

```
delay(1000);
```

```
if(val==1)// sprawdź położenie nakrętki
```

```
{
```

```
digitalWrite(silnikp, HIGH); // jeśli nie ma jej na miejscu kręć w prawo aż się tam znajdzie
```

```
}
```

```
else
```

```
{
```

```
digitalWrite(silnikp, LOW);
```

```
}
```

```
delay(10000); // odczekaj do ponownego pomiaru 10 sekund
```

```
break;
```

6. Montaż i uruchomienie urządzenia

Całe urządzenie zostało zamontowane na drewnianej sklejce z nóżkami, która służy jako podstawka (stolik). Na tym stoliku została umieszczona doniczka wypełniona ziemią, następnie zaprojektowaliśmy „szkielet” wykonany z cienkich drewnianych listewek połączonych w kształt odwróconej litery „U” za pomocą kleju na gorąco, i również tym klejem został przyklejony do podstawki. „Szkielet” ten został zrobiony w celu zamontowania na nim silnika elektrycznego oraz łożyskowanego pręta gwintowanego wraz z nakrętką oraz wyłącznika krańcowego. Do nakrętki na stałe zamocowany został wężyk pompki, służący do podlewania roślin oraz magnes współpracujący z wyłącznikiem krańcowym. Obok szkieletu zostały umiejscowione zbiornik na wodę pod postacią kolejnej, mniejszej doniczki, w której została zamocowana na stałe pompka, również przy użyciu kleju na gorąco. Wszystkie układy elektroniczne – mostek H, układ na bazie timera NE555, shield nakładany oraz mikrokontroler Arduino, zostały umieszczone na stoliku w bezpiecznej odległości aby zapobiec zalaniu całego układu wodą.

7. Podsumowanie

Spis elementów:

IRF7204 x2

IRFL44N X2

BAT54 X2

10KOM X4

1KOM X2