

1. Ogólny zarys projektu:

Projekt „*Mikrokontrolerowy system kontroli i regulacji nawodnienia gleby*” został opracowany w celu zautomatyzowania tak prozaicznej czynności jaką może wydawać się podlewanie roślin doniczkowych. Dzięki urządzeniu miłośnicy zieleni nie będą musieli pamiętać o regularnym podlewaniu, ponieważ system robi to samoczynnie wykorzystując do tego czujnik monitorujący wilgotność ziemi w doniczce w czasie rzeczywistym.

2. Przeznaczenie i montaż:

System jest odpowiedni dla wszystkich roślin, które „lubią” wodę i wymagają regularnego podlewania. Należą do nich między innymi: *Euphorbia pulcherrima* (poinsecja nadobna), *Cupressus* (cyprys), *Cyclamen* (cyklamen), *Calathea* (kalatea).

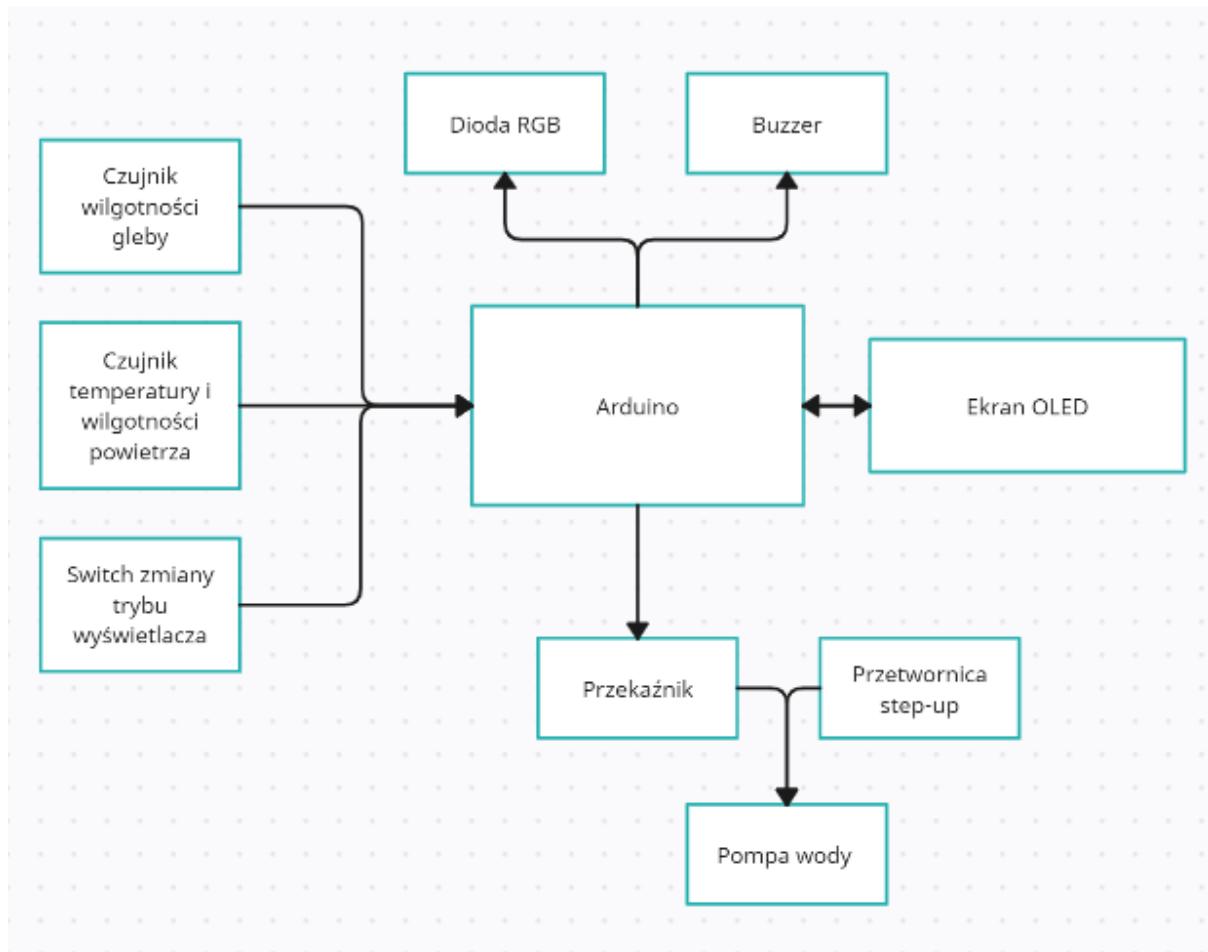
Dużym atutem jest możliwość zastosowania urządzenia przy dowolnej wielkości doniczki. W celu dostosowania wystarczy jedynie skrócić lub wydłużyć wężyk wychodzący z pompy wody.

Urządzenie jest przeznaczone do działania wewnątrz budynków ze względu na brak wodoodporności. Montaż urządzenia polega na umieszczeniu wężyka pompy wody wokół ścian doniczki w formie spirali. Takie ułożenie gwarantuje równomierne nawodnienie w każdym miejscu w doniczce. Pompę wody należy umieścić w pojemniku z wodą (posiada ona również otwory montażowe, ale nie jest konieczne przytwierdzenie jej do podłoża). Następnie do doniczki należy włożyć roślinę, a czujnik wilgotności wbić w ziemię do poziomu białej linii na nim widocznej i wcisnąć przycisk zasilania. System zasilany jest baterią 9V typu 6F22.

3. Schemat blokowy i elektryczny układu:

Najczęściej wyszukując systemy automatycznego podlewania można spotkać małe pojemniki z wodą wbijane w ziemię, które opróżniając się podlewają roślinę. Efektem tego jest ciągle utrzymywanie wysokiej wilgotności gleby co może doprowadzić do gnicia korzeni i zniszczenia rośliny. Należy więc wprowadzić pomiar wilgotności gleby, jednakże elektroniczne systemy nawadniania roślin nie są zbyt popularne. Najczęściej można spotkać urządzenia do samodzielnego montażu, oparte o czasomierze, które podlewają rośliny w

równych odstępach czasowych. Mój system wychodzi temu naprzeciw, nawadniając glebę dopiero wtedy gdy jest to potrzebne.



Rys.1; Blokowy schemat ideowy układu.

Dobór komponentów:

- **Arduino UNO** - “serce” systemu zarządzające innymi komponentami i przetwarzające dane. Arduino jest idealną platformą pod tego typu projekty ze względu na popularność, dostępność bibliotek do obsługi elektroniki oraz łatwość obsługi. Dodatkową zaletą jest fakt uniwersalności; w przyszłości projekt można przenieść do dedykowanej obudowy i jako platformę zastosować np. Arduino NANO lub pójść o krok dalej i użyć samego mikrokontrolera np. ATmega328P. W projekcie wykorzystany został model Arduino UNO, ponieważ byłem w jego posiadaniu już wcześniej.

W projekcie wykorzystujemy 7 wejść/wyjść cyfrowych, jedno wejście analogowe oraz piny SDA i SCL służące do obsługi ekranu OLED. Powyższe wymagania spełnia każdy model Arduino, więc przy wyborze należy zwrócić uwagę głównie na cenę i dostępność. Istotnym kryterium jest też rozmiar płytki, ponieważ wpłynie na on na ostateczne gabaryty urządzenia. Pod tym względem najlepsze będzie zastosowanie Arduino NANO, która jest najmniejsza i najtańsza z całej "rodziny".

Cena Arduino UNO - około 120 złotych oryginał, około 40 zł "klon". Klony to płytki posiadające te same parametry oraz możliwości co oryginalne Arduino, jednak produkowane przez inne firmy. Nie posiadają logo Arduino. Jest to możliwe ze względu na brak posiadania patentu przez firmę Arduino Software

- **wyświetlacz OLED SH1106 128x64px** - wyświetlacz wybrany ze względu na obsługiwane protokołu I²C oraz niską cenę. Posiada przekątną 1,3 cala i wyświetla znaki w kolorze białym. Dodatkowo atutem ekranu OLED jest oszczędność energii ze względu na podświetlanie tylko wyświetlanych pikseli.

Cena wyświetlacza OLED ze sterownikiem SH1106 - około 40 złotych. Czyni go to najtańszym ekranem OLED o tej przekątnej ekranu korzystającym z magistrali I²C.

- **czujnik wilgotności gleby M335** - wybór uzależniony od dostępności i ceny. Jest to najtańszy pojemnościowy czujnik wilgotności gleby jaki udało się znaleźć, w zupełności wystarczający przy braku konieczności bardzo dokładnych pomiarów. Zdecydowałem się na zrezygnowanie z rezystancyjnych czujników wilgotności gleby. ze względu na duże przekłamanie pomiarów w przypadku nawożonej gleby. Czujnik rezystancyjny wykrywa również sole mineralne zawarte w nawozach. Czujnik M335 wyróżnia wykonanie z materiałów odpornych na korozję, co przy jego przeznaczeniu znacznie wydłuży żywotność. Pracuje na popularnym module NE555.

Cena czujnika M335 - około 7 złotych.

- **czujnik temperatury i wilgotności powietrza DHT11** - czujnik wybrany ze względu na ilość bibliotek do jego obsługi, co przekłada się na łatwość w odczytaniu i przetwarzaniu danych. Oferuje dość dobrą dokładność pomiarów za niewielką cenę. Na rynku istnieje wiele czujników oferujących większą dokładność czy też kompaktować, ale ich cena jest kilkakrotnie wyższa od ceny czujnika DHT11. W projekcie nie wymagam dużej dokładności pomiarów temperatury i wilgotności powietrza. Dane z tego czujnika mają charakter informacyjny dla użytkownika odnośnie warunków panujących w pomieszczeniu..

Cena czujnika DHT11 - około 15 złotych.

- **moduł przekaźnika SRD-5VDC-SL-C** - wybór ze względu na cewkę przekaźnika zasilaną napięciem 5V DC, czyli takim jakie jesteśmy w stanie osiągnąć na wyjściu

cyfrowym Arduino. Dodatkowym atutem jest niska cena i obszerna dokumentacja techniczna.

Cena modułu przekaźnika - około 5 złotych.

- **przetwornica step-up MT3608** - zastosowana do zasilania pompy wody. Przetwornica zasilana napięciem stałym od 3 do 24V, na wyjściu dająca od 4 do 25V.
- Napięciem wyjściowym można sterować przy użyciu potencjometru znajdującego się na przetwornicy. Wydajność prądowa 2A, co w zupełności wystarcza do zasilania pompy. Posiada wejście microUSB, dzięki czemu można zasilić ją z podstawowej ładowarki do telefonu. Minusem jest potrzeba samodzielnego wlutowania goldpinów (których brak w zestawie) do wyprowadzeń wyjścia zasilania.

Cena przetwornicy MT3608 - około 5 zł.

- **pompa wody DollaTek AD20P-1230C** - pompa zasilana napięciem 12VDC, pobiera prąd około 400mA podczas pracy. Jest to pompa bezszczotkowa, dzięki czemu jest oszczędna, lżejsza i mniej awaryjna od modeli posiadających szczotki. Wyprowadzenia pompy pasują pod wąż o średnicy 10 mm, co bardzo ułatwia proces montażu i eliminuje potrzebę stosowania opasek zaciskowych. Pompa może osiągnąć przepływ na poziomie 240L/h, wystarczający do podlewania roślin doniczkowych.

Cena pompy wody - około 40 złotych.

- **Buzzer z generatorem 5V** - wykorzystywany do sygnalizacji dźwiękowej, posiada wbudowany generator co oznacza, że aby wydał z siebie dźwięk wystarczy podać jego zaciski zasilanie. Wybór konkretnego modelu nie ma tutaj znaczenia, należy jedynie zwrócić uwagę na posiadanie wbudowanego generatora.

Cena buzzera - około 2 złote.

- **Diody RGB** - dioda ze wspólną katodą, emitująca 3 podstawowe kolory: czerwony, zielony i niebieski. Poniżej podstawowe parametry diody RGB:

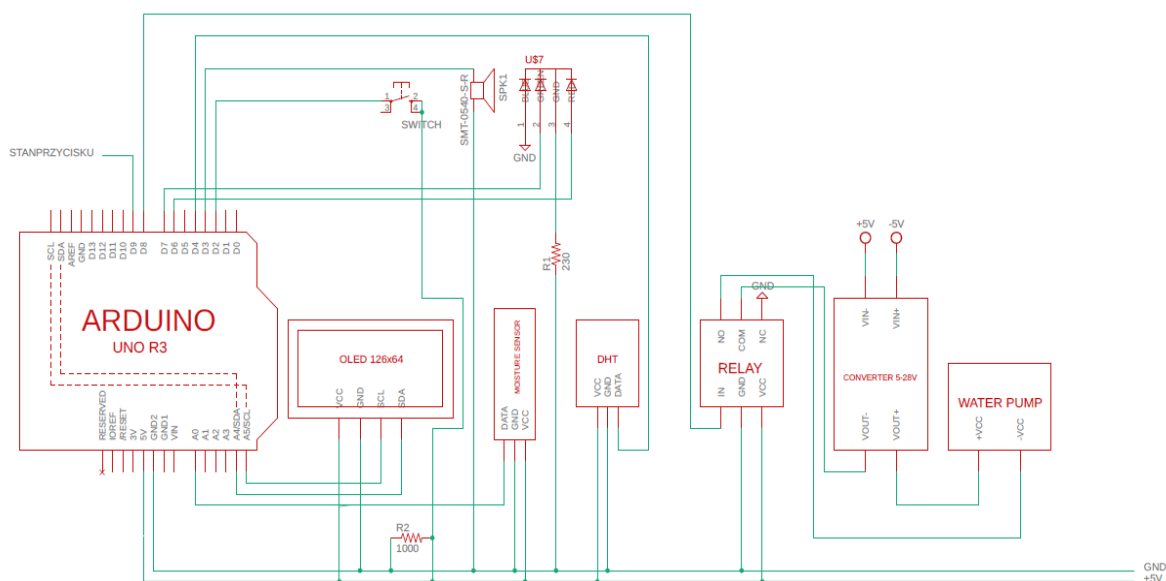
- Typ diody: THT
- Typ soczewki diody: przezroczysta
- Napięcie zasilania: R: 2 V / G: 2,5 V / B: 3,3 V
- Kolor diody: czerwony / zielony / niebieski
- Prąd maksymalny diody: 20 mA
- Strumień świetlny diody: od 600 do 799 mcd
- Kąt świecenia diody: 30°

Najważniejszymi parametrami dla założeń projektu jest strumień świetlny diody oraz jej kąt świecenia. Strumień świetlny odpowiada za ilość światła emitowanego przez

Cena diody RGB - około 1 złoty.

- Cena switcha - około 50 groszy.

- Cena rezystorów - około 50 groszy za sztukę.



Rys.2: Schemat elektryczny układu.

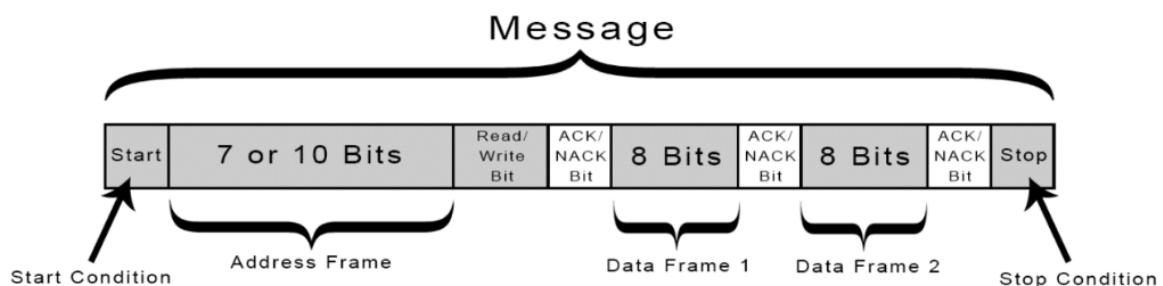
4. Opis działania systemu:

Projekt oparty jest na platformie Arduino, która wyposażona jest w mikrokontroler Atmega328P. Atmega328P to mikrokontroler AVR produkowany przez firmę Atmel. Najważniejsze parametry oraz wygląd samego mikrokontrolera (w obudowie DIP) poniżej.

- Zasilanie: 1,8 V - 5,5 V
- Taktowanie: do 20 MHz
- Pamięć Flash: 32 KB
- 23 linie wyjścia/wejścia
- Dwa 8-bitowe liczniki
- Jeden 16-bitowy licznik
- 6 kanałów PWM
- 6 kanałów 10-bitowego przetwornika analogowo-cyfrowego
- Sprzętowe interfejsy komunikacyjne: USART, SPI, TWI (I2C)
- Obudowa DIP

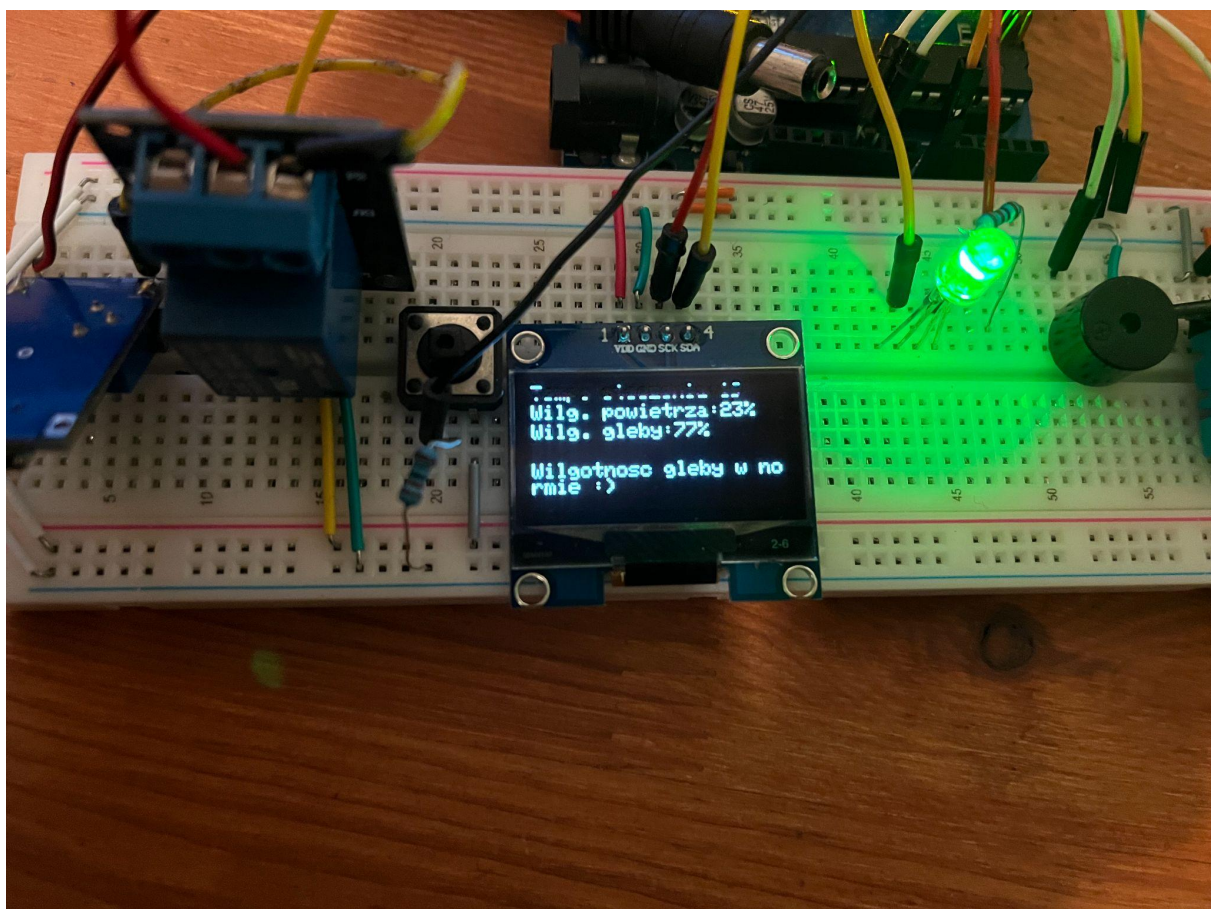


Większość danych na jakich pracuje system jest wyświetlanych na ekranie OLED, który jest podłączony do Arduino za pomocą magistrali I²C. I²C to szeregową, dwukierunkową magistralę danych służącą do przesyłania danych. Do obsługi magistrali I²C służą pin-y SDA (przesył danych) oraz SCL (zegar). Komunikacja korzysta z modelu master-slave, czyli Arduino w tym przypadku to master - zarządzający komunikacją, mogący wysyłać i odbierać dane od urządzeń podłączonych do magistrali I²C (ekran OLED).

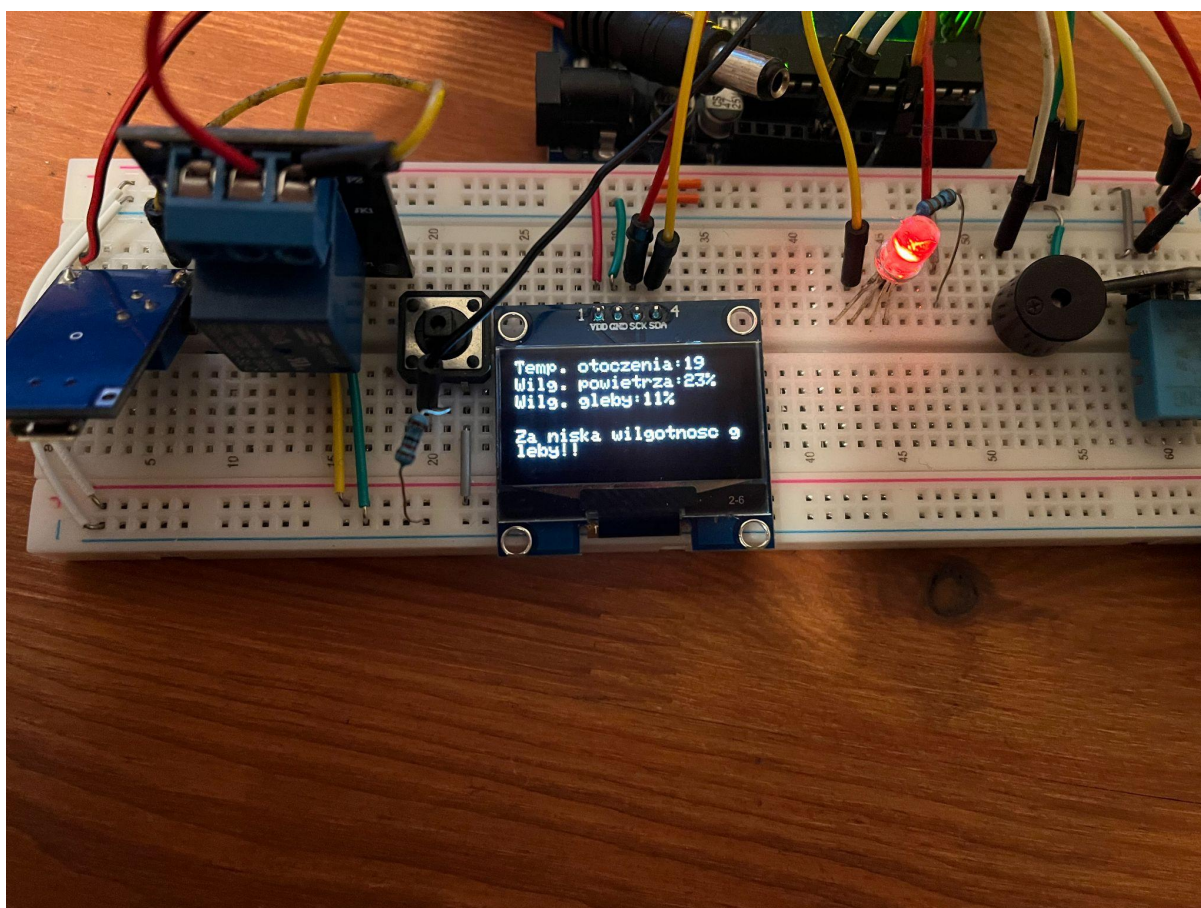


Powyżej przedstawiona jest model przesyłania danych magistralą I²C. Warunkiem rozpoczęcia przesyłu danych jest zmiana stanu na linii SDA na niski. To samo tyczy się linii SCL, jednak ważna jest tutaj kolejność (najpierw musi zmienić się stan na linii SDA). Następnie master (w tym przypadku Arduino) wysyła adres komponentu z którym chce

“rozmawiać” do wszystkich komponentów typu slave. Jeśli adres który wysłał master pokrywa się z tym który jest przypisany do jednostki slave, wysyła ona bit ACK (potwierdzenia) do mastera aby potwierdzić otrzymanie wiadomości. Urządzenia, których adres nie pokrywa się z tym wysłanym przez mastera ignorują wiadomość. W adresie komponentu zawarty dodatkowo jest bit, który informuje o tym, czy master chce wysyłać lub odbierać dane. Po odebraniu bitu ACK zaczyna się przesył danych. Każda ramka danych składa się z ośmiu bitów po których następuje bit ACK potwierdzający otrzymanie wiadomości. Po zakończeniu przesyłania danych master wysyła warunek stopu, którym jest zmiana stanu linii SDA i SCL na wysoki. Kolejność po raz kolejny musi zostać zachowana i dodatkowo na linii SCL pozostaje już stan wysoki.



Rys.3:Ekran startowy - wilgotność w normie.



Rys.4:Ekran startowy - zbyt niska wilgotność.

Najistotniejszy element systemu to czujnik wilgotności gleby M335. Jest to czujnik pojemnościowy, co wyróżnia go na tle większości czujników rezystancyjnych. Zasilany jest napięciem 3,3 - 6 VDC, a na wyjściu w zależności od poziomu wilgoci pojawia się napięcie z zakresu 0 ~ 3 VDC. Wyjście czujnika podłączone jest do wejścia analogowego A0, które mierzy napięcie od 0 do 5 VDC przy rozdzielczości 10 bitów, czyli z dokładnością 0,005V.

Urządzenie wyświetla również dane o temperaturze i wilgotności powietrza w pomieszczeniu. Realizuje to z pomocą czujnika DHT11. DHT11 zasilany jest napięciem 3,3 - 5 VDC, a dane przesyła w formie cyfrowej na pin PD4. Czujnik nie ma dużej dokładności (dokładność: 2°C, ±4 RH* (przy 25 °C)) , ale jego cena sprawia, że jest jednym z lepszych wyborów do zastosowań domowych.

Do sygnalizacji stanu obecnej wilgotności gleby służą: dioda LED RGB oraz buzzer. Katoda diody podłączona jest przez rezystor 220Ω do masy, anoda koloru zielonego jest podłączona do wyjścia cyfrowego PD7, anoda koloru czerwonego jest podłączona do wyjścia cyfrowego PD6. Buzzer jest podłączony do wyjścia cyfrowego PD3. Kiedy wilgotność gleby jest w

normie, pojawia się sygnał wysoki na wyjściu PD7, czyli świeci dioda zielona. Po spadnięciu wilgotności gleby poniżej pewnego poziomu, sygnał wysoki na PD7 znika (dioda zielona gaśnie), pojawia się sygnał wysoki na wyjściu PD6 (świeci dioda czerwona) oraz pojawia się sygnał PWM na wyjściu PD3 (buzzer zaczyna "brzęczeć").

Do obsługi automatycznego podlewania służą 3 elementy: przekaźnik, przetwornica step-up i pompa wody.

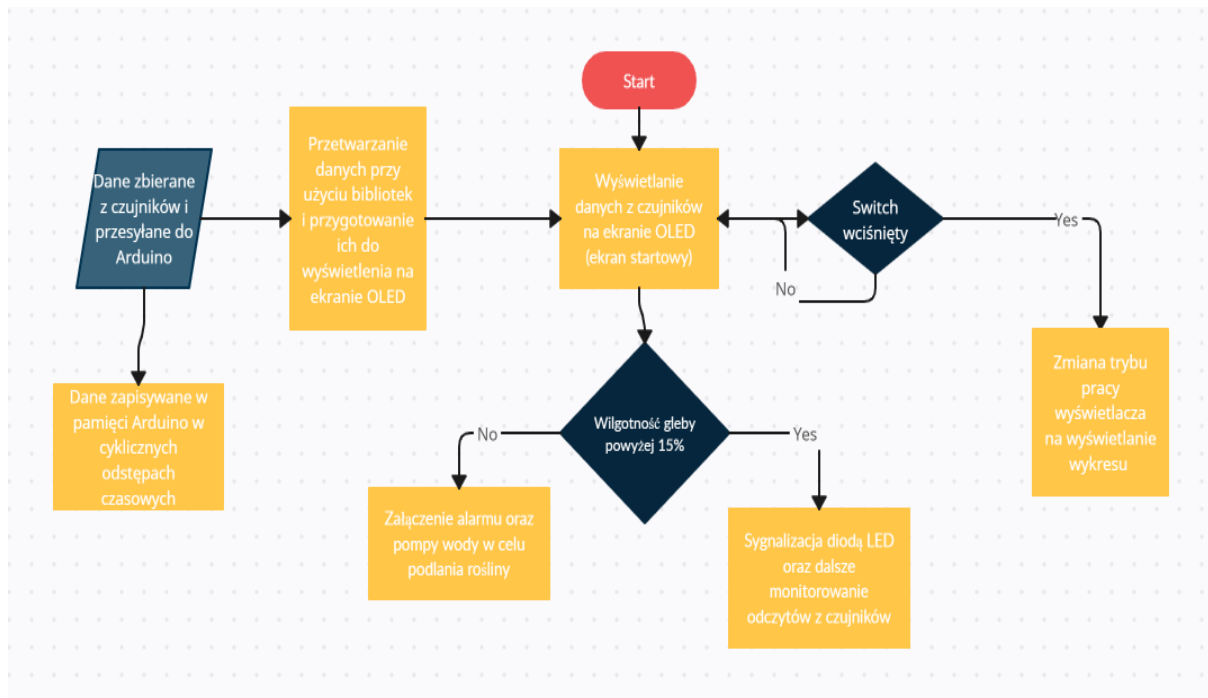
Cewka przekaźnika działa na napięciu 5V i jest podłączona do wyjścia cyfrowego PD8, które jest aktywowane przy spadku wilgotności gleby poniżej pewnego poziomu. Podanie stanu wysokiego powoduje przełączenie styku przekaźnika i połączenie między COM a NO. Po powrocie poziomu wilgotności gleby do poprawnego poziomu zanika stan wysoki na wyjściu PD8 i styk przekaźnika przełącza się, tworząc połączenie między COM a NC.

Przetwornica step-up 5-28V została użyta aby podbić napięcie zasilania 5V, które otrzymujemy na wyjściu typowej ładowarki np. do telefonu, do napięcia zasilania pompy wody czyli 12V. Przetwornica musiała być również użyta ze względu na duży pobór prądu pompy, który znacznie przekracza wydajność prądową płytki Arduino. Wyjście "-" przetwornicy podłączone jest do portu COM przekaźnika, a wyjście "+" do wejścia "+" zasilania pompy. Styk NO przekaźnika podłączony jest do wejścia "-" zasilania pompy, co po przełączeniu styku przekaźnika powoduje zamknięcie obwodu i zadziałanie pompy wody.

Pompa wody zasilana jest napięciem stałym 12V i może pobierać nawet 4,5W energii. Z tego względu konieczne było zastosowanie oddzielnego źródła zasilania. Przepływ wody może dochodzić nawet do 240L/h. Ważne jest aby pompa cały czas była zanurzona w wodzie, aby nie uległa uszkodzeniu podczas pracy na "sucho".

Urządzenie wyposażone jest również w switch, który odpowiada za sterowanie trybem pracy wyświetlacza OLED. Po wciśnięciu switch przekazuje stan wysoki na wejście cyfrowe PD2. Następnie sprawdzany jest stan wyjścia cyfrowego PD9. Jeśli na wejściu PD2 pojawi się stan wysoki, a na wyjściu PD9 jest stan niski to następuje zmiana wartości PD9 na stan wysoki i analogicznie jeśli na wejściu PD2 pojawi się stan wysoki i na wyjściu PD9 będzie stan wysoki to nastąpi jego zmiana na stan niski. Stan wyjścia PD9 determinuje tryb pracy wyświetlacza, który może wyświetlać aktualne dane z czujników i poziom wilgotności gleby lub wykres zmian poziomu wilgotności gleby w czasie z danych zapisywanych w określonym interwale czasowym.

4. Schemat blokowy algorytmu, kod programu:



Rys.7:Schemat blokowy algorytmu.

```
#include <EEPROM.h> // biblioteka używana do
zapisywania danych w pamięci arduino
#include <DFRobot_DHT11.h> // biblioteka używana do
obsługi czujnika temperatury i wilgotności powietrza DHT11
#include <Wire.h> // podstawowa biblioteka
arduino używana do komunikacji I2C/TWI
#include <Adafruit_SH1106.h> // biblioteka używana do
obsługi wyświetlacza OLED

#define OLED_RESET 4
Adafruit_SH1106 display(OLED_RESET);
DFRobot_DHT11 DHT;

unsigned long timer = 0;
const unsigned long interwal = 60000;
int address = 25; // deklaracja zmiennych
do obsługi wyświetlania wykresów z danych pomiarowych

void setup() {
```

```

EEPROM.begin();

pinMode(8, OUTPUT);
pinMode(PD3, OUTPUT);
pinMode(PD5, OUTPUT);
pinMode(PD6, OUTPUT);
pinMode(PD7, OUTPUT);
pinMode(PD4, INPUT);
pinMode(PD2, INPUT);
pinMode(9, OUTPUT);
Serial.begin(9600); // deklaracja używanych
pinów i rozpoczęcie transmisji szeregowej

display.begin(SH1106_SWITCHCAPVCC, 0x3C); // inicjalizacja ekranu
OLED, zdefiniowanie jego adresu
delay(2000);
display.clearDisplay();
}

void loop() {

int stanprzycisku = digitalRead(PD2);
if (stanprzycisku == HIGH & digitalRead(9) == LOW) {
    digitalWrite(9, HIGH);
    delay(500);
} else if (stanprzycisku == HIGH & digitalRead(9) == HIGH)
{
    digitalWrite(9, LOW);
} // funkcja sprawdzająca
stan przycisku w celu przełączenia trybu pracy

START:

display.clearDisplay();
DHT.read(PD4);
display.setTextColor(WHITE);
display.setCursor(0, 0);

```

```

display.setTextSize(1);
display.print("Temp. otoczenia:");
display.print(DHT.temperature);
display.setCursor(0, 10);
display.print("Wilg. powietrza:");
display.print(DHT.humidity);
display.print("%");
display.setCursor(0, 20);
display.print("Wilg. gleby:");
int wilgotnosc = analogRead(A0);
int wil = map(wilgotnosc, 182, 455, 100, 0);
display.setTextSize(1);
display.print(wil);
display.print("%");
display.setCursor(0, 30); // odczytywanie danych z
wejść cyfrowych i analogowych z czujników, przetwarzanie oraz
wyświetlanie ich na ekranie

if (wil < 15) {
    digitalWrite(PD7, LOW);
    display.setCursor(0, 40);
    display.setTextSize(1);
    display.println("Za niska wilgotnosc gleby!!");
    digitalWrite(PD3, HIGH);
    digitalWrite(PD6, HIGH);
    delay(100);
    digitalWrite(PD3, LOW);
    digitalWrite(PD6, LOW);
    digitalWrite(8, HIGH);
} // funkcja
odpowiadająca za działanie urządzenia podczas zbyt niskiego odczytu z
czujnika wilgotności gleby

if (wil > 15) {
    digitalWrite(PD6, LOW);
    display.setCursor(0, 40);
    display.setTextSize(1);
    display.print("Wilgotnosc gleby w normie :)");
    digitalWrite(PD7, HIGH);

```

```

    digitalWrite(8, LOW);
} // funkcja
odpowiadająca za działanie urządzenia podczas prawidłowego odczytu z
czujnika wilgotności gleby

if (millis() - timer > interwal) { //zapisywanie danych
do pamięci EEPROM arduino

    EEPROM.write(address, wil);
    Serial.println(EEPROM.read(address));
    Serial.println(timer);
    Serial.println(address);
    Serial.println(wil);
    address++;

    if (address > 124) { // resetowanie
wartości adresu po wypełnieniu ekranu
        address = 25;
    }
    timer = millis();
}

while (digitalRead(9) == HIGH) { // wyświetlanie
wykresu z danych zapisanych w pamięci arduino
    display.clearDisplay();
    display.setCursor(0, 0);
    display.print("Wykres wilgotnosci gleby:");
    display.setCursor(0, 54);
    display.print("0%");
    display.setCursor(0, 36);
    display.print("50%");
    display.setCursor(0, 20);
    display.print("100%");
    for (int i = 25; i < 124; i++) {
        int wil1 = EEPROM.read(i);
        int wil2 = map(wil1, 0, 100, 64, 20);
        if (wil2 < 20) {
            wil2 = 64;
        }
    }
}

```



```

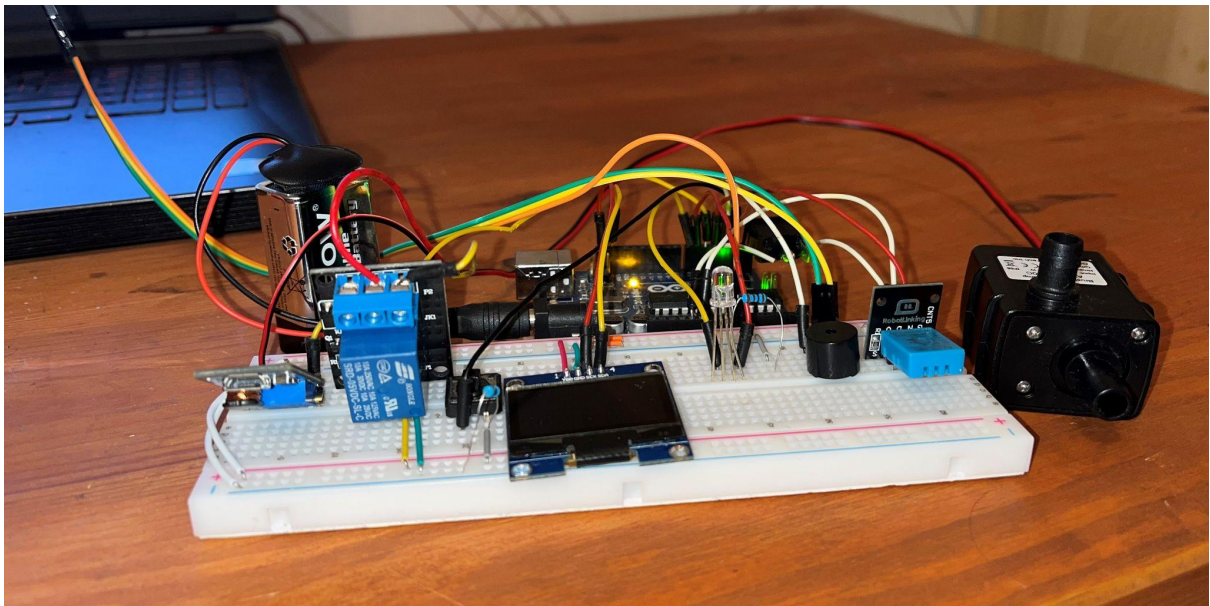
display.setCursor(i, 64);
display.drawLine(i, 64, i, wil2, 1);
display.display();
if (digitalRead(2) == HIGH) {
    digitalWrite(9, LOW);
    display.clearDisplay();
    break;
    goto START;
}
delay(50);
}

display.display();
delay(500);
}

```

5. Działanie gotowego urządzenia oraz dalszy rozwój projektu:

Urządzenie złożone na płytce prototypowej wygląda następująco:

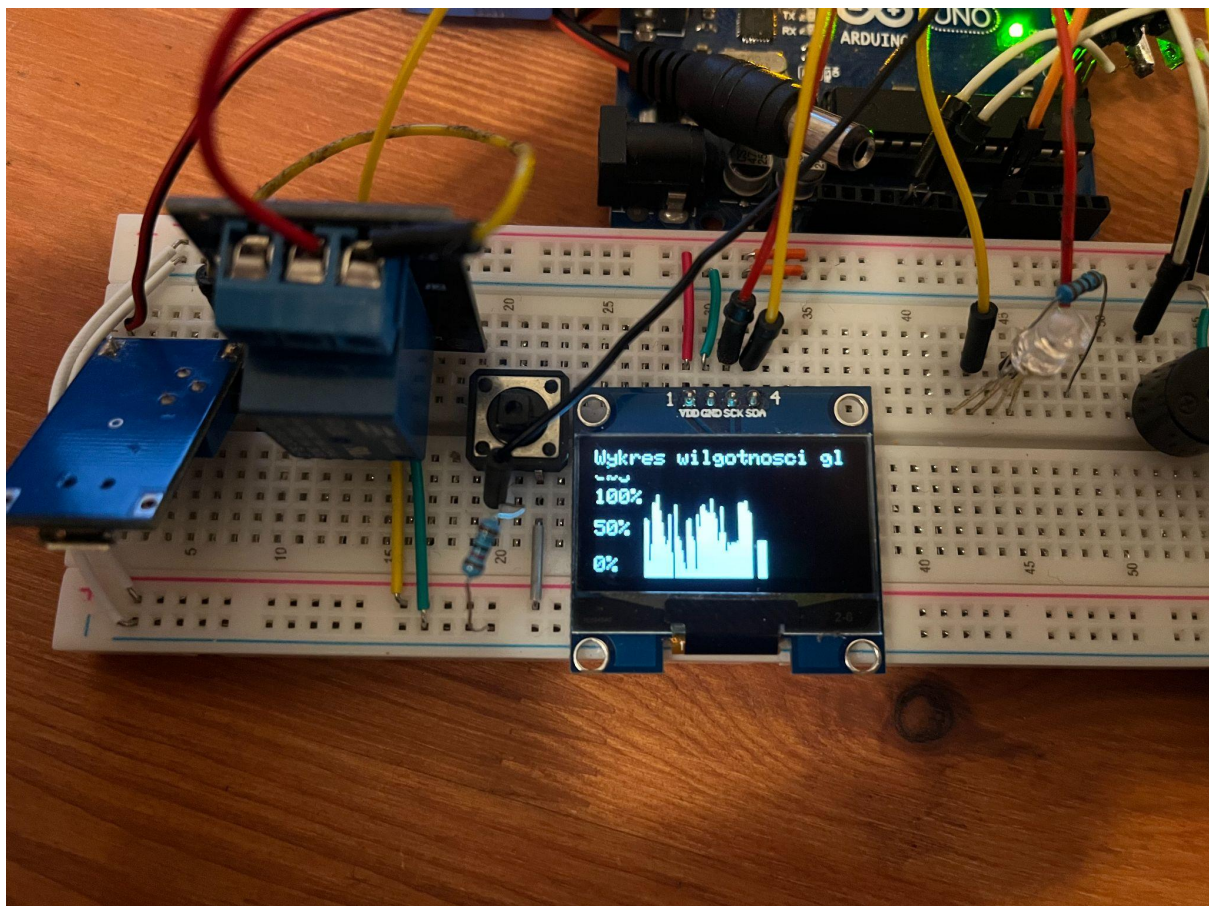


Rys.6:Urządzenie po złożeniu.

Po podłączeniu zasilania wyświetla się ekran startowy z podstawowymi informacjami takimi jak: wilgotność gleby, temperatura i wilgotność powietrza oraz informacja czy wilgotność gleby jest na odpowiednim poziomie. Nasuwa się pierwszy wniosek; ekran mógłby być nieco większy aby pomieścić więcej znaków (wyeliminowanie konieczności pisania skrótami).

Aktualizacje danych na wyświetlaczu odbywają się płynnie, informacje są czytelne. Problem pojawił się przy próbie przełączenia trybu pracy wyświetlacza za pomocą switcha. Nie każde wciśnięcie przycisku powoduje zmianę trybu. Skuteczność wynosi 100% przy dłuższym przytrzymaniu przycisku, jednak zbyt długie utrzymanie stanu wysokiego powoduje kolejne przełączenie. W efekcie tego wyświetlacz czasem potrafi na ułamek sekundy przełączyć się w tryb wyświetlania wykresów po czym od razu wrócić do ekranu startowego.

Wyświetlacz wyświetla wykres zmian poziomu wilgotności gleby w czasie. Dane z czujnika zapisują się poprawnie w równych odstępach czasowych. Do tej funkcjonalności chciałbym dodać nadpisywanie się wykresu w czasie rzeczywistym, czyli aby po wypełnieniu wykresu najstarsze dane były usuwane a reszta automatycznie przesuwiała się robiąc miejsce dla kolejnych.



Rys.7:Przykładowy wykres wilgotności gleby.

Proces inicjacji podlewania działa dobrze. Momentalnie po spadku poziomu wilgotności gleby przełącza się sygnalizacja dźwiękowa i świetlna oraz wysyłany jest sygnał do przekaźnika załączający pompę wody.

Proces podlewania działa poprawnie. Pompa okazała się nieco za mocna do podlewania roślin. Szybko dochodziła do maksymalnej prędkości i woda pod sporym ciśnieniem wylatywała z węża. Problem częściowo rozwiązałem zakrywając wylot węża i robiąc na nim małe otwory do wylotu wody. Zmieniłem też sposób ułożenia węża wewnątrz doniczki na spiralny. Z czasem chciałbym wymienić pompę wodę na taką o mniejszym przepływie.

Możliwość rozwoju jest tak naprawdę nieograniczona i zależy od wymagań i finansów użytkownika. W przyszłości chciałbym dodać możliwość wyboru gatunku rośliny do jakiej użyty jest system aby automatycznie dostosowywać parametry i funkcje układu. Obecnie dokonanie zmian parametrów jest możliwe tylko w programie, co wymaga podłączenia mikrokontrolera do komputera.

Urządzenie zostanie przeniesione do wodoodpornej obudowy co pozwoli na zastosowanie go również na zewnątrz.

Przeniesienie do obudowy będzie wymagało również przejście z płytki prototypowej na płytkę PCB oraz zamienienie Arduino UNO na mniejszy model Arduino NANO.

System pobiera dość dużo energii jak na zasilanie baterią 9V. W związku z tym planuje wyposażyć go w panel słoneczny montowany na obudowie, który wspomogę baterię w zasilaniu układu.

6. Bibliografia:

- dokumentacje techniczne użytych komponentów;
- *"Plant Power. Jak wyhodować szczęśliwe rośliny"* - Małgorzata Augustyn;
- *"The care of house plants"* - David Longman;
- <https://docs.arduino.cc/> - dokumentacja Arduino;