

# **Raport końcowy**

## **Inteligentna elektronika - Laboratorium otwarte**

### **Projekt: Inteligentna doniczka**

#### **Projekt zrealizowali:**

Kamil Błach 268524

Jakub Kołodziejczyk 268528

Mateusz Wilk 268558

Mateusz Kowalczyk 268533



## **Spis treści**

- 1. Wstęp**
- 2. Założenia projektowe**
- 3. LTspice**
  - a. Szczegółowy opis układu**
  - b. Symulacja**
- 4. Eagle**
- 5. Kod**
- 6. Obudowa**
- 7. Wykaz elementów**
- 8. Przebieg projektu(?)**
- 9. Wnioski**

## 1. Wstęp

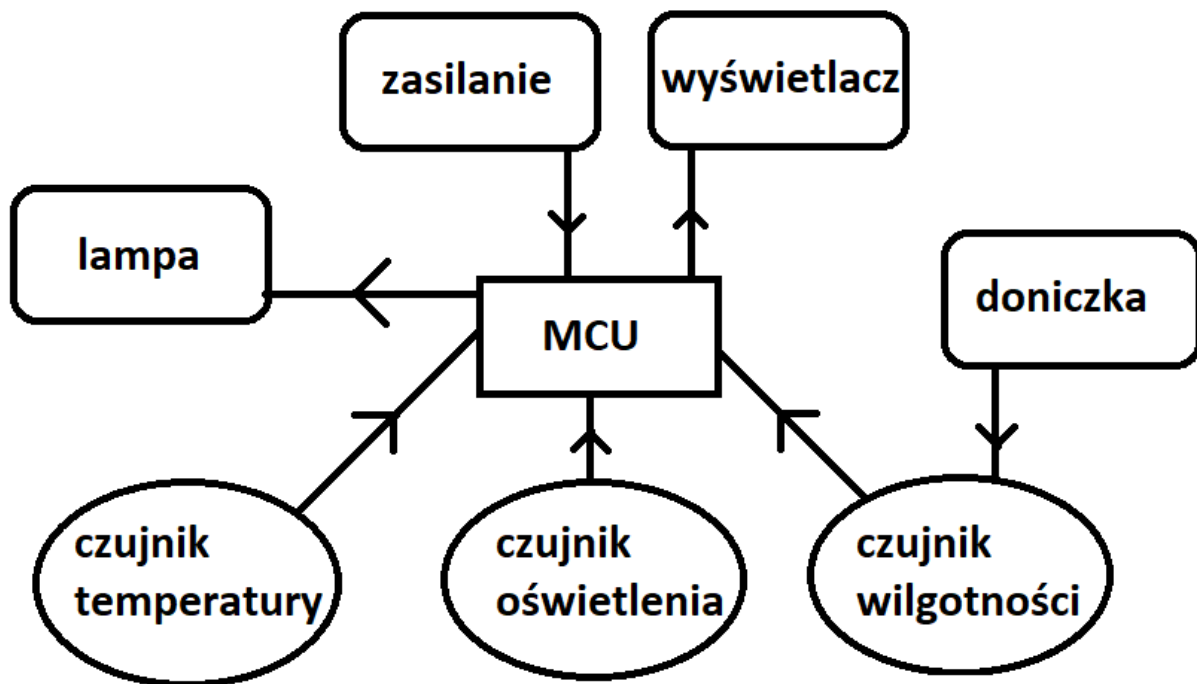
W ramach laboratorium otwartego nasza grupa zdecydowała się na projekt polegający na zaprojektowaniu inteligentnej doniczki. Na przestrzeni ostatnich lat ludzie coraz chętniej wybierają rośliny jako dodatek do domowej dekoracji, rozwiązanie takie wpływa pozytywnie nie tylko na wystrój wnętrz naszych domów, ale i na nasze samopoczucie i zdrowie. Wraz ze wzrostem popularności na dany produkt, rośnie również potrzeba wdrażania innowacyjnych rozwiązań, w opisywanym przypadku rozwiązania takie mogą dotyczyć np. pielęgnacji roślin i zapewnienia optymalnych warunków do rozwoju. W myśl tego zjawiska postanowiliśmy opracować doniczkę umożliwiającą ciągłe monitorowanie kluczowych parametrów takich jak: temperatura otoczenia, wilgotność gleby czy natężenie oświetlenia.

Rozwój inteligentnych doniczek do codziennego użytku to krok w stronę zachęcenia większej ilości ludzi do hodowania roślin w domu, jest więc to zagadnienie mające możliwość istotnego wpływu zarówno na pojedynczego człowieka, jak i na środowisko. Dodatkowo, jak wcześniej zostało zauważone, jest to innowacja w zagadnieniu stale zyskującym na popularności, a to oznacza, że jest to temat z potencjałem na generowanie znacznych zysków i pozyskanie inwestorów.

## 2. Założenia projektowe

Nasz projekt zakłada zaprojektowanie od podstaw inteligentnej doniczki(Rys.1). Do jej głównych funkcji będzie należeć:

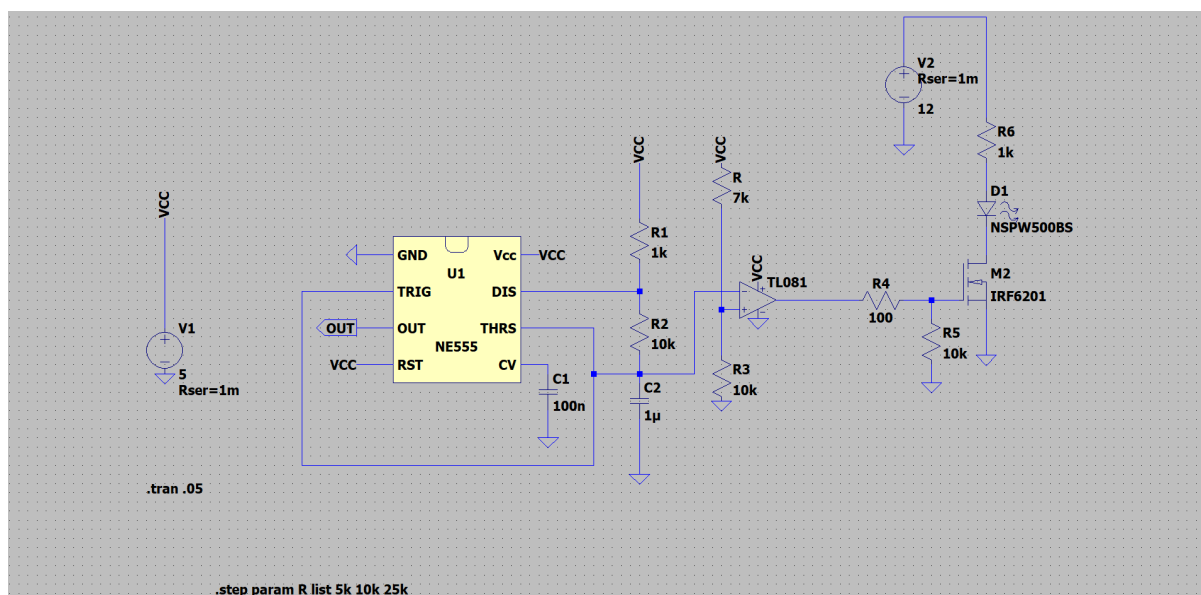
- Wyświetlanie temperatury otoczenia na dołączonym wyświetlaczu
- Wyświetlanie natężenia światła na dołączonym wyświetlaczu
- Wyświetlanie wilgotności gleby na dołączonym wyświetlaczu, z dodatkową informacją o aktualnym stanie nawodnienia kwiatka, dostępne możliwości to: "Podlej", "Podlana" i "Przelana"
- Automatyczna regulacja oświetlenia dołączonego do doniczki na podstawie odczytu z czujnika natężenia światła, zapewnia to stałe i optymalne warunki do rozwoju rośliny niezależnie od pory dnia i warunków pogodowych.
- Do przetwarzania danych z czujników i wyświetlania ich na dołączonym wyświetlaczu LCD wykorzystany zostanie mikrokontroler Arduino Uno R3, natomiast kod odpowiedzialny za procedure tego przetwarzania napisany zostanie w środowisku Arduino IDE.
- Do sterowania oświetleniem wykorzystany zostanie zaprojektowany przez nas układ analogowy generujący sygnał PWM ze zmiennym wypełnieniem zależnym od odczytu z czujnika natężenia światła(fotorezystorem)
- Wybór obudowy zapewniającej schludny, stosunkowo innowacyjny oraz minimalistyczny wygląd



Rys.1 Schemat poglądowy założeń projektowych

### 3. LTspice

Do zaprojektowania analogowej części układu wykorzystano program LTspice (Rys.2). Jest to część układu odpowiedzialna za generowanie sygnału PWM o zmiennym wypełnieniu, a tym samym sterująca oświetleniem doniczki.



Rys.2 Schemat układu analogowego wykonanego w programie LTSpice

### a) Szczegółowy opis układu

Głównym elementem powyższego układu analogowego jest układ scalony NE555(Rys.3). Odpowiada on za generację sygnału PWM, układ ten może pracować w trybie monostabilnym, bądź astabilnym. W naszym projekcie wykorzystujemy tryb astabilny w celu generacji ciągłego sygnału prostokątnego.

Rezystory R1, R2 odpowiadają za określenie częstotliwości generowanego sygnału.

Kondensator C1 służy jako kondensator bocznikowy i zapewnia zasilanie o niskiej impedancji, stabilizuje również działanie generatora PWM.

Wzmacniacz operacyjny TL081 pełni rolę komparatora, porównując napięcie odniesienia z napięciem z dzielnika napięcia(w rzeczywistej wersji z napięciem z mikrokontrolera Arduino Uno R3).

Tranzystor MOSFET IRF740 pod wpływem generowanego sygnału PWM włącza się i wyłącza sterując diodą LED w sposób umożliwiający manipulację natężeniem świecenia tych diod.

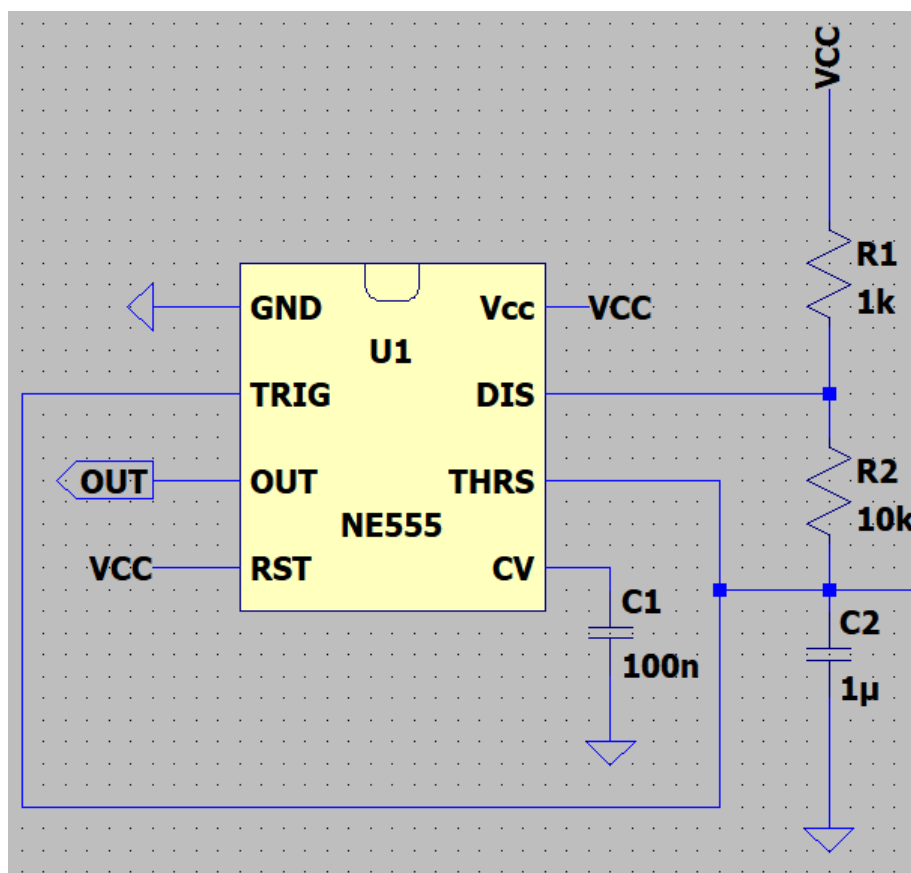
Pasek diod LED SMD5050 12V IP65 emituje światło o natężeniu zależnym od współczynnika wypełnienia PWM, jej zasilanie kontrolowane jest przez tranzystor.

Źródło zasilania V1 dostarcza napięcie zasilania o wartości 5V do układu, jest to napięcie podawane przez mikrokontroler.

Kondensator C2 oraz rezystor R3 filtrują napięcie odniesienia komparatora.

Rezystor R6 ogranicza prąd płynący przez pasek diod LED, chroniąc je tym samym przed przepięciami.

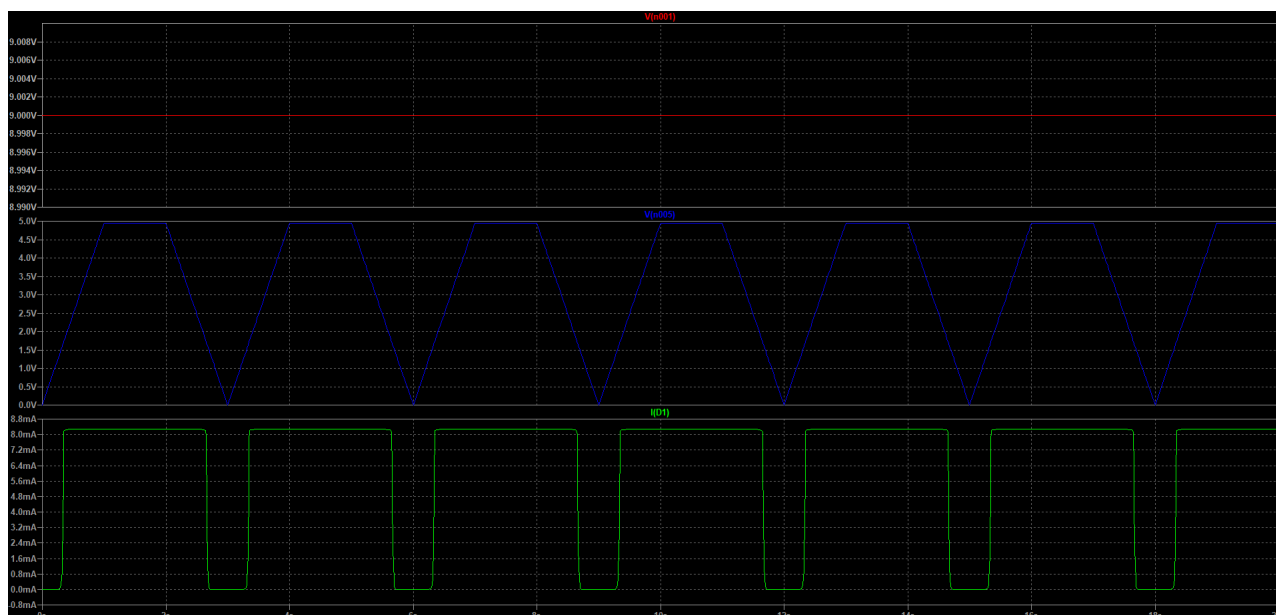
Źródło zasilania V2 ma wartość 9V i dostarcza napięcie do obwodu diod LED.



Rys.3 Część schematu przedstawiająca układ NE555

## b) Symulacja

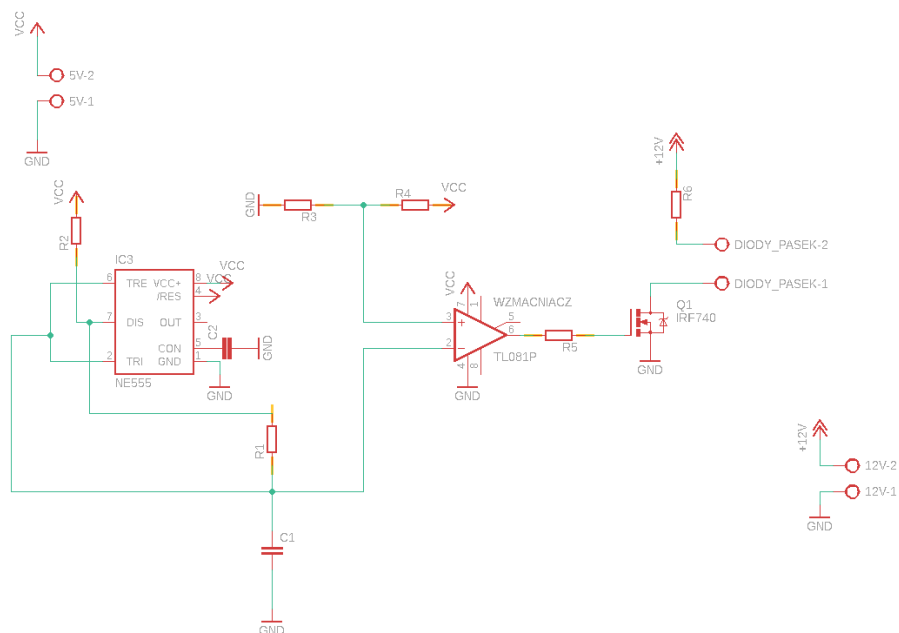
Przeprowadzono symulację działania, wcześniej omawianego układu zaprojektowanego w programie LTspice. Podczas symulacji (Rys.4) zbadano napięcie zasilania obwodu diod ( $V(n001)$ ), sygnał PWM podawany na bramkę tranzystora ( $V(n005)$ ) oraz prąd indukowany na diodach LED ( $I(D1)$ ). Z wykorzystaniem kursorów ustalono, że napięcie na bramce od którego prąd zaczyna przepływać przez diody wynosi 1.72V, jest to wartość  $V_{th}$  (Gate Threshold Voltage) poniżej zakresu podanego przez producenta, jednak należy wziąć pod uwagę, że warunki dla których te zakresy zostały podane znacznie odbiegają od warunków w zaprojektowanym, na potrzeby tego projektu układzie.



Rys.4 Symulacja działania układu w programie LTspice

#### 4. Eagle

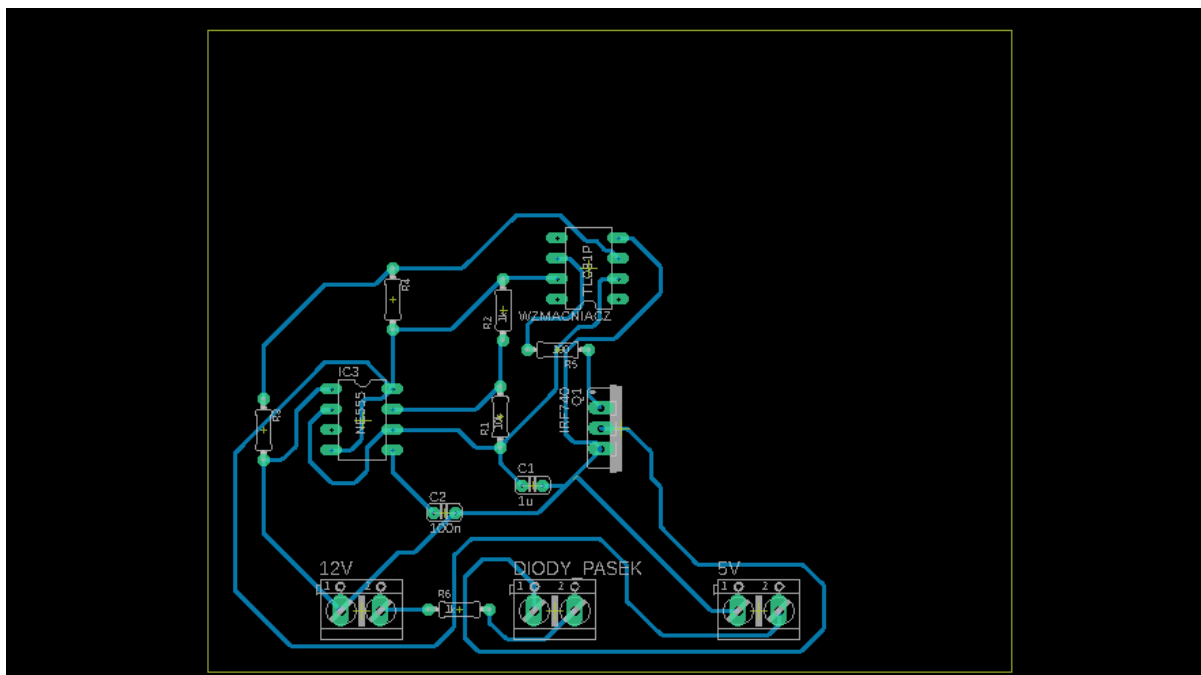
Następnym krokiem po zaprojektowaniu układu i jego symulacji w programie LTspice było wykonanie schematu płytki PCB w programie Eagle. W pierwszej kolejności należało wykonać schemat układu w sposób podobny do wykonanego uprzednio w LTspice, z tą różnicą, że schemat w programie Eagle uwzględnia wszystkie bloki układu, jak i wymaga dobrania złączy kołkowych, listw zaciskowych, bądź innego elementu umożliwiającego podłączenie zasilania(Rys.5). Na schemacie można wyróżnić dwa bloki zasilające, główny obwód NE555 oraz wzmacniacza wykorzystywanego jako komparator i obwód paska diod LED z tranzystorem pełniącym rolę klucza między obwodem głównym, a obwodem diod LED.



Rys. 5. Schemat układu analogowego w programie Eagle z uwzględnieniem listw zaciskowych

Następnie wykonano projekt płytki obwodu drukowanego z topologią połączeń (Rys.6), wszystkie ścieżki poprowadzono w warstwie dolnej, przy projektowaniu stosowano się do zasad zamieszczonych w dołączonym pliku "OpenLab.dru". Popelniono błąd przy dobieraniu wielkości pól lutowniczych, jednak po wytrawieniu płytki udało się zrealizować w pełni działający układ. Topologia połączeń została przemyślana i kilkakrotnie była optymalizowana, tak aby spełniała kryteria i jednocześnie była uproszczona w jak największym stopniu. Zastosowano trzy listwy zaciskowe - anody znajdują się po prawej stronie, natomiast katody po lewej.





Rys. 6. Projekt płytki PCB w programie Eagle z topologią połączeń

## 5. Kod

Kod został przygotowany w środowisku Arduino IDE.

```
C/C++
#include <LiquidCrystal.h>
#include <OneWire.h>
#include <DallasTemperature.h>

#define ONE_WIRE_BUS 12

// Setup OneWire
OneWire oneWire(ONE_WIRE_BUS);
DallasTemperature sensors(&oneWire);

// deklaracja portów LCD:
LiquidCrystal lcd (2, 3, 4, 5, 6, 7);
```

List. 1. Deklaracyjna część kodu.

W powyższym listingu deklarowane jest korzystanie z biblioteki dedykowanej do wyświetlacza LCD dla środowiska arduino oraz dwie biblioteki do obsługi czujnika temperatury przez interfejs one wire. Dodatkowo określono, które piny zostaną przeznaczone do obsługi wyświetlacza.

```

C/C++
void setup() {
  Serial.begin(9600);
  // biblioteka czujnik temperatury
  sensors.begin();
  setPwmFrequency(9, 64);
  // lcd start:
  lcd.begin (16,2);
  lcd.clear();
  // czujniki:
  pinMode(A5, INPUT); // fotorezystor
  pinMode(A4, INPUT); // wilgotności gleby
  // piny do płytek:
  pinMode(A1, OUTPUT);
  // pin do wysyłania info do płytki:
  pinMode(9, OUTPUT);
}

```

List. 2. Zawartość funkcji setup

W tej funkcji określono, które piny zostały przewidziane na poszczególne czujniki. Pin nr 9 przeznaczony jest do wysyłania sygnału o odpowiednim napięciu na wejście płytki drukowanej. Funkcja Serial.begin służy do konfiguracji interfejsu one wire a sensors.begin to funkcja inicjująca transmisję z czujnikiem (jest to funkcja z dedykowanej biblioteki od producenta).

Poniżej umieszczono kolejne elementy kodu znajdujące się w głównej funkcji programu "loop".

```

C/C++
//odczyt temperatury:
sensors.requestTemperatures();

// wyświetl temperature:
lcd.setCursor(0,1);
lcd.print("T:");
lcd.print(sensors.getTempCByIndex(0));
//lcd.print((int)tempKoncowa);
lcd.print("C");

```

List. 3. Część kodu realizująca odczyt temperatury z czujnika i wyświetlająca tą wartość na wyświetlaczu LCD

```

C/C++
// odczyt natężenia światła:
int odczytRezyst = analogRead(A5);

```

```

int NatSwiatla = map(odczytRezyst, 0, 1023, 0, 100); // pokazanie wartości
w %

// wyświetl natężenie światła:
lcd.setCursor(8, 1);
lcd.print(" SW:");
lcd.print(NatSwiatla);
lcd.print("%");

```

List. 4. Część kodu realizująca odczyt natężenia światła z fotorezystora i wyświetlającą tą wartość na wyświetlaczu LCD

```

C/C++

// Sterowanie jasnością światła w zależności od natężenia światła:
(nie można dawać zera), 1,7[V] = zaczyna świecić, 3[V] = maksymalnie
if (NatSwiatla >= 50) {
    analogWrite(9, 80);
} else if (NatSwiatla <= 49 && NatSwiatla >= 40) {
    analogWrite(9, 100);
} else if (NatSwiatla <= 39 && NatSwiatla >= 30) {
    analogWrite(9, 120);
} else if (NatSwiatla <= 29 && NatSwiatla >= 20) {
    analogWrite(9, 130);
} else if (NatSwiatla <= 19 && NatSwiatla >= 10) {
    analogWrite(9, 140);
} else if (NatSwiatla <= 9 && NatSwiatla >= 0) {
    analogWrite(9, 160);
}

```

List. 5. Część kodu wysyłająca sygnał o zadanym napięciu na wejście układu sterującego paskiem LED.

Wyrażona w procentach wartość natężenia światła padającego na doniczkę została podzielona na zakresy. W zależności od zakresu, w który został odczytany dostosowane jest napięcie sygnału wychodzącego z mikrokontrolera.

```

C/C++

//odczyt wilgotności gleby:
int wilGleb = analogRead(A4);
int wilGlebProc = map(wilGleb, 0, 1023, 0, 100); // pokazanie wartości w %
int wilGlebProcPopr = 100 - wilGlebProc; // odwrócenie wartości

//wyświetl wilgotnosc gleby:
lcd.setCursor(0,0);
lcd.print("W: ");
lcd.print(wilGlebProcPopr);

```

```
lcd.print("% ");
```

List. 6. Część kodu realizująca odczytywanie, przetwarzanie i wyświetlanie procentowej wartości wilgotności gleby.

```
C/C++
lcd.setCursor(7, 0);
lcd.print("");

if (wilGlebProcPopr >= 76)
{
    lcd.print("Przelana ");
} else if (wilGlebProcPopr <= 75 && wilGlebProcPopr >= 40)
{
    lcd.print("Podlana");
} else if (wilGlebProcPopr <= 50)
{
    lcd.print("Podlej ");
}
// opóźnienie
delay(3000);
```

List. 7. Część kodu realizująca wyświetlanie komunikatów na wyświetlaczu.

Na końcu programu przygotowana została funkcja opóźniająca działanie pinu wysyłającego napięcie na wejście układu analogowego.

```
C/C++
// Funkcja PWM:
void setPwmFrequency(int pin, int divisor) {
    byte mode;
    if (pin == 9 || pin == 10) {
        switch (divisor) {
            case 1: mode = 0x01; break;
            case 8: mode = 0x02; break;
            case 64: mode = 0x03; break;
            case 256: mode = 0x04; break;
            case 1024: mode = 0x05; break;
            default: return;
        }
        TCCR1B = TCCR1B & 0b11111000 | mode;
    }
}
```

List. 8. Funkcja zmniejszająca częstotliwość generowanego PWM na pinie.

W zależności od podanej wartości częstotliwość wygenerowanego na pinie sygnału PWM jest podzielona o preskaler. W projekcie zastosowany został preskaler 64 (List. 1).

## **6. Obudowa**

Proces przygotowania obudowy dla inteligentnej doniczki był kluczowym etapem projektu, mającym na celu zapewnienie optymalnych warunków dla prawidłowego funkcjonowania wszystkich komponentów elektronicznych, jednocześnie gwarantując estetyczny wygląd oraz wygodę użytkowania. Głównym celem było stworzenie takiej struktury, która oddzieli część elektroniczną od ziemi, wody oraz samej rośliny, co zapewni ochronę przed ewentualnymi uszkodzeniami spowodowanymi wilgocią czy innymi czynnikami zewnętrznymi. Dodatkowym wyzwaniem była minimalizacja rozmiarów, aby całość była kompaktowa i estetyczna.

Obudowa doniczki została dobrana w sposób umożliwiający łatwy dostęp do komponentów elektronicznych, jednocześnie zapewniając ich ochronę. Konstrukcja doniczki została podzielona na dwie części. Górna część pełni rolę klasycznej doniczki, w której znajduje się ziemia oraz roślina, natomiast dolna część jest miejscem, gdzie umieszczone są wszystkie układy elektroniczne oraz płytki drukowane.

Dolna część doniczki może zostać łatwo odłączona, co umożliwia łatwy dostęp do układów elektronicznych w przypadku potrzeby ich serwisowania lub modyfikacji. Proces ten jest niezwykle prosty i intuicyjny, co znacznie ułatwia użytkowanie doniczki na co dzień. W bocznej części doniczki umieściliśmy również wyświetlacz, który jest dopasowany w wycięty wcześniej otwór, co wymagało dokładnego i precyzyjnego cięcia przy użyciu wyrzynarki.

W górnej części doniczki, poza przestrzenią na roślinę i ziemię, wykonaliśmy również otwór, który ma na celu umożliwienie przeprowadzenia przewodów. Prowadzą one do zewnętrznych elementów, takich jak lampa LED oraz czujniki wilgotności i światła. Otwory zostały wywiercone w sposób zapewniający minimalne ingerowanie w estetykę doniczki, jednocześnie umożliwiając wygodne rozmieszczenie wszystkich niezbędnych komponentów.

Całość konstrukcji została zaprojektowana z myślą o minimalizacji rozmiarów, co pozwoliło na stworzenie kompaktowego i estetycznego urządzenia. Dzięki temu inteligentna doniczka nie tylko spełnia swoje funkcje techniczne, ale również stanowi elegancki dodatek do wnętrza.

## 7. Wykaz elementów

Elementy wykorzystane do przygotowania płytki drukowanej:

- NE555
- Rezystory:
  - 2x 10k $\Omega$
  - 1x 100 $\Omega$
  - 2x 1k $\Omega$
- Kondensatory:
  - ceramiczny 100nF
  - elektrolityczny 1 $\mu$ F
- Wzmacniacz operacyjny TL081
- Tranzystor polowy IRF740

Poza elementami niezbędnymi do przygotowania płytki drukowanej generującej sygnał PWM, w projekcie wykorzystano:

- Czujniki:
  - czujnik temperatury DS18B20 - cyfrowy czujnik temperaturowy z interfejsem one-wire
  - czujnik wilgotności gleby firmy OEM - analogowy czujnik składający się z sondy pomiarowej i modułu detektora
  - fotorezystor 10k $\Omega$  - pełniący rolę czujnika naświetlenia
- Arduino UNO R3 jako serce całego układu
- Wyświetlacz LCD
- Pasek LED SMD5050 12V IP65 pełniący rolę źródła światła
- Rezystory: x2 1[k $\Omega$ ], x1 100 [ $\Omega$ ], x1 10 [k $\Omega$ ]
- Kondensatory: x1 100 [nF], x1 1 [ $\mu$ F]
- Doniczka Agate 23 cm czarna VERDENIA
- Tranzystor IRF740

Całość układu jest zasilana bateryjnie z 6 baterii umieszczonych w koszyku.

## 8. Przebieg projektu(?)

## 9. Wnioski

W ramach projektu udało się zrealizować wszystkie założenia projektowe, wykonana inteligentna doniczka może zostać wykorzystana do hodowania rośliny w sposób kontrolowany i do pewnego stopnia zautomatyzowany. Wszystkie wykorzystane elementy pracują zgodnie z założeniami. Możliwym jest dalsze rozszerzenie funkcjonalności doniczki, w oparciu o wykonany projekt można uzupełnić finalny produkt o pompę wody umożliwiającą automatyczne i precyzyjne podlewanie rośliny na podstawie odczytu z czujnika wilgotności gleby. W przypadku

niektórych roślin istotnym czynnikiem może być również temperatura, wykorzystując już zamontowany i zaprogramowany czujnik temperatury, istnieje możliwość rozszerzenia projektu o moduł chłodzący/podgrzewający otoczenie w bezpośredniej odległości od rośliny. Produkt uzupełniony o wyżej wymienione funkcjonalności byłby możliwy do sprzedaży. Dzięki realizacji tego projektu wszyscy jego wyżej podpisani wykonawcy mieli okazję na rozwinięcie swoich zdolności,