

SPRAWOZDANIE				PROSZĘ PODAĆ NR GRUPY:						
				ZIISS1	3	5	1	2	IO	
IMIE	NAZWISKO	Temat ćwiczenia zgodny z wykazem tematów:		PONIŻEJ PROSZĘ PODAĆ TERMIN ZAJĘĆ:				ROK:		
Amelia	Lis	Podstawy programowania mikroprocesorów Przetworniki A/C						2023r.		
				PN	WT	SR	CZ	PT	SB	ND
				GODZINA ROZPOCZĘCIA ZAJĘĆ:						11 : 30
UWAGA !!! Wypełniamy tylko białe pola. W punkcie 1, proszę zakreślić odpowiednie pola i podać godzinę w której odbywają się zajęcia, zgodnie z planem zajęć.										

Wprowadzenie teoretyczne:

Opisz rodzaje przetworników analogowo cyfrowych spotykanych w mikrokomputerach jednoukładowych.

Przetworniki analogowo-cyfrowe (ADC) w mikrokomputerach jednoukładowych stanowią kluczowy element, umożliwiający konwersję sygnałów analogowych na cyfrowe. Ich różnorodność oraz zaawansowane funkcje sprawiają, że są nieodzownym komponentem w dzisiejszych systemach elektronicznych.

Pierwszym rodzajem przetworników jest sukcesywny aproksymacyjny przetwornik ADC. Jest to popularna i efektywna metoda, w której sygnał jest porównywany z napięciem referencyjnym. Proces ten jest powtarzany sukcesywnie, aż do uzyskania dokładnej cyfrowej reprezentacji sygnału wejściowego. Sukcesywne aproksymacyjne przetworniki są często wykorzystywane w aplikacjach wymagających wysokiej rozdzielczości i dokładności.

Kolejnym ważnym rodzajem jest przetwornik sigma-delta. Charakteryzuje się on wysoką rozdzielczością i jest często stosowany w zastosowaniach audio i pomiarowych. Przetworniki sigma-delta są zdolne do osiągania bardzo wysokich stosunków sygnał/szum, co sprawia, że są idealne do precyzyjnych zastosowań, takich jak konwersja dźwięku na sygnał cyfrowy.

Innym istotnym typem przetworników ADC w mikrokomputerach jednoukładowych są przetworniki rampowe. Te przetworniki mierzą czas, jaki potrzebny jest na naładowanie kondensatora od zera do wartości proporcjonalnej do sygnału wejściowego. Pomiar ten jest następnie przeliczany na wartość cyfrową. Przetworniki rampowe znajdują zastosowanie w wielu prostych aplikacjach pomiarowych.

W kontekście mikrokomputerów jednoukładowych przetworniki ADC są niezbędne do konwersji sygnałów z otaczającego środowiska na dane cyfrowe, zrozumiałe dla mikrokontrolera. Wprowadzenie precyzyjnych przetworników ADC do mikrokomputerów jednoukładowych umożliwia rozwój zaawansowanych systemów pomiarowych, kontroli i komunikacji.

Jednakże, implementacja przetworników ADC w mikrokomputerach jednoukładowych niesie ze sobą pewne wyzwania. Jednym z nich jest równowaga pomiędzy rozdzielczością a prędkością konwersji. W praktyce często zachodzi potrzeba osiągnięcia jak największej dokładności pomiarów przy jednoczesnym utrzymaniu odpowiedniej szybkości przetwarzania. To wyzwanie wymaga zastosowania zaawansowanych technik projektowania oraz optymalizacji.

Podsumowując, przetworniki analogowo-cyfrowe w mikrokomputerach jednoukładowych stanowią kluczowy element współczesnych systemów elektronicznych. Sukcesywny aproksymacyjny, sigma-delta i rampowe to tylko niektóre z rodzajów przetworników ADC, które znajdują zastosowanie w różnorodnych aplikacjach. Wraz z rosnącymi oczekiwaniami co do precyzji pomiarów, rozwój tych przetworników staje się coraz bardziej istotny, wymagając równoczesnego uwzględnienia aspektów związanych z rozdzielczością, prędkością konwersji i optymalizacją.

* rysunki proszę zamieszczać na drugiej stronie a w tekście podać odnośniki

Zadanie 1

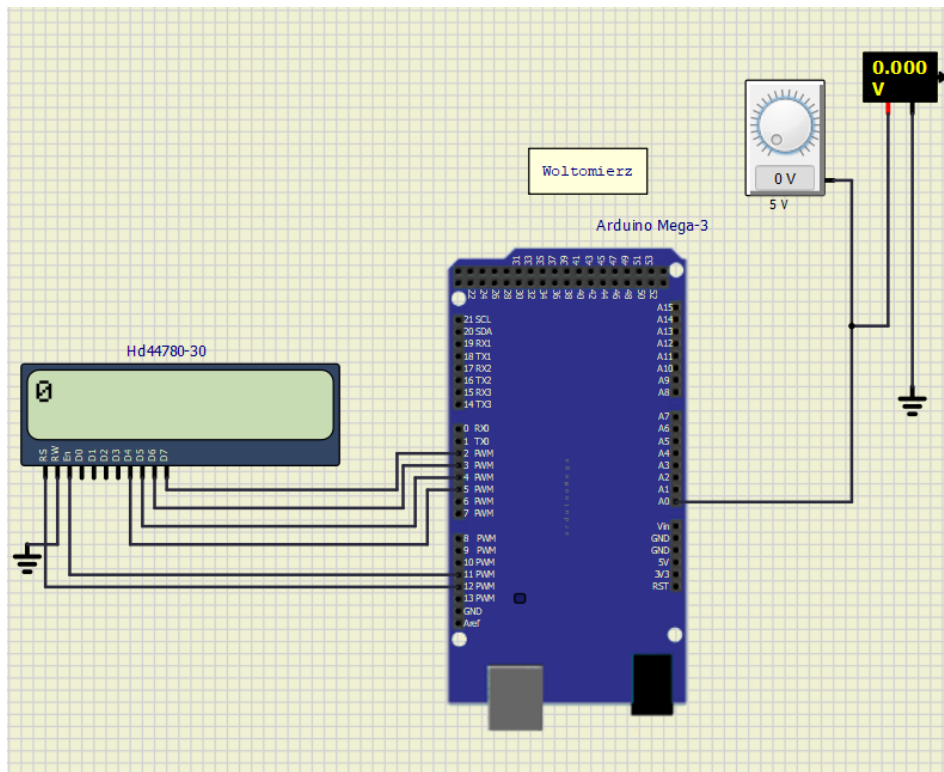
Zestaw schemat jak na rysunku. Zbuduj cyfrowy woltomierz z wyświetlaczem LCD na którym wartości będą wyświetlane w voltach. Napięcie na woltomierzu będzie regulowane w zakresie od 0 do 5V

Uwaga, piny D4 do D7 mają odpowiadać czterem ostatnim cyfom Twojego numeru indeksu.

RS i En To dwie pierwsze cyfry Twojego numeru indeksu.

Podłączenie na rysunku odpowiada numerowi indeksu: 122345 czyli piny 11 12 2,3,4,5. Przetwornik podłącz do dowolnego pinu analogowego

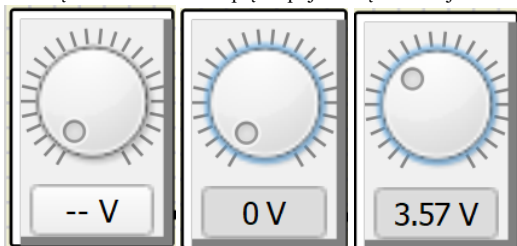
Schemat układu:



Na zrzucie ekranu widać wyświetlacz ciekłokrystaliczny, płytke Arduino Mega oraz źródło regulowane źródło napięcia i woltomierz kontrolny.

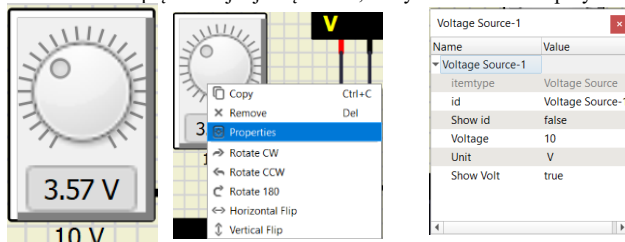
Aby rozpocząć symulację należy przycisnąć przycisk na u góry ekranu a następnie załączyć przyciskiem zmienne źródło napięcia i pokrętełm regulować jego nastawy.

Po załączeniu na źródle napięcia pojawi się liczba i jednostka V (volt).



Rys. 1 Regulowane źródło napięcia A- wyłączone B-załączone C- ustawione 3.75V

Pod źródłem napięcia znajduje się zakres, który można zmienić przyciskając prawy przycisk myszy i zmieniając jego właściwości.



Rys. 2 Konfiguracja regulowanego źródła napięcia.

Wskazówka:

Oprogramowanie wyświetlacza Hd44780 komendy:

```
const int rs = 12, en = 11, d4 = 5, d5 = 4, d6 = 3, d7 = 2;  
LiquidCrystal lcd(rs, en, d4, d5, d6, d7);
```

Oprogramowanie przetwornika analogowocyfrowego:

```
int potencjometr = A0;  
int war = 0;  
war = analogRead(potencjometr);
```

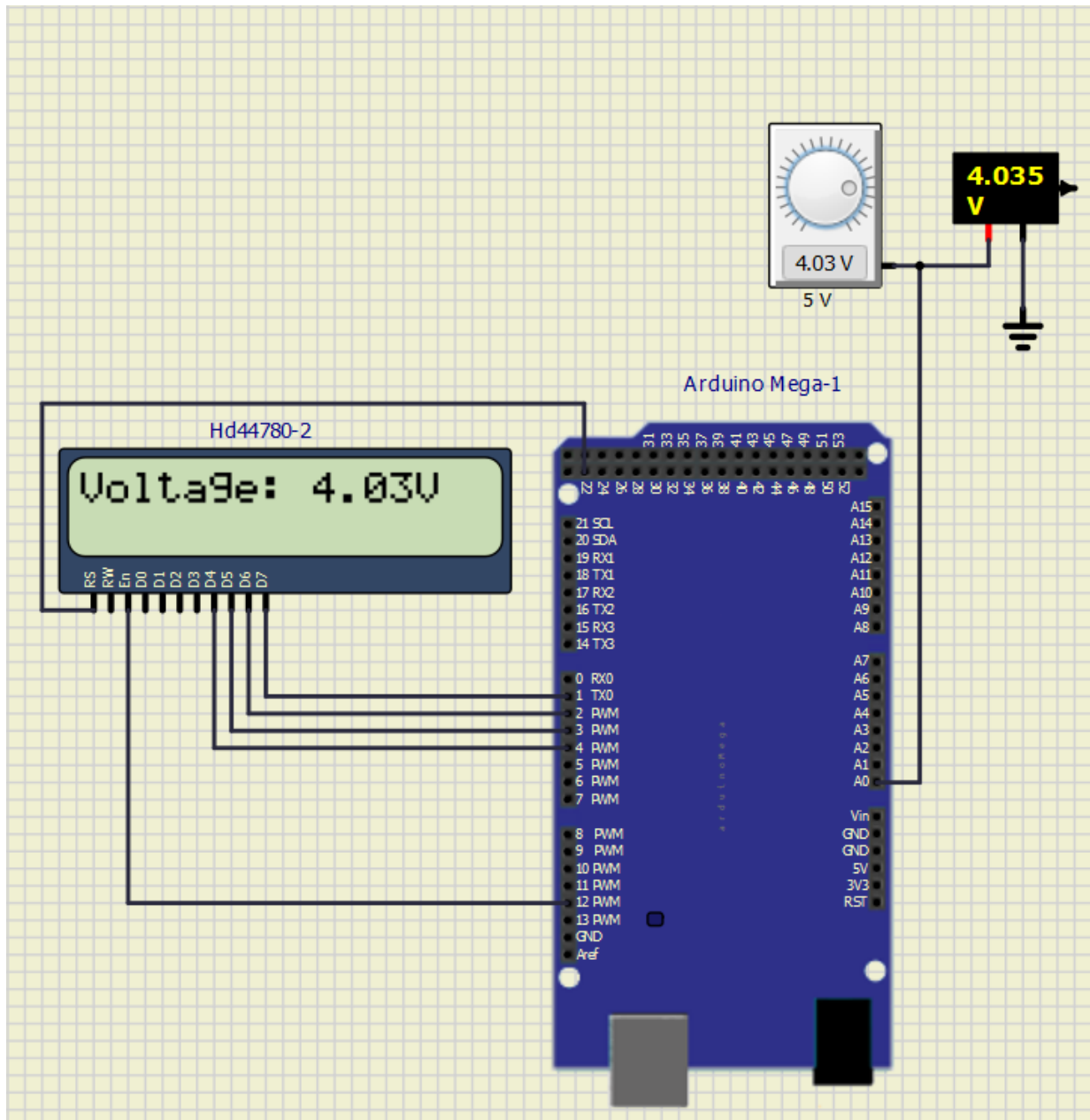
Potencjometr przetwarza napięcie podawane na port np. A0 w zakresie od 0 do 1023.

Pamiętaj, że na port można podać tylko napięcie z zakresu od 0 do 5V

Gdzie: 0 odpowiada wartości bitowej 0 a 5V wartości bitowej 1023.

Na wyświetlaczu należy wyświetlić napięcie w woltach V zastanów się jak należy to oprogramować.

Zrzut ekranu Schematu układu zaprojektowanego przez Ciebie.



Tutaj proszę wkleić kod programu.

```
//221423
// 1 4 2 3 -> D7 -> 1 D6 -> 2 D5 -> 3 D4 -> 4
//22 -> RS
//12 -> En
#include <LiquidCrystal.h>
const int rs = 22, en = 12, d4 = 4, d5 = 3, d6 = 2, d7 = 1;
LiquidCrystal lcd(rs, en, d4, d5, d6, d7);

void setup() {
  lcd.begin(16, 2);
}

void loop() {
  int potPin = A0;
  int potValue = analogRead(potPin);
```

```
// analogowe wartości zmapowane na wolty (0-1023 to 0-5V)
float voltage = map(potValue, 0, 1023, 0, 5000) / 1000.0;

lcd.clear();
lcd.setCursor(0, 0);
lcd.print("Voltage: ");
lcd.print(voltage);
lcd.print("V");

delay(500);
}
```

Zadanie 2

Zbuduj wielozakresowy cyfrowy woltomierz z wyświetlaczem LCD,) na którym wartości będą wyświetlane w woltach. Napięcie na woltomierzu będzie regulowane w zakresie od 0 do (napięcia, które definiują ostatnie trzy cyfry Twojego numeru indeksu)
Na przykład: Nr 222123. Napięcie, do którego ma mierzyć woltomierz to 123 V

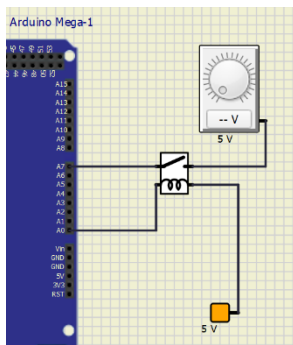
Spróbuj samodzielnie rozwiązać ten problem.

Wskazówka: Po przekroczeniu wartości granicznej (1023 b) należy wykorzystać dzielnik napięcia i przełączniki.

Przełącznik jest to element elektroniczny (taki cyfrowy wyłącznik) który pozwala na załączenie obwodu za pomocą sygnału, który podawany jest z wyjściowego pinu procesora. Na rysunku 3 przedstawiono przykład podłączenia przełącznika do procesora.

Jeżeli na pin A0 podamy komendę LOW przełącznik zostanie załączony i na porcie A7 można będzie dokonać pomiaru napięcia.

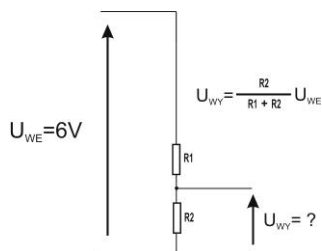
Jeżeli na pin A0 podamy komendę HIGH przełącznik zostanie wyłączony i tym samym odłączy port A7 od źródła napięcia regulowanego.



Rys. 3 Przykład podłączenia przełącznika do procesora

Wykorzystując powyższe wskazówki możesz przygotować wielozakresowy woltomierz.

Aby jednak zmierzyć napięcie na różnych zakresach należy wykorzystać dzielniki napięcia czyli dwa rezystory połączone tak jak na rysunku 4. Dzielnik napięcia jak sama nazwa wskazuje dokonuje podziału napięcia zgodnie ze wzorem (Rys. 4).



Rys. 4 Nieobciążony dzielnik napięcia.

Jeżeli wartości rezystorów $R_1 = R_2$ (będą identyczne) napięcie na wyjściu będzie wynosiło dokładnie połowę napięcia wejściowego zgodnie z rysunkiem Rys 4 będzie to 3V.

Konfiguracja rezystorów pozwala na dowolny podział napięcia wyjściowego.

Aby uzyskać wielozakresowy miernik napięcia należy użyć kilku dzielników dołączanych do portów za pomocą przełącznika.

Poniżej przykład doboru dzielników dla poszczególnych zakresów (na wyjściu dzielnika napięcie nie może przekraczać 5V):

- (0V -5V) bez dzielnika
- (0V- 10V, dzielnik dzieli 2

(0V - 20V dzielnik dzieli na 4
 (0V - 30V dzielnik dzieli na 6
 (0V - 40V dzielnik dzieli na 8
 (0V - 50V dzielnik dzieli na 10
 (0V - 60V dzielnik dzieli na 12

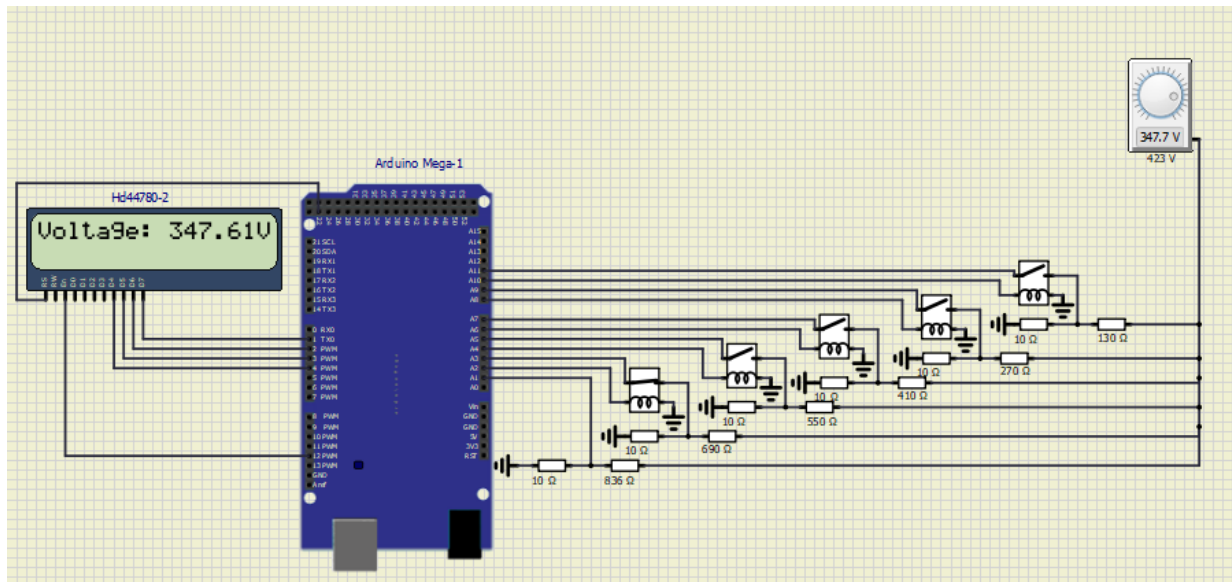
ITD.

0V - 100V dzielnik dzieli na 20
 0V - 123V Tutaj należy podzielić tak aby nie przekroczyć 5V czyli 25

Moje zakresy:

0-70V -> dzielnik dzieli na 14
 0-140V -> dzielnik dzieli na 28
 0-210V -> dzielnik dzieli na 42
 0-280V -> dzielnik dzieli na 56
 0-350V -> dzielnik dzieli na 70
 0-423V (żeby nie przekroczyć 5V to max = 420V) -> dzielnik dzieli na 84

Zrzut ekranu Schematu układu zaprojektowanego przez Ciebie.



Tutaj proszę wkleić kod programu.

```
#include <LiquidCrystal.h>

const int rs = 22, en = 12, d4 = 4, d5 = 3, d6 = 2, d7 = 1;
LiquidCrystal lcd(rs, en, d4, d5, d6, d7);

void setup() {
  lcd.begin(16, 2); // Inicjalizacja LCD z 16 kolumnami i 2 wierszami
  pinMode(A1, INPUT);
  pinMode(A3, INPUT);
  pinMode(A5, INPUT);
  pinMode(A7, INPUT);
  pinMode(A9, INPUT);
  pinMode(A11, INPUT);
  pinMode(A2, OUTPUT);
  pinMode(A4, OUTPUT);
  pinMode(A6, OUTPUT);
  pinMode(A8, OUTPUT);
  pinMode(A10, OUTPUT);
}

void loop() {
  // zakres 0-423V
  delay(100);
```

```

    int voltReader = analogRead(A1);
    float voltage = (voltReader / 1023.0) * 423;
    // zakres 0-350, czyli co 70V
    if (voltage < 350.0){
        digitalWrite(A2, HIGH);
        voltReader = analogRead(A3);
        voltage = (voltReader / 1023.0) * 350;
    } else {
        digitalWrite(A2, LOW);
    }
    // zakres 0-280V
    if (voltage < 280.0) {
        digitalWrite(A4, HIGH);
        voltReader = analogRead(A5);
        voltage = (voltReader / 1023.0) * 280;
    } else {
        digitalWrite(A4, LOW);
    }
    // zakres 0-210V
    if (voltage < 210.0) {
        digitalWrite(A6, HIGH);
        voltReader = analogRead(A7);
        voltage = (voltReader / 1023.0) * 210;
    } else {
        digitalWrite(A6, LOW);
    }
    // zakres 0-140V
    if (voltage < 140.0) {
        digitalWrite(A8, HIGH);
        voltReader = analogRead(A9);
        voltage = (voltReader / 1023.0) * 140;
    } else {
        digitalWrite(A8, LOW);
    }
    //zakres 0-70V
    if (voltage < 70.0) {
        digitalWrite(A10, HIGH);
        voltReader = analogRead(A11);
        voltage = (voltReader / 1023.0) * 70;
    } else {
        digitalWrite(A10, LOW);
    }

    displayVoltage(voltage);
}

void displayVoltage(float voltage) {
    lcd.clear();
    lcd.setCursor(0, 0);
    lcd.print("Voltage: ");
    lcd.print(voltage);
    lcd.print("V");
}

```

Zadanie 3.

Zaprojektuj układ pozwalający na pomiar prądu w zakresie od 0 do 2 Amperów (Cyfrowy Amperomierz). Wyniki pomiaru przedstaw na wyświetlaczu ciekłokrystalicznym. Jednostką natężenia prądu elektrycznego jest A (Amper).

Wskazówka.

Wykorzystaj prawo OHMA

$$I = \frac{U}{R}$$

Gdzie:

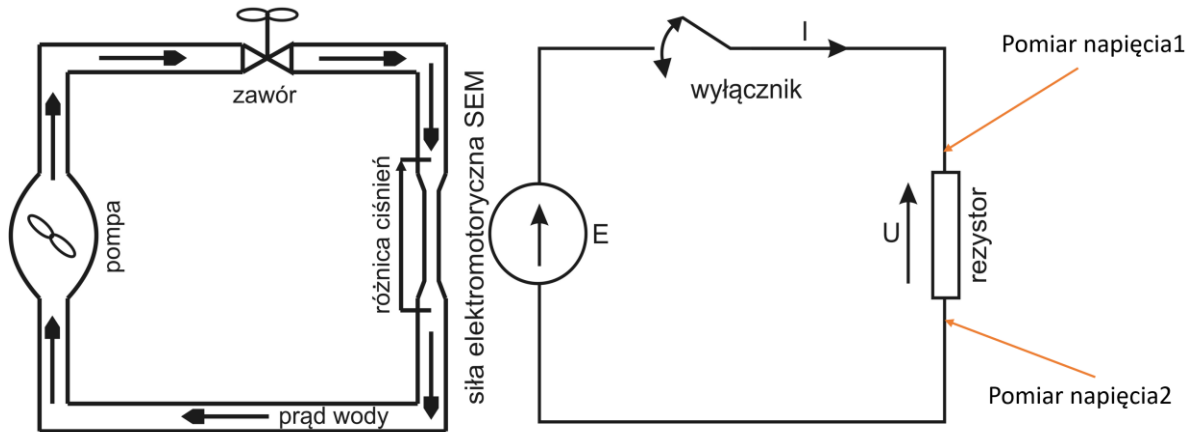
I – Natężenie prądu elektrycznego.

U -Napięcie.

R – Rezystancja (opór).

Wskazówka 2

Każdy rezystor powoduje spadek napięcia w obwodzie elektrycznym tak więc można go porównać do przewężenia w układzie hydraulicznym

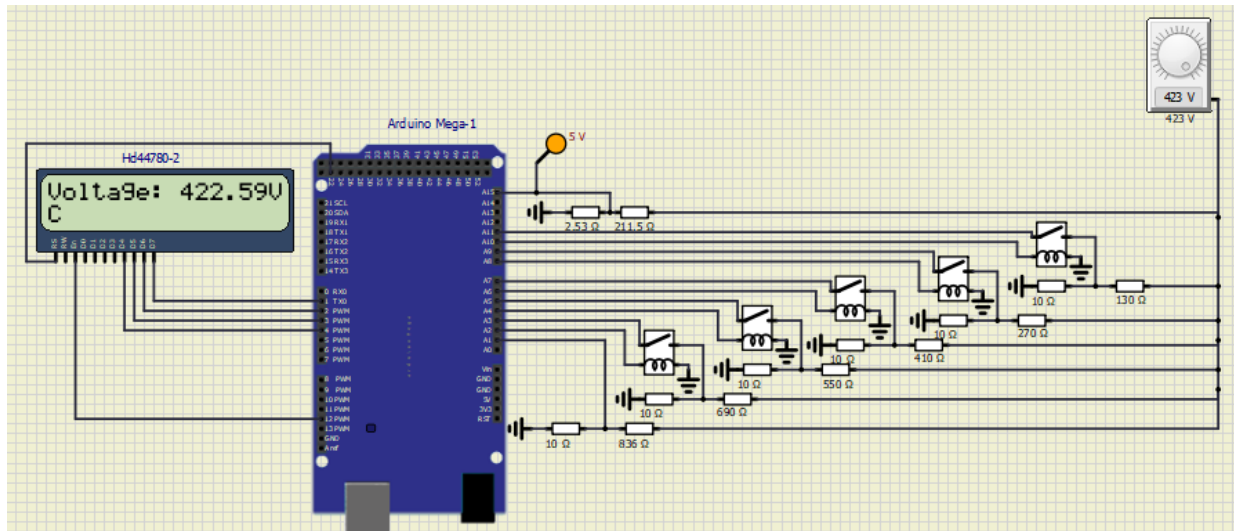


Rys. 5 Porównanie obwodu elektrycznego do układu hydraulicznego

Jeżeli zmierzysz napięcie w dwóch punktach, znasz wartość rezystancji i wykorzystasz prawo OHMA, otrzymasz natężenie prądu elektrycznego.

Jeżeli napięcie drugie zewrzesz do masy nie musisz go mierzyć, ponieważ wynosi 0 i wystarczy zmierzyć napięcie 1 aby otrzymać wartość prądu.

Zrzut ekranu Schematu układu zaprojektowanego przez Ciebie.



Gdzie max. 423 V / max. 2A = 211,5 Ω – pierwszy opornik

A drugi opornik $\approx 2,53 \Omega$

Ponieważ $423V / 5V = 84,6$, a stosunek wynosi $\frac{10}{10+836} = \frac{10}{846}$, to $\frac{R}{(211,5+R)} = \frac{10}{846} \Leftrightarrow R = 2.52990 \approx 2.53 \Omega$

Tutaj proszę wkleić kod programu.

```
#include <LiquidCrystal.h>
```

```
const int rs = 22, en = 12, d4 = 4, d5 = 3, d6 = 2, d7 = 1;
LiquidCrystal lcd(rs, en, d4, d5, d6, d7);
```

```
void setup() {
  lcd.begin(16, 2); // Inicjalizacja LCD z 16 kolumnami i 2 wierszami
  pinMode(A1, INPUT);
  pinMode(A3, INPUT);
  pinMode(A5, INPUT);
  pinMode(A7, INPUT);
  pinMode(A9, INPUT);
  pinMode(A11, INPUT);
  pinMode(A2, OUTPUT);
  pinMode(A4, OUTPUT);
  pinMode(A6, OUTPUT);
  pinMode(A8, OUTPUT);
  pinMode(A10, OUTPUT);
  pinMode(A15, INPUT);
}

void loop() {
  // zakres 0-423V
  delay(100);
  int voltReader = analogRead(A1);
  float voltage = (voltReader / 1023.0) * 423;
  // zakres 0-350, czyli co 70V
  if (voltage < 350.0){
    digitalWrite(A2, HIGH);
    voltReader = analogRead(A3);
    voltage = (voltReader / 1023.0) * 350;
  } else {
    digitalWrite(A2, LOW);
  }
  // zakres 0-280V
  if (voltage < 280.0) {
    digitalWrite(A4, HIGH);
    voltReader = analogRead(A5);
    voltage = (voltReader / 1023.0) * 280;
  } else {
    digitalWrite(A4, LOW);
  }
  // zakres 0-210V
  if (voltage < 210.0) {
    digitalWrite(A6, HIGH);
    voltReader = analogRead(A7);
    voltage = (voltReader / 1023.0) * 210;
  } else {
    digitalWrite(A6, LOW);
  }
  // zakres 0-140V
  if (voltage < 140.0) {
    digitalWrite(A8, HIGH);
    voltReader = analogRead(A9);
    voltage = (voltReader / 1023.0) * 140;
  } else {
    digitalWrite(A8, LOW);
  }
  //zakres 0-70V
  if (voltage < 70.0) {
    digitalWrite(A10, HIGH);
    voltReader = analogRead(A11);
    voltage = (voltReader / 1023.0) * 70;
  } else {
    digitalWrite(A10, LOW);
  }
}
```

```
float current = 0.0;
float voltReader2 = analogRead(A15);
float voltage2 = (voltReader2 / 1023.0) * 423;
current = voltage2 / 211.5;

displayVoltage(voltage, current);
}

void displayVoltage(float voltage, float current) {
  lcd.clear();
  lcd.setCursor(0, 0);
  lcd.print("Voltage: ");
  lcd.print(voltage);
  lcd.print("V");
  lcd.setCursor(0, 1);
  lcd.print("Current: ");
  lcd.print(current);
  lcd.print("A");
}
```

Wnioski:

We wnioskach proszę się zastanowić czy układy przedstawione na symulacjach działałyby w rzeczywistości.

Należy pamiętać, że są to układy idealne i w rzeczywistości aby przedstawione schematy funkcjonowały należałoby dodać jeszcze kilka elementów elektronicznych.

Wskazówka: Czy cewkę przekaźnika można podłączyć bezpośrednio do portu? Jeżeli potrafisz podłączyć przekaźnik do portu procesora tak aby go nie uszkodzić i zamieścisz poniżej prawidłowy schemat podłączenia zyskasz dodatkowe 10 pkt. W szczególnym przypadku w tym zadaniu można uzyskać 110 pkt. (bezbłędnie wykonane zadanie oddane w pierwszej kolejności 100 pkt i prawidłowe podłączenie przekaźnika to 10 pkt = 110pkt).

Według mnie świat nie jest tak prosty jak komputerowa symulacja, więc w rzeczywistości zapewne dany schemat nie działałby w rzeczywistości, ze względu na możliwe wystąpienie wysokiego napięcia, które mogłoby zepsuć działanie płytki mikroprocesora. Woltomierz oraz amperomierz działa prawidłowo. Jeśli chodzi o woltomierz, występuje 6 zakresów: 0-70V, 0-140V, 0-210V, 0-280V, 0-350V oraz 0-420V.