
		Instytut Informatyki Politechniki Śląskiej Zespół Mikroinformatyki i Teorii Automatów Cyfrowych			
Rok akademicki	Rodzaj studiów*: SSI/NSI/NSM	Przedmiot: (Języki Asemblerowe/SMIW)		Grupa	Sekcja
2023/2024	NSI	SMIW		2	1
Prowadzący przedmiot:	Dr inż. Gabriel Drabik			Termin: (dzień tygodnia, godzina)	
Imię:	Adam			Sobota	
Nazwisko:	Kuraczyński			11:45	
Email:	ak301824@student.polsl.pl				
<i>Raport końcowy</i>					
Temat projektu:					
Sterowanie procesem spalania w piecu węglowym					

Spis treści:

1. Temat projektu, opis założeń, opis funkcji urządzenia
2. Analiza zadania, lista potrzebnych elementów do zbudowania projektu, kosztorys
3. Specyfikacja wewnętrzna urządzenia
 - 1) Schemat blokowy i ideowy urządzenia
 - 2) Opis funkcji poszczególnych bloków układu
 - 3) Schemat montażowy obejmujący projekt płytki drukowanej, rozmieszczenie elementów na płycie
 - 4) Algorytm oprogramowania urządzenia (schemat blokowy)
 - 5) Opis wszystkich zmiennych
 - 6) Opis funkcji procedur
 - 7) Opis integracji oprogramowania z układem elektronicznym
 - 8) Szczegółowy opis działania ważniejszych procedur
4. Specyfikacja zewnętrzna urządzenia
 - 1) Opis funkcji elementów sterujących urządzeniem
 - 2) Opis funkcji elementów wykonawczych
 - 3) Opis reakcji oprogramowania na zdarzenia zewnętrzne
 - 4) Skrócona instrukcja obsługi urządzenia
5. Opis montażu układu, sposobu uruchamiania oraz testowania
 - 1) Problemy, które wystąpiły w trakcie montażu, uruchomienia i ich rozwiązania
 - 2) Testy poprawności działania urządzenia
6. Wnioski końcowe

1. Temat projektu, opis założeń, opis funkcji urządzenia

Tematem projektu jest stworzenie układu zarządzającego sterowaniem piecykiem węglowym starego typu. Taki układ po podłączeniu do prądu powinien mieć możliwość zmiany wartości zadanych w trakcie działania układu oraz wyświetlać aktualnie zadane parametry. Na podstawie zadanych parametrów układ powinien sterować pompą i dmuchawą pieca.

2. Analiza zadania, lista potrzebnych elementów do zbudowania projektu, kosztorys

Wymagane elementy do zbudowania projektu to następująco:

- Zasilanie
- Mikrokontroler
- Wyświetlacz LCD
- Klawiatura
- Czujnik temperatury

Analiza i wybór elementów:

- Mikrokontroler – zdecydowałem się na wybór Arudino Uno R3, gdyż korzystałem już tego urządzenia. Arduino obsługuje prace pod napięciem 5V i pozwala na zasilanie elementów napięciem 3.3V. Obsługuje wymagane I²C, jest kompatybilne z dużą ilością sprzętu i posiada wiele bibliotek.



- Wyświetlacz LCD – zdecydowałem się na 84x48 nokia 5110 3310 (pcd8544). Wyświetlacz jest kompatybilny z Arduino Uno, jest stosunkowo tanim rozwiązaniem, pozwala na wyświetlanie informacji w wielu wierszach co pozwala na wyświetlenie intuicyjnych ekranów. Z powodu prostoty rozwiązania nie zdecydowałem się na wyświetlacz 2x16 HD44780 – nie wprowadza on niczego do projektu. Pracuje pod napięciem 3.3 V co pozwala na podłączenie do Arduino.



- Klawiatura – zdecydowałem się na klawiaturę membranową 4x4 z powodu prostej obsługi z Arduino, dużej ilości przycisków, które będą potrzebne do potencjalnych dodatkowych funkcjonalności systemu, prostoty działania i niskiego kosztu.



- Czujnik temperatury – zdecydowałem się na model DS18B20 z powodu bibliotek dostępnych do Arduino, dodatkowego zabezpieczenia wodoodpornego, długiego kabla pozwalającego odseparowanie czujnika od reszty elementów, które z pewnością nie chcemy blisko wysokich temperatur.



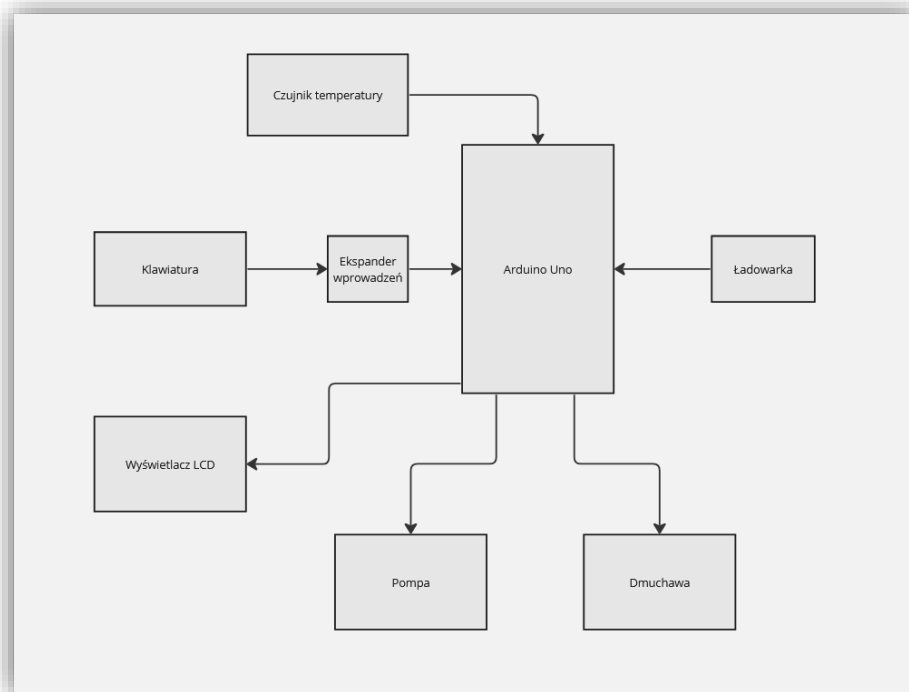
- Zasilanie układu – zdecydowałem się na zasilanie za pomocą kabla USB typ B z powodu prostej dostępności, możliwości wpięcia do komputera jak i gniazdka używając adaptera. Zasilanie dostarcza wystarczające napięcie do zasilenia Arduino. Jest to również tańsze rozwiązanie w porównaniu do osobnego zasilacza.

Kosztorys:

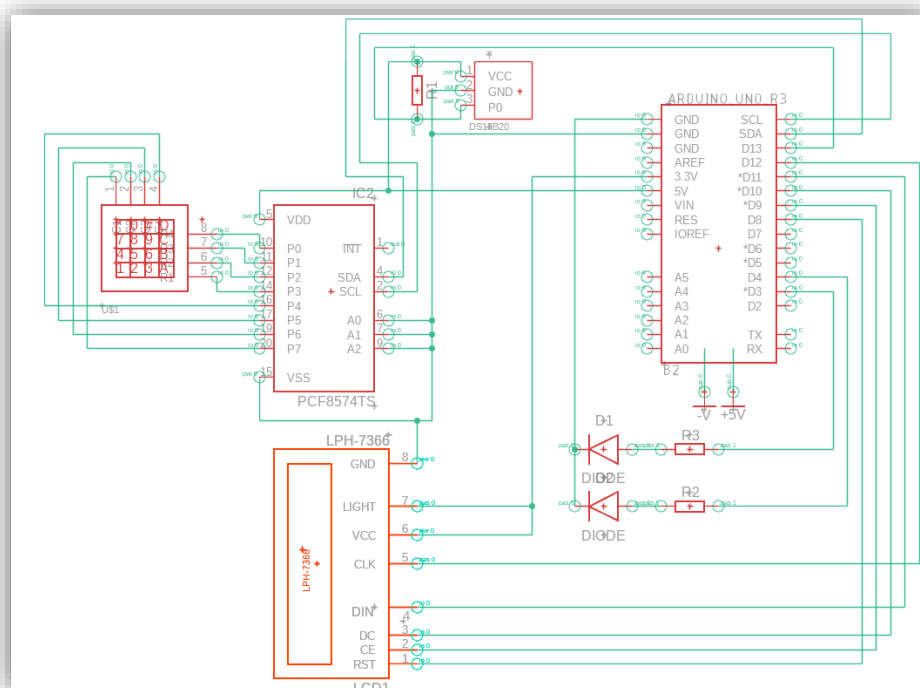
Rzecz	Sztuk	Koszt
Arduino UNO R3	1	120
Płytki stykowa 30	1	7
Kabel USB B	1	20
Klawiatura membranowa samoprzylepna 4x4 - 16 klawiszy	1	5
Czujnik temperatury DS18B20	1	18
Dioda LED	2	3
Rezystor 330 Om	2	2
Rezystor 4,7k Om	1	2
Ekspander wprowadzeń Pcf8574	1	15
84 48 Nokia 5110 3310 wyświetlacz (pcd8544)	1	25
Przewody połączeniowe męsko-męskie	40	20
	Suma:	237

3. Specyfikacja wewnętrzna urządzenia

3.1. Schemat blokowy i ideowy urządzenia



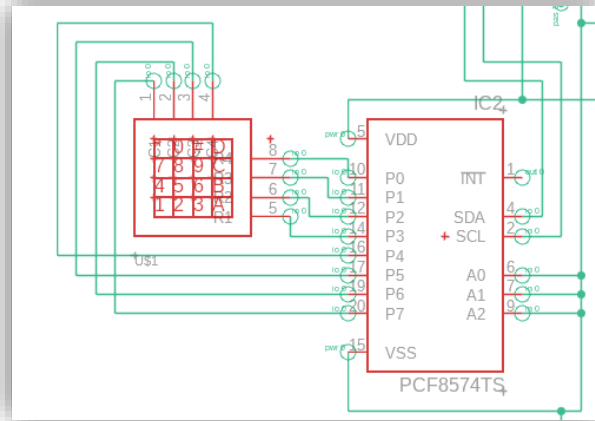
Rysunek 1 Schemat blokowy



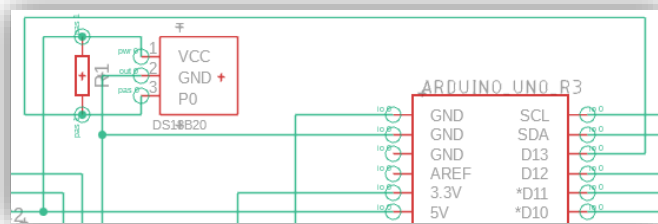
Rysunek 2 Schemat ideowy

3.2. Opis funkcji poszczególnych bloków układu

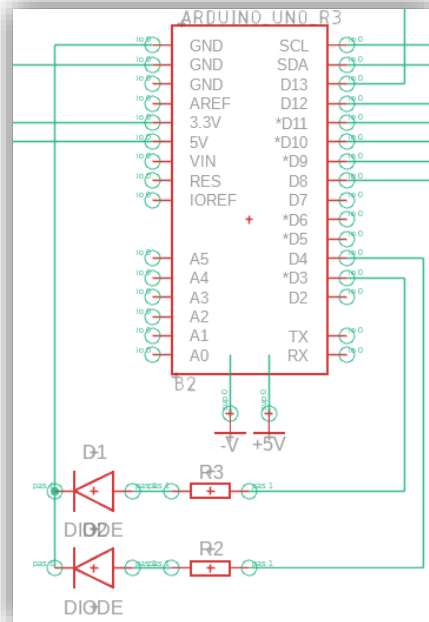
Podłączenie klawiatury do ekspandera wprowadzeń – służy do interakcji użytkownika z systemem. Ekspander jest interfejsem w komunikacji klawiatura <-> Arduino.



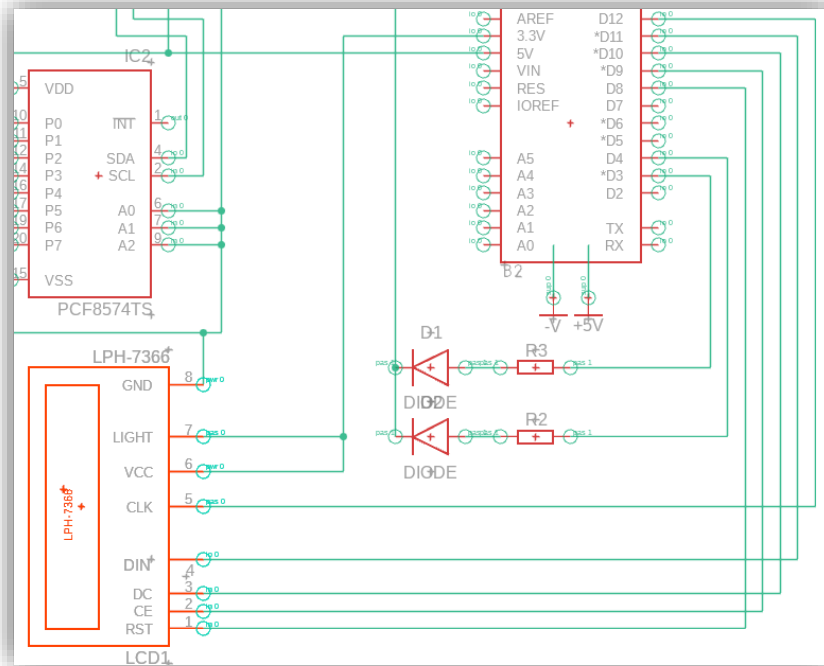
Podłączenie czujnika temperatury z mikrokontrolerem – służy do stałego odczytu temperatury. Używa protokołu OneWire.



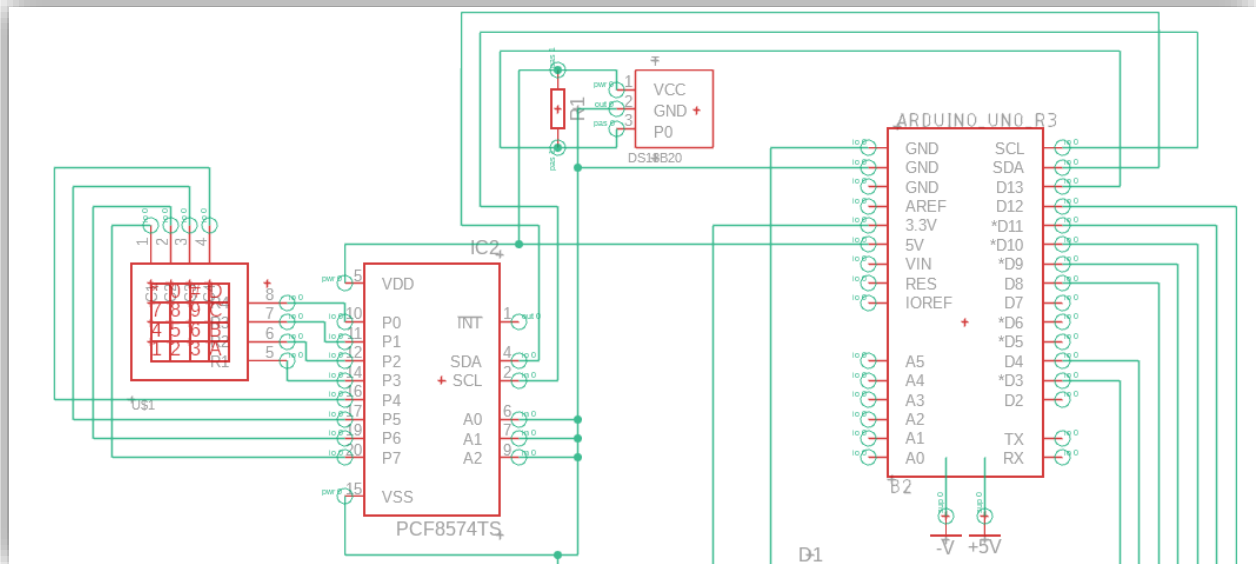
Podłączenie diod do mikrokontrolera – diody symbolizują pompę i dmuchawę, których włączaniem/wyłączaniem steruje Arduino.



Podłączenie wyświetlacza – Arduino wysyła dane w celu wyświetlenia ich użytkownikowi. Wyświetlacz używa protokołu SPI.

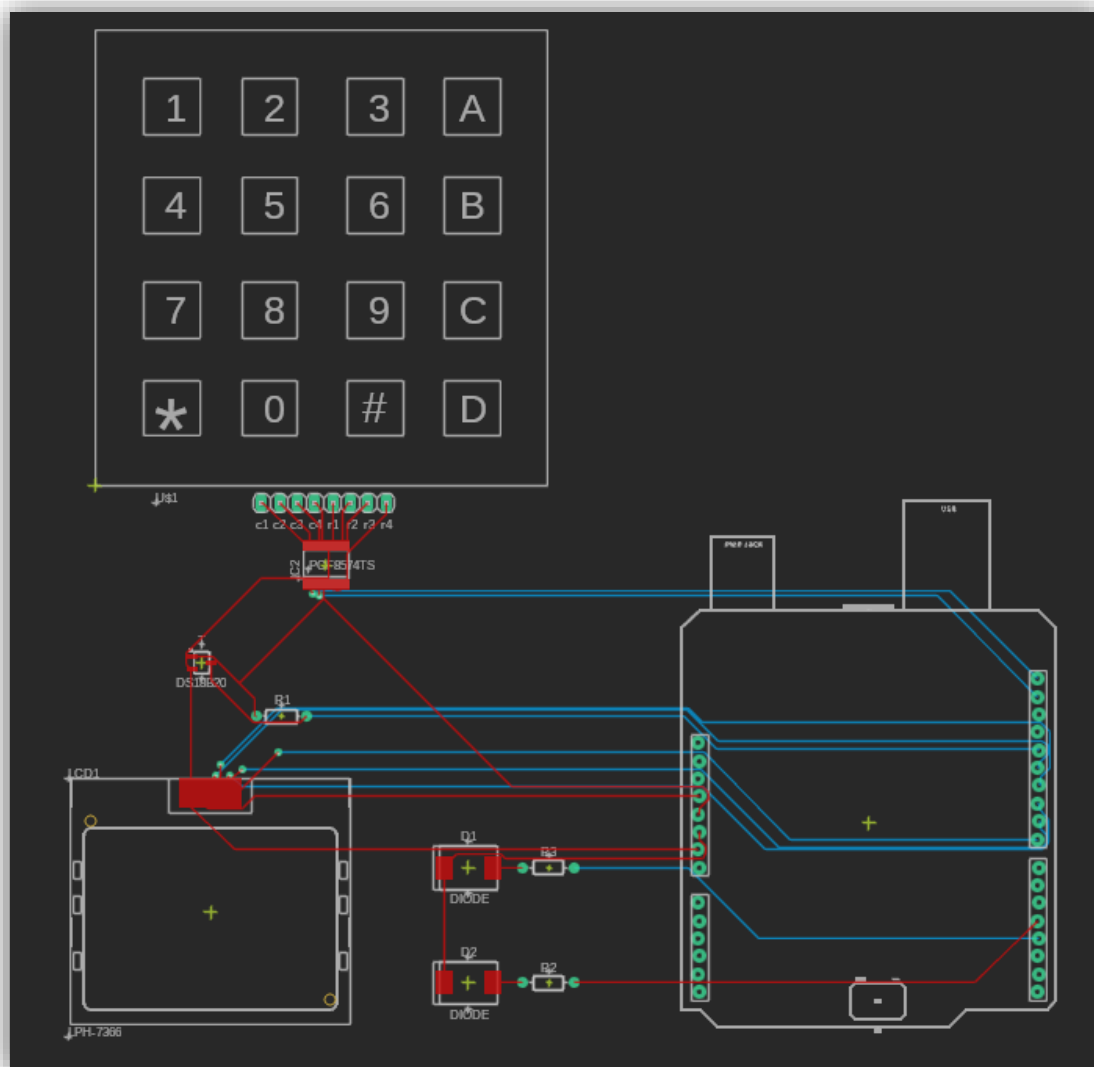


Podłączenie ekspandera wprowadzeń do mikrokontrolera – ekspander służy jako interfejs między klawiaturą a Arduino, dzięki niemu jest możliwość podpięcia większej ilości pinów cyfrowych. Komunikacja zachodzi przez magistrale I²C.



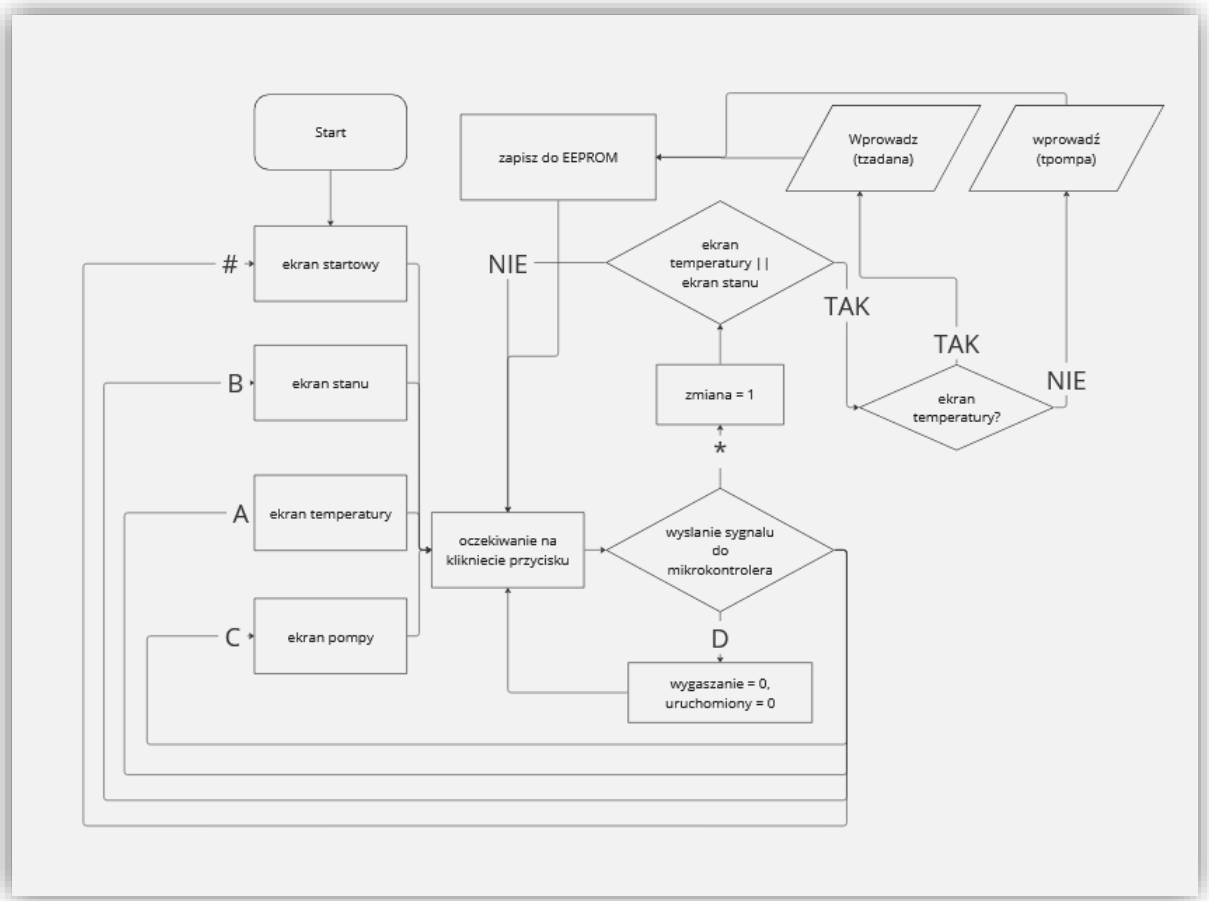
3.3. Schemat montażowy obejmujący projekt płytki drukowanej, rozmieszczenie elementów na płycie

B2	ARDUINO_UNO_R3	ARDUINO_UNO_R3_SHIELDUNO_R3_SHIELD_NO LABELS_LOCK	UNO_R3_SHIELD_NO LABELS_LOCK	Arduino R3 Shield Footprint			
D1	DIODE	DIODEDO-214AB(SMC)	DIOM7959X625N	Diode Rectifier - Generic	Diode	Rectifier	DIODE
D2	DIODE	DIODEDO-214AB(SMC)	DIOM7959X625N	Diode Rectifier - Generic	Diode	Rectifier	DIODE
IC2	PCF8574TS	PCF8574TS	SSOP20	I2C-Bus Remote 8-bit I/O expander			
LCD1	LPH-7366	LPH-7366	LPH-7366				
R1		R_AXIAL-7.2MM-PITCH	RESAD724W46L381D178B	Resistor Fixed - Generic	Resistor	Fixed	
R2		R_AXIAL-7.2MM-PITCH	RESAD724W46L381D178B	Resistor Fixed - Generic	Resistor	Fixed	
R3		R_AXIAL-7.2MM-PITCH	RESAD724W46L381D178B	Resistor Fixed - Gen	Resistor	Fixed	
T	DS18B20	A1324LLHLT-TLH	SOT95P285X113-3N	SENSOR HALL EFFECT ANALOG SOT23W	Sensor	Magnetic	A1324LLHLT
U\$1		KEYPAD4X4	KEYPAD4X4				



Rysunek 3 Projekt Płytki

3.4. Algorytm oprogramowania urządzenia (schemat blokowy)



Rysunek 4 Algorytm blokowy

3.5. Opis wszystkich zmiennych

```

#define ONE_WIRE_BUS 13
OneWire oneWire(ONE_WIRE_BUS);
DallasTemperature sensors(&oneWire);
Adafruit_PCD8544 display =
  Adafruit_PCD8544(12, 11, 10, 9, 8);
const byte pompa = 3;
const byte dmuchawa = 4;
int tżadana = 29;
int tpompa = 27;
int temperatura;
  
```

- zmienna z adresem czujnika temperatury
- zmienna tworząca komunikację OneWire
- zmienna tworząca komunikację DallasTemperature używając OneWire
- zmienna tworząca komunikację z wyświetlaczem
- zmienna z adresem pompy
- zmienna z adresem dmuchawy
- zmienna z wartością, do której działa dmuchawa
- zmienna z wartością temperatury, od której włączana jest pompa
- zmienna z wartością aktualnej temperatury

<pre> byte ekran = 0; byte zmiana = 0; byte uruchomiony=0; byte wygaszanie=0; const byte ROWS = 4; const byte COLS = 4; Char keys[ROWS][COLS] = { {'1','2','3','A'}, {'4','5','6','B'}, {'7','8','9','C'}, {'*','0','#','D'}}; byte rowPins[ROWS] = {0, 1, 2, 3}; byte colPins[COLS] = {4, 5, 6, 7}; int i2caddress = 0x20; Keypad_I2C kpd = Keypad_I2C(makeKeymap(keys), rowPins, colPins, ROWS, COLS, i2caddress); temperatura=sensors.getTempCByIndex(0); char klawisz = kpd.getKey(); </pre>	<ul style="list-style-type: none"> - zmienna pomocnicza do przejść między ekranami - zmienna sygnalizująca wejście do Trybu Edycji - zmienna sygnalizująca uruchomienie Pompy - zmienna sygnalizująca skończenie 1 cyklu działania układu - zmienna pomocnicza z ilością wierszy - zmienna pomocnicza z ilością kolumn - zmienna pomocnicza mapująca wartości klawiszy - zmienna pomocnicza z adresami wierszy klawiatury - zmienna pomocnicza z adresami kolumn klawiatury - zmienna pomocnicza z adresem i2c ekspandera wprowadzeń - zmienna tworząca komunikację klawiatury - zmienna do przechowywania temperatury odebranej przez czujnik - zmienna do przechowywania ostatniego wciśniętego przycisku
--	--

3.6. Opis funkcji procedur

<pre> int odczytLiczba() int odczytCyfra() void EkranStartowy() void EkranTemperatury () void EkranStanu() void EkranPompa() void setup() void loop() switch (klawisz) switch (ekran) display.clearDisplay(); display.setTextSize(1); display.setTextColor(BLACK); display.setCursor(0, 0); </pre>	<ul style="list-style-type: none"> - funkcja, która zwraca zmienioną wartość temperatury - funkcja, która odczytuje cyfry z klawiatury w celu zmiany temperatury zadanej - funkcja inicjująca ekran startowy - funkcja inicjująca ekran temperatury - funkcja inicjująca ekran stanu - funkcja inicjująca ekran pompy - funkcja ustawiająca interfejsy do obsługi - główna funkcja mikrokontrolera, zarządza zmianami ekranów i zachowaniem pompy i dmuchawy - przełącznik obsługujący zmianę ekranów - przełącznik obsługujący zmianę ekranów - czyści ekran - zmienia wielkość czcionki na ekranie - zmienia kolor czcionki na ekranie - ustawia pozycję kursora
---	--

<code>display.print();</code>	- drukuje tekst do ekranu
<code>display.println();</code>	- przechodzi do nowej linii na ekranie
<code>display.display();</code>	- wyświetla cały kontekst na ekranie
<code>display.begin();</code>	- inicjalizuje ekran
<code>display.setContrast(60);</code>	- zmienia kontrast ekranu
<code>Serial.begin(9600);</code>	- rozpoczyna komunikację seryjną
<code>sensors.begin();</code>	- inicjalizuje czujniki temperatury
<code>Wire.begin();</code>	- inicjalizuje komunikację I2C
<code>kpd.begin();</code>	- inicjalizuje klawiaturę
<code>pinMode(pompa, OUTPUT);</code>	- konfiguruje pin jako wyjście
<code>pinMode(dmuchawa, OUTPUT);</code>	- konfiguruje pin jako wyjście
<code>EEPROM.put(100, tzadana);</code>	- wpisuje wartość tzadana do EEPROM na adresie 100
<code>EEPROM.get(100, tzadana);</code>	- odczytuje wartość tzadana z EEPROM z adresu 100
<code>sensors.requestTemperatures();</code>	- inicjalizuje pomiary temperatury z sensora temperatury
<code>sensors.getTempCByIndex(i);</code>	- odczytuje wartość temperatury z sensora
<code>digitalWrite(dmuchawa,HIGH);</code>	- ustawia pin dmuchawa jako 1
<code>digitalWrite(dmuchawa,LOW);</code>	- ustawia pin dmuchawa jako 0
<code>digitalWrite(pompa,HIGH);</code>	- ustawia pin pompa jako 1
<code>digitalWrite(pompa,LOW);</code>	- ustawia pin pompa jako 0

3.7. Opis integracji oprogramowania z układem elektronicznym

Oprogramowanie wykorzystuje następujące biblioteki:

- **Adafruit_GFX** oraz **Adafruit_PCD8544** – wykorzystane w celu zapewnienia komunikacji z wyświetlaczem NOKIA typu PCD8544,
- **Keypad_I2C** - wykorzystana do zapewnienia komunikacji z klawiaturą, użyta została Keypad_I2C a nie Keypad jako że komunikacja zachodzi poprzez ekspander wprowadzeń I2C,
- **Wire, OneWire, DallasTemperature** - wykorzystane, aby zapewnić komunikację z czujnikiem temperatury,
- **EEPROM** - wykorzystana, aby wartości zadane dla pompy oraz dmuchawy były zachowane między włączeniami systemu.

3.8. Szczegółowy opis działania ważniejszych procedur

Gdy aktualnie odczytana temperatura jest wyższa niż zadana dla systemu oraz jego tryb to wygaszanie to dmuchawa jest wyłączona. Jeśli nie to dmuchawa działa, jest to zabezpieczenie przed działaniem dmuchawy cały czas

```
if ((temperatura>=tzadana)||(wygaszanie==1)) digitalWrite(dmuchawa,LOW);  
else digitalWrite(dmuchawa,HIGH);
```

Gdy aktualnie odczytana temperatura jest wyższa od temperatury zadanej dla pompy to pompa się uruchamia, a system przechodzi w tryb uruchomiony tj. grzeje dom.

```
if(temperatura>tpompa) {  
    digitalWrite(pompa,HIGH);  
    uruchomiony=1;  
} else digitalWrite(pompa,LOW);
```

Gdy cykl został uruchomiony tj. pompa choć raz się włączyła oraz aktualny stan pompy to 0, czyli wyłączona to rozpoczyna się proces wygaszania mający na celu zabezpieczenie systemu przed działaniem dmuchawy cały czas.

```
if((uruchomiony==1)&&(digitalRead(pompa)==0)) wygaszanie=1;
```

Zawsze pierwsza liczba wprowadzona w Trybie edycji to liczba dziesiątek a druga liczba jednostki.

```
int odczytLiczba()  
{  
    return odczytCyfra() * 10 + odczytCyfra();  
}
```

4. Specyfikacja zewnętrzna urządzenia

4.1. Opis funkcji elementów sterujących urządzeniem

Sterowanie używając klawiatury:

- Przycisk „A” – przejście do Ekranu Temperatury
- Przycisk „B” – przejście do Ekranu Stanu
- Przycisk „C” – przejście do Ekranu Pompy
- Przycisk „#” – przejście do Ekranu Startowego
- Przycisk „D” – zresetowanie zmiennych i ponowne uruchomienie cyklu
- Przycisk „*” – przejście do Trybu Edycji temperatury na Ekranie Temperatury i Ekranie Pompy
- Przyciski 0-9 – w Trybie Edycji pozwalają na wybranie temperatury

Układ jest wyposażony w przycisk restartu pozwalający na restart mikrokontrolera.

4.2. Opis funkcji elementów wykonawczych

84 48 Nokia 5110 3310 Wyświetlacz LED (pcd8544) – wyświetla informacje o temperaturze i stanach pompy i dmuchawy oraz zadanych wartościach

Dioda czerwona – symbolizuje uruchomienie pompy pieca węglowego

Dioda żółta – symbolizuje uruchomienie dmuchawy pieca węglowego

Czujnik temperatury DS18B20 – odczytuje temperaturę

4.3. Opis reakcji oprogramowania na zdarzenia zewnętrzne

Wciśnięcie przycisku „A” – przejście do Ekranu Temperatury

Wciśnięcie przycisku „B” – przejście do Ekranu Stanu

Wciśnięcie przycisku „C” – przejście do Ekranu Pompy

Wciśnięcie przycisku „#” – przejście do Ekranu Startowego

Wciśnięcie przycisku „D” – zresetowanie zmiennych i ponowne uruchomienie cyklu

Wciśnięcie przycisku „*” – przejście do Trybu Edycji temperatury na Ekranie Temperatury i Ekranie Pompy

Wciśnięcie przycisków 0-9 – w Trybie Edycji pozwalają na wybranie temperatury

Wciśnięcie przycisku restartu na mikrokontrolerze – resetuje mikrokontroler a ekran wraca do Ekranu Startowego

Podłączenie i odłączenie źródła zasilania – uruchamia i wyłącza urządzenie

Czujnik temperatury – ogrzanie lub oziębienie czujnika zmienia działanie całego układu - włącza bądź wyłącza pompę i dmuchawę

4.4. Skrócona instrukcja obsługi urządzenia

Po podłączeniu urządzenia do prądu na wyświetlaczu pojawia się ekran startowy. Informuje nas o tym, że klikając na klawiaturze „A” przenosimy się do Ekranu Temperatury, klikając „B” do Ekranu Stanu a klikając „C” do Ekranu Pompy. Klikając „#” wracamy do ekranu startowego. Na ekranie temperatury otrzymujemy informacje o temperaturze pieca tj. aktualnej temperaturze zmierzonej przez czujnik temperatury oraz temperaturze zadanej to jest temperatury, którą chcemy osiągnąć – temperatura, do której ma być uruchomiona dmuchawa. Ekran Stanu informuje nas o aktualnym stanie dmuchawy i pompy, czy są włączone czy nie.

Ekran Pompy informuje nas o temperaturze włączenia pompy. Klikając „*” na klawiaturze na Ekranie Temperatury jesteśmy w stanie zmienić zadana temperaturę dmuchawy, a na ekranie Pompy klikając „*” na klawiaturze jesteśmy w stanie zmienić temperaturę zadana pompy. Po zadaniu tych temperatur w standardowym przypadku dmuchawa powinna się włączyć, po tym pompa powinna się włączyć, następnie wyłącza się dmuchawa, a po tym pompa – dochodzi do resetu systemu. Po tym wystarczy, że na klawiaturze wciśniemy „D” aby zresetować system – dmuchawa znowu zacznie działać. To zabezpiecza system przed działaniem cały czas – nawet jak tego nie chcemy.

5. Opis montażu układu, sposobu uruchamiania oraz testowania

Układ został podłączony używając płytki prototypowej. Na płytce umieszczone zostały diody, rezystory, wyświetlacz, czujnik temperatury oraz ekspander wprowadzeń. Następnie używając przewodów połączeniowych wszystko zostało ze sobą połączone i osobno przetestowane w kolejności czujnik temperatury, diody i rezystory, wyświetlacz, klawiatura i ekspander wprowadzeń. Po podłączeniu wszystkiego wykonane zostało oprogramowanie, które testuje każde z urządzeń.

5.1. Problemy, które wystąpiły w trakcie montażu, uruchomienia i ich rozwiązania

W trakcie testowania urządzenia z przyczyn mi nieznanych wyświetlacz LCD przestał działać, zmuszony byłem do zakupu nowego egzemplarza.

5.2. Testy poprawności działania urządzenia

Testy opierały się na sprawdzeniu pełnej funkcjonalności urządzenia: działanie klawiatury, wyświetlacza, bieżący pomiar temperatury, jak reagują pompa i dmuchawa na odmierzone temperatury oraz czy układ działa po podłączeniu do zasilania. Wszystkie testy przeszły pomyślnie.

6. Wnioski końcowe

Układ sprawdza się, jeśli chodzi o podstawowe wymogi. Można jednak go rozbudować o dodatkowe funkcjonalności:

- Zamknąć go w obudowie, aby zabezpieczyć elementy,
- Odczytanie temperatury mogłoby zostać wykonywane w części mieszkalnej budynku a informacja o temperaturze przekazywana do głównego kontrolera,
- Można usprawnić wyświetlanie komunikatów na wyświetlaczu, aby wszystko odbywało się na jednej stronie bez wymogu zmiany ekranów wyświetlania.