

UNIwersytet WarMińsko-Mazurski w Olsztynie  
Wydział Matematyki i Informatyki

**Kierunek: Informatyka**

**Michał Bartosz Ludwikowski**

**Opracowanie automatycznego systemu  
chłodzącego stacji roboczej z wykorzystaniem  
Arduino**

Praca inżynierska wykonana  
w katedrze Matematycznych Metod Informatyki  
pod kierunkiem  
dra Krzysztofa Sopyły

Olsztyn, 2016 rok

UNIVERSITY OF WARMIA AND MAZURY IN OLSZTYN  
FACULTY OF MATHEMATICS AND COMPUTER SCIENCE

**Field of Study: Computer Science**

**Michał Bartosz Ludwikowski**

# **Developing automatic cooling system for workstation using Arduino**

Engineer's Thesis is performed  
in the Department of Mathematical Methods of Informatics  
under supervision of  
dr Krzysztof Sopyła

Olsztyn, 2016

# Spis treści

<b>Streszczenie</b>	3
<b>Abstract</b>	4
<b>Rozdział 1. Wstęp</b>	5
1.1. Wprowadzenie	5
1.2. Cel pracy	5
1.3. Układ pracy	6
<b>Rozdział 2. Opis technologii wykorzystanych podczas realizacji projektu</b>	7
2.1. Arduino	7
2.1.1. Język programowania Arduino	7
2.1.1.1. Wiring	8
2.1.2. Arduino Software (IDE)	8
2.1.2.1. Processing	8
2.2. Narzędzia do projektowania elektroniki	9
2.2.1. Eagle	9
2.2.2. Fritzing	9
2.3. CAD	10
2.3.1. AutoCAD	10
2.4. Narzędzia do testów stabilności i benchmark	11
2.4.1. OCCT	11
2.4.2. Catzilla	11
<b>Rozdział 3. Realizacja projektu systemu chłodzącego</b>	12
3.1. Elektroniczny układ współpracujący z Arduino	12
3.1.1. Projekt układu elektronicznego	12
3.1.2. Wykonanie układu elektronicznego	13
3.2. Implementacja sterownika układu chłodzącego w środowisku Wiring	15
3.2.1. Projekt algorytmu głównego	15
3.2.2. Implementacja algorytmu	15
3.3. System chłodzący	15
3.3.1. Projekt systemu chłodzącego	15
3.3.2. Wykonanie systemu chłodzącego	15
<b>Rozdział 4. Testy wydajnościowe, optymalizacja osiągnięć stacji roboczej oraz porównanie wyników z danymi wyjściowymi</b>	17
4.1. Opis zagadnienia overclockingu	17
4.2. Opis parametrów przed optymalizacją	17
4.3. Opis optymalizacji	17
4.4. Testy wydajnościowe	17
4.5. Analiza wyników optymalizacji	17
4.6. Analiza wyników testów wydajnościowych	17

<b>Rozdział 5. Podsumowanie</b> . . . . .	18
<b>Spis rysunków</b> . . . . .	19
<b>Bibliografia</b> . . . . .	20

# Streszczenie

Doskonale wiadomo iż elektronika, która otacza nas podczas pracy nagrzewa się. Jest to oczywiście zjawisko niekorzystne i nieporządane w większości przypadków, jeżeli nie we wszystkich. Komputery stacjonarne chłodzone są najczęściej powietrzem. Wykorzystuje się do tego radioatory i wentylatory. Ostatnimi czasy pojawiły się rozwiązania wykorzystujące ciecz do transportowania ciepła, te rozwiązania odprowadzają ciepło tylko z punktów, do których przyłączone są elementy układu chłodzącego.

Poniższa praca przedstawia zaprojektowanie oraz wykonanie automatycznego systemu chłodzącego, który odprowadza ciepło z całej powierzchni elektroniki poprzez zanurzenie jej w cieczy, która nie przewodzi prądu.

Rozwiązanie to pozwala na skuteczne odprowadzanie ciepła z podzespołów komputera, oraz rozwiązuje problem utrzymywania porządku wewnątrz obudowy naszego komputera - problem przestaje istnieć.

# Abstract

It is well known that the electronics that surrounds us at work and at home are overheating, getting warmer and warmer. This is obviously disadvantageous and undesirable phenomenon in most cases. Desktop Computers are usually cooled by air using these radiators and fans. There is a solution to our issue which is based on the fluid which transports heat. This solution is only conducting the heat from the points where cooling system components are mounted.

This paper presents the design and creation of automatic cooling system that draws heat away from the electronic parts of our Desktop, which entire surface is immersed in a liquid that does not conduct electricity.

This allows for efficient heat dissipation from computer components, and solves the problem of keeping computer parts dirt-free since they are submerged in a fluid.

## Rozdział 1

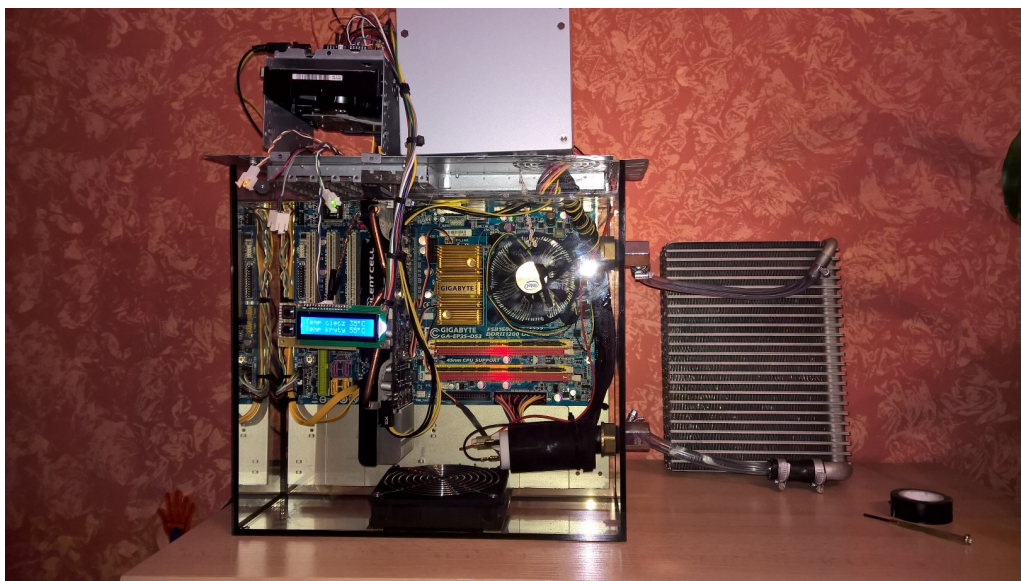
# Wstęp

### 1.1. Wprowadzenie

Chłodzenie podzespołów komputera to bardzo istotny aspekt poprawnej pracy stacji roboczej. W dzisiejszych czasach najpopularniejszym sposobem chłodzenia są zaawansowane radiatory chłodzone powietrzem aktywnie (zastosowanie wentylatora) lub pasywnie (brak wentylatora). Coraz popularniejsze stają się bloki wodne, które zastępując radiatory umożliwiają wymianę ciepła podzespołu z cieczą. Sposób ten pozwala jedynie na przekazanie ciepła z ograniczonych powierzchni elementów z których zbudowany są podzespoły komputera. Rozwiązaniem, które pozwoli zwiększyć te powierzchnie jest zanurzenie całego komputera w chłodziwie.

### 1.2. Cel pracy

Niniejsza praca ma na celu przedstawienie procesu zaprojektowania oraz wykonania automatycznego systemu chłodzącego stację roboczą. W pracy poza opisami narzędzi takich jak Arduino wykorzystanych do realizacji projektu znajdziemy bogato ilustrowane opisy poszczególnych kroków dążących do stworzenia skutecznego systemu chłodzącego. Skuteczność systemu omówiona i podsumowana zostanie na wykresach temperatur podzespołów zebranych podczas wzmożonej pracy stacji roboczej.



Rysunek 1.1. Gotowy projekt

### 1.3. Układ pracy

W drugim rozdziale przedstawiony został proces projektowania i wykonania układu elektronicznego współpracującego z Arduino. Zawiera on schematy elektryczne układu pomiaru temperatury, układu sterowania pompy, układu wyświetlania danych.

Rozdział trzeci poświęcony został implementacji sterownika systemu chłodzenia. Znajdziemy w nim opis technologii oraz środowiska w którym został napisany, poglądowe diagramy przedstawiające ideę działania sterownika oraz kod wraz z obszernymi opisami.

Projekt systemu i sposób jego wykonania przedstawiony jest w rozdziale czwartym. Rozdział ten zawiera koncept pojemnika oraz dostosowania obudowy do potrzeb projektu. Proces wykonania systemu chłodzącego przedstawiony został na opisanych zdjęciach.

Kolejny rozdział traktuje o pomiarach temperatury podzespołów podczas obciążenia. Wyniki pomiarów zostały zestawione na czytelnych wykresach oraz porównane zostały do wyników tychże testów przeprowadzonych przed modernizacją.

Ostatni rozdział przedstawia możliwości optymalizacji osiągniętych wyników stacji roboczej, wynikające z zastosowanego chłodzenia.



## Rozdział 2

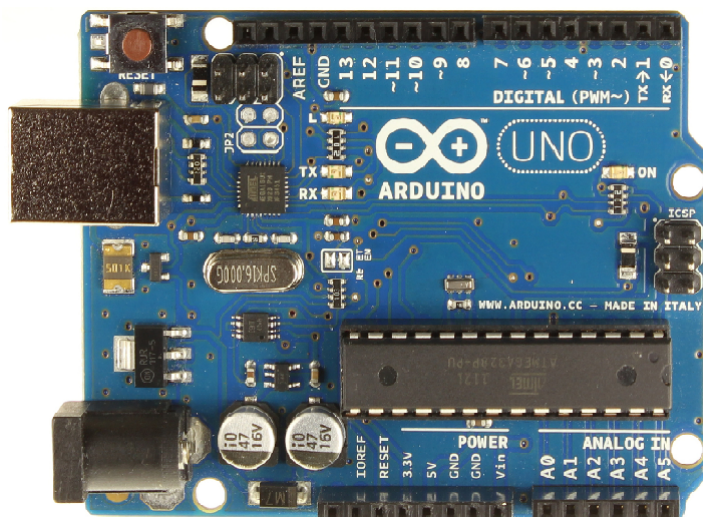
# Opis technologii wykorzystanych podczas realizacji projektu

### 2.1. Arduino

Arduino jest platformą prototypową typu open source, opartą na łatwym w użyciu hrdware oraz software. Płytki Arduino (Rysunek 2.1) są w stanie odczytywać dane różnego pochodzenia - światło padające na sensor, przyciśnięty przycisk, dane z sieci - i przekształcać je w żądane sygnały wyjścia - zapalenie diod LED, opublikowanie czegoś w sieci czy załączenie silnika.

Sama płytki Arduino jest mikrokontrolerem zamontowanym wraz z obsługą wejść/wyjść na pojedynczym układzie drukowanym. Na urządzeniu znajdziemy kontroler, wygodne w użyciu, cufrowe i analogowe linie wejścia/wyjścia, interfejs UART lub USB do komunikacji z komputerem.

Celem projektu Arduino jest dostarczenie szerokiemu gronu ludzi narzędzi wykorzystywanych do tworzenia własnych interaktywnych projektów wymagających sterowania bądź przetwarzania odbieranych sygnałów.



Rysunek 2.1. Arduino UNO

Źródło: [http://blog.khron.net/wp-content/uploads/2014/05/ArduinoUno\\_R3\\_Front.jpg](http://blog.khron.net/wp-content/uploads/2014/05/ArduinoUno_R3_Front.jpg)

#### 2.1.1. Język programowania Arduino

Arduino posiada własny język, język programowania Arduino, który bazuje na środowisku Wiring oraz językach C/C++.

### 2.1.1.1. Wiring

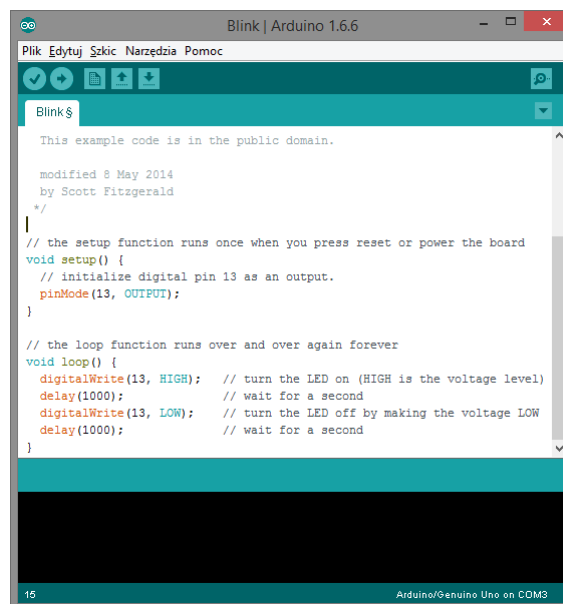
Wiring to platforma developerska, która jest podstawą języka programowania Arduino. Jest to platforma na licencji typu open-source złożona z języka programowania, zintegrowanego środowiska deweloperskiego oraz mikrokontrolera wbudowanego w jedną płytę (jak w przypadku Arduino). Wiring oparty został o projekt Processing, który jest językiem i zintegrowanym środowiskiem stworzonym na MIT (ang. Massachusetts Institute of Technology) Media Lab. Projekt Wiring rozpoczął się w 2003 roku w Instytucie Wzornictwa Interakcji Ivrea. Obecnie rozwijany w Szkole Architektury i Projektowania na Uniwersytecie Los Andes w Kolumbii. Wynalazcą platformy jest Hernando Barragán. Środowisko Wiring IDE zostało napisane w języku JAVA. Zaprojektowano je by wprowadzać elektroników, hobbystów i artystów do programowania elektroniki.

### 2.1.2. Arduino Software (IDE)

Wgrywanie projektów na płytke Arduino odbywa się za pośrednictwem Arduino IDE (ang. Integrated Development Environment). Jest to środowisko zawierające (Rysunek 2.2):

- edytor tekstu posiadający funkcje jak podświetlanie składni, wklejania, kopiowania kodu, jego przeszukiwanie i zamienianie czy automatycznego utrzymywanie wcięć,
- pole komunikatów zwracające wyniki działania Arduino Software IDE np. rezultat zapisywania czy eksportu oraz informujące o błędach, które wystąpiły,
- konsolę wyświetlającą wyjście Arduino Software oraz błędy występujące w kodzie,
- szeregowy monitor wyświetlający dane odebrane z płytki Arduino oraz umożliwiający wysyłanie danych do płytki.

Pliki będące wynikiem pracy w Arduino Software IDE nazywane są skeczami i posiadają rozszerzenie .ino. Arduino Software zbudowane jest na podstawie języka Processing.



Rysunek 2.2. Arduino IDE

### 2.1.2.1. Processing

Processing jest językiem programowania IDE stworzonym przez Ben'a Fry i Casey'a Reas. Powstał w MediaLab działającym w MIT (ang. Massachusetts Institute of Technology). Prace

nad nim rozpoczęły się na potrzeby sztuki elektronicznej oraz projektowania graficznego. Język bazuje na języku Java, jednak ma uproszczoną składnię i graficzny model programowania.

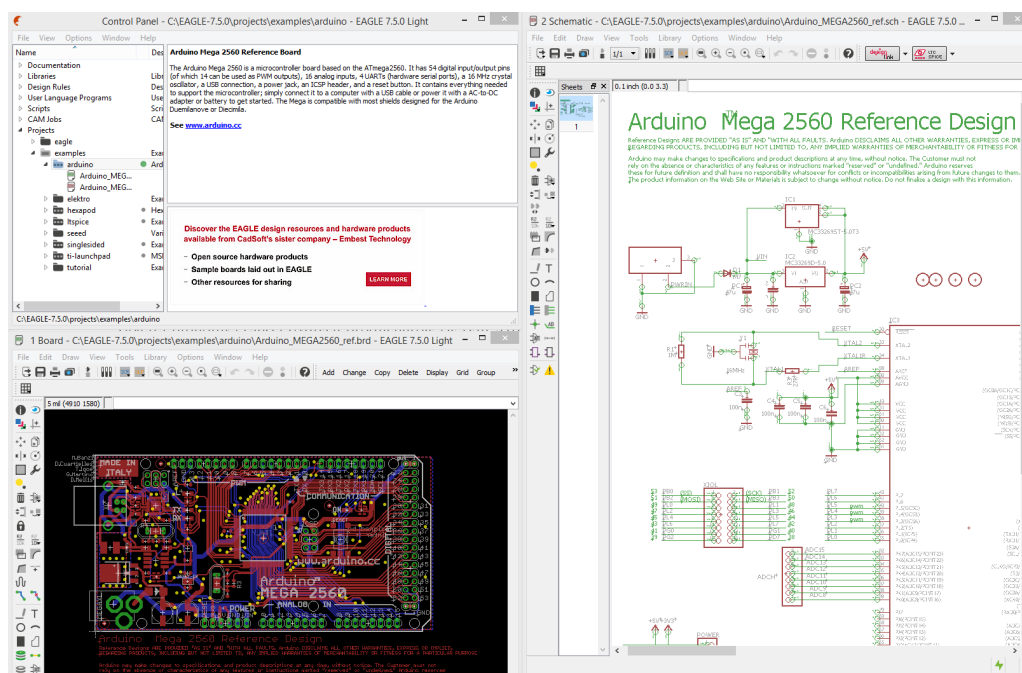
## 2.2. Narzędzia do projektowania elektroniki

### 2.2.1. Eagle

EAGLE (z ang. Easy Applicable Graphical Layout Editor) jest wszechstronnym oprogramowaniem do tworzenia projektów PCB (z ang. Printed Circuit Board). EAGLE wykorzystywany jest przez profesjonalistów pracujących dla światowych koncernów elektronicznych, hobbystów oraz studentów.

Oprogramowanie składa się z trzech modułów z identycznym interfejsem użytkownika (Rysunek 2.3):

- edytor schematów - umożliwia tworzenie symbolicznych, czytelnych schematów, które wzbogacają dokumentację projektu,
- edytor układu elektronicznego - przy pomocy tego modułu zaprojektujemy układ gotowy do wydrukowania,
- autorouter - moduł automatycznego tworzenia ścieżek elektrycznych na projektowanym układzie elektronicznym.

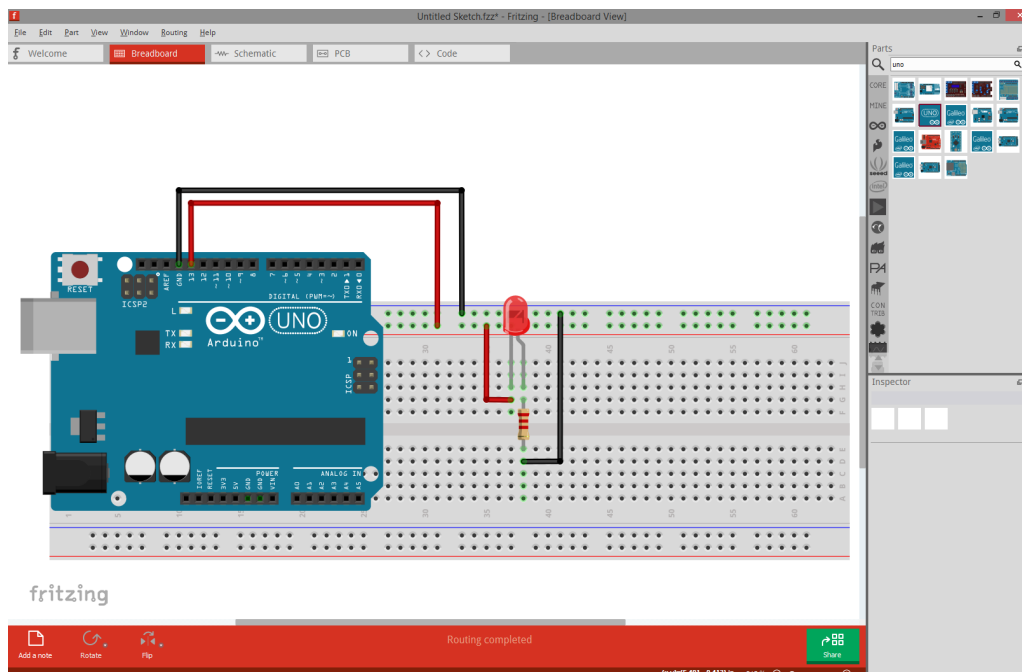


Rysunek 2.3. Program EAGLE - wszystkie okna

### 2.2.2. Fritzing

Fritzing to narzędzie służące do tworzenia schematów oraz rysunków poglądowych projektów elektronicznych. Fritzing jest narzędziem open source. Narzędzie to pozwala na tworzenie czytelnych profesjonalnych schematów, jednak to co wyróżnia to narzędzie to schematy poglądowe (Rysunek 2.4), które pokazują nam faktyczny wygląd skomponowanego układu.

Bogata biblioteka części różnych producentów (w tym Arduino), ciągle rośnie ze względu na możliwość dodawania własnych części i dzieleniem się nimi z innymi użytkownikami Fritzing. Oprogramowanie jest na licencji open source.



Rysunek 2.4. Program Fritzing - edytor schematów poglądowych

## 2.3. CAD

CAD (ang. computer aided design) jest wykorzystaniem systemów komputerowych do wspomagania tworzenia, modyfikacji, analizy lub optymalizacji projektu. Zamajduje zastosowanie w takich dziedzinach jak inżynieria mechaniczna, inżynieria elektryczna czy inżynieria budowlana.

### 2.3.1. AutoCAD

Program umożliwiający projektowanie wspomagane komputerowo. Program stworzony został przez firmę Autodesk, a jego pierwsza wersja zaprezentowana została na targach COMDEX w Las Vegas w listopadzie 1982 roku, sprzedaż produktu ruszyła miesiąc później. Program jest nadal rozwijany i jego najnowsza wersja nosi nazwę AutoCAD 2016. AutoCAD umożliwia wspomaganie projektowania 2D, 3D oraz dzięki specjalnym dodatkom 2.5D. Pierwotnie wykorzystywany był tylko przez mechaników, jednak dzięki rozszerzeniu programu przez wiele nakładek np. AutoCAD Electrical, AutoCAD Mechanical, Architectural Desktop, Civil Design itp. stał się platformą wykorzystywaną przez architektów i projektantów różnych dziedzin inżynierii.

## **2.4. Narzędzia do testów stabilności i benchmark**

### **2.4.1. OCCT**

OCCT (ang. OverClock Checking Tool) jest programem służącym do sprawdzania stabilności pracy stacji roboczej. Poza stabilnością sprawdzimy także temperatury osiągane przez poszczególne podzespoły czy napięcie występujące na tych podzespołach. Poza wymienionymi danymi program dostarcza również informacje o posiadanym procesorze czy karcie graficznej. Program podczas działania bardzo mocno obciąża komputer (wykorzystanie procesora ~98 - 100%), jednocześnie dając możliwość stałego kontrolowania wyżej wymienionych danych. W programie dostępne są cztery testy: autorski, test na bazie biblioteki LINPACK stworzonej przez Intel, test sprawdzający kartę graficzną pod względem renderowania obiektów 3D oraz test zasilacza. Wyniki przeprowadzonych testów obrazowane są na wykresach zapisanych na dysku w postaci plików .png.

### **2.4.2. Catzilla**

Program typu benchmark stworzony przez autorów popularnego programu ALLPlayer. Program wykorzystuje silnik graficzno-fizyczny stworzony przez studio Plastic. Catzilla testuje wydajność sprzętu komputerowego z naciskiem na jego graficzne możliwości. Poza zdolnościami do generowania grafiki program sprawdzi zdolność procesora do symulowania fizyki. Wyniki testu przedstawione są w punktach, co daje bardzo czytelny i łatwy do porównania wynik (im więcej punktów tym lepiej).

## Rozdział 3

# Realizacja projektu systemu chłodzącego

### 3.1. Elektroniczny układ współpracujący z Arduino

Płytką Arduino zapewnia wyprowadzenie pinów komunikacyjnych mikrokontrolera. Zadaniem osoby budującej projekt na jego podstawie jest zaprojektowanie i wykonanie całego zestawu urządzeń peryferyjnych.

W przypadku tego projektu niezbędne będą urządzenia do komunikacji z użytkownikiem, ponieważ to on decyduje o momencie załączenia pompy oraz będzie miał możliwość śledzenia temperatury cieczy. Cały układ poza odebraniem danych od użytkownika musi być w stanie zmierzyć temperaturę cieczy wewnątrz pojemnika z komputerem. Jak wiadomo płyta główna jest sporym podzespołem więc rozmiary pojemnika, w którym zanurzony będzie komputer będą znaczące dla odczytu temperatury. Zastosowano więc dwa czujniki temperatury, które odczytywać będą temperaturę z dwóch różnych miejsc w pojemniku. Ciecz znajdująca się wewnątrz pojemnika musi mieć gdzie oddać ciepło przejęte z podzespołów. W tym celu zastosowano pompę, która przetłoczy ciecz przez zewnętrzną chłodnicę. Pracując przy komputerze użytkownicy często skupiają się tylko na swoim zadaniu. By system był efektywny zastosowano właśnie automatyczne sterowanie załączeniem pompy, wykorzystujące Arduino.

#### 3.1.1. Projekt układu elektronicznego

Komunikacja z mikrokontrolerem jest istotnym elementem tego projektu. Arduino ma pokazywać aktualną średnią temperaturę cieczy, oraz temperaturę po przekroczeniu której załączona zostanie pompa. Temperatura załączenia pompy ustawiana jest przez użytkownika. Tak więc do komunikacji niezbędny będzie wyświetlacz. Do tego typu danych wybrany został wyświetlacz LCD posiadający dwa rzędy znaków po 16 znaków w każdym. Do ustawiania temperatury wykorzystane zostały dwa przyciski monostabilne, których zwarcie monitorowane będzie przez mikrokontroler.

Układ elektroniczny systemu chłodzącego musi również być w stanie mierzyć temperaturę cieczy. Do tego zadania wybrane zostały dwa czujniki LM35, które cechują się szerokim zakresem pomiaru (od  $-50^{\circ}\text{C}$  do  $+150^{\circ}\text{C}$ ), w którym mieści się zakres odczytywanych temperatur. Ciecz minimalnie będzie osiągała temperaturę pokojową, a maksymalnie nie przekroczy  $100^{\circ}\text{C}$ .

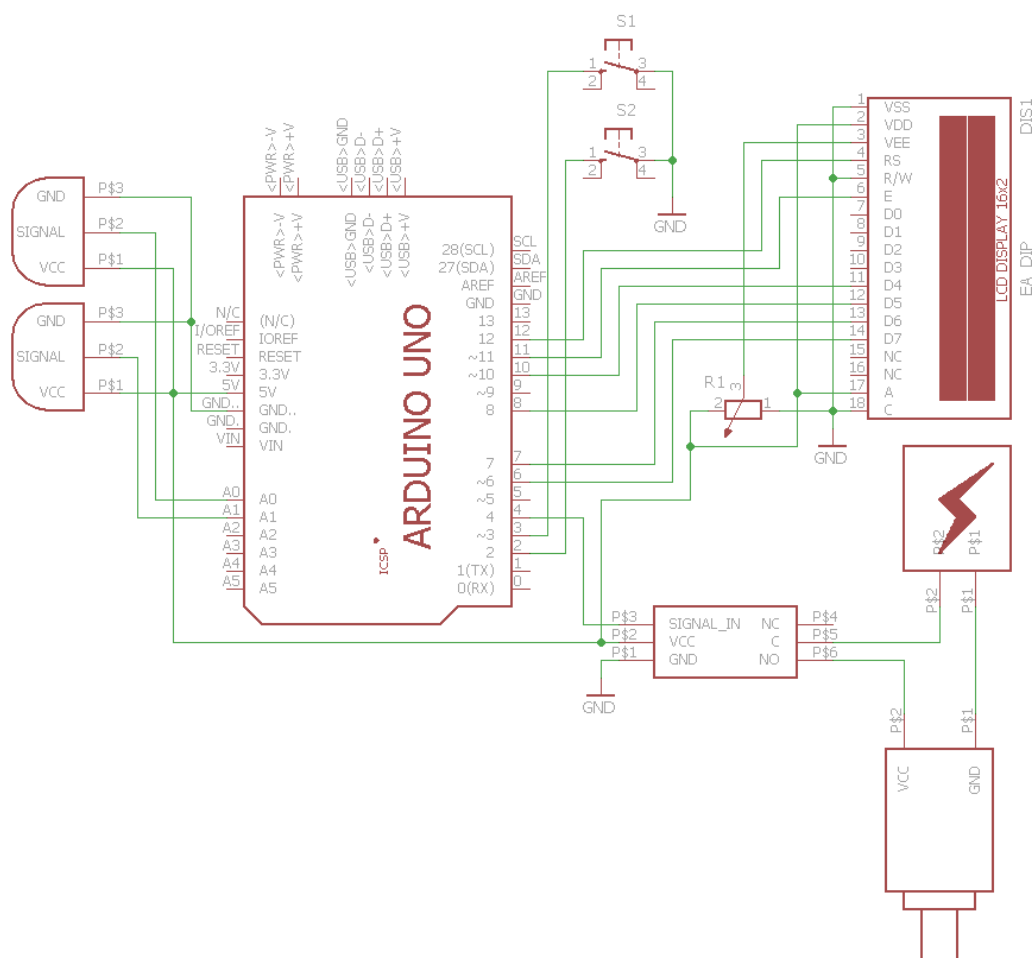
Nagrzaną przez podzespoły cieczy należy wychłodzić poza zbiornikiem z komputerem. Aby przetłoczyć ciecz poza pojemnik wykorzystano pompkę elektryczną, której wymagane zasilanie znacznie przewyższa możliwości Arduino. Wykorzystano więc przekaźnik, który wysterowany przez Arduino załączy pompkę w odpowiednim momencie.

Podsumowując, nasz układ elektroniczny składać się będzie z:

- płytki Arduino,
- wyświetlacza LCD 2 razy po 16 znaków,
- dwóch przycisków monostabilnych,
- dwóch czujników temperatury LM35,

- przekaźnika,
- pompki,
- zewnętrznego źródła zasilania.

Połączenia między wszystkimi podzespółami zostały zaprojektowane zgodnie z notami katalogowymi poszczególnych podzespołów. Projekt układu elektronicznego przedstawiony został na schemacie (Rysunek 3.1).

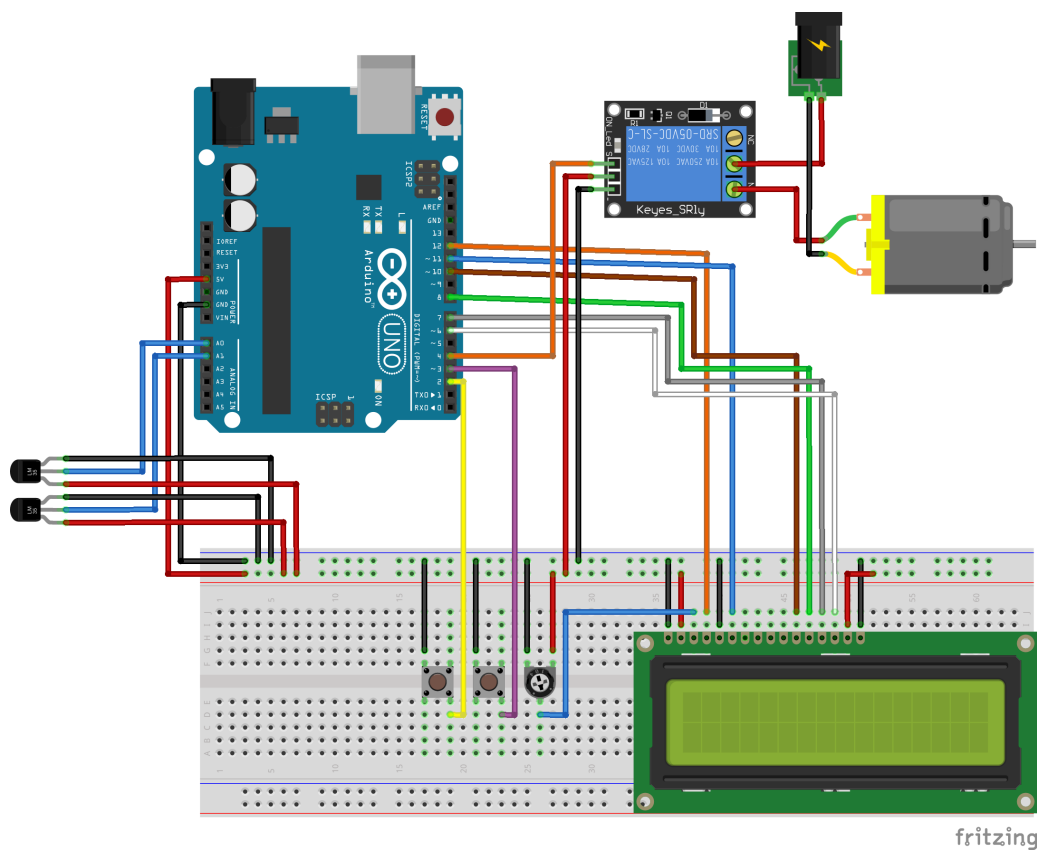


Rysunek 3.1. Schemat elektryczny układu współpracującego z Arduino wykonany w EAGLE

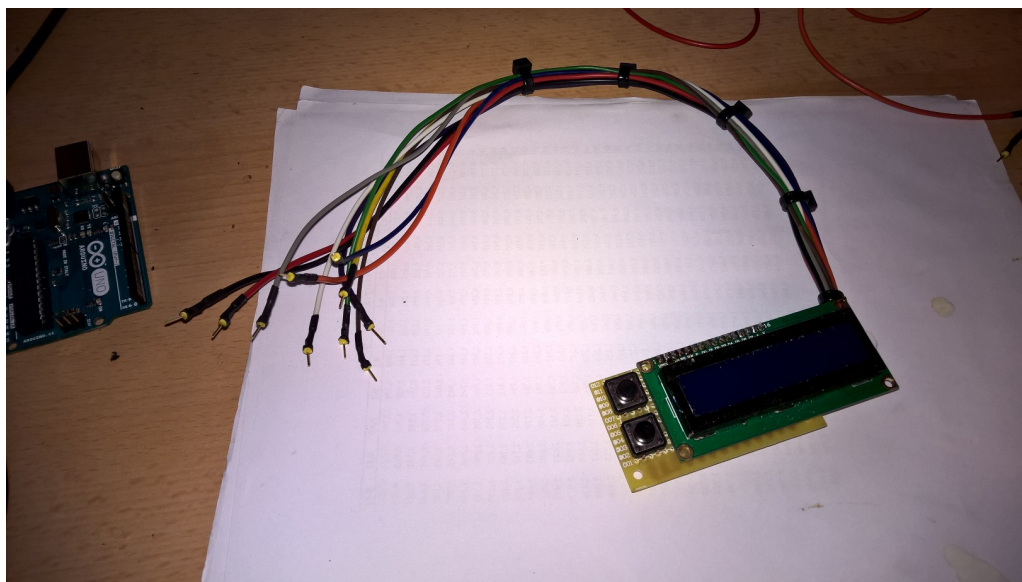
By zobrazować połączenie między poszczególnymi elementami wykonano również schemat poglądowy. Na tym schemacie (Rysunek 3.2) symbole zostały zastąpione grafikami przybliżający faktyczny wygląd poszczególnych elementów (nie zostały zachowane proporcje ze względu na zróżnicowany rozmiary elementów układu), oraz przedstawione wszystkie niezbędne fizyczne połączenia.

### 3.1.2. Wykonanie układu elektronicznego

Wykonanie układu elektronicznego rozpoczęto od wykonania modułu odpowiedzialnego za komunikację z użytkownikiem. Elementy zamontowane na płytce stykowej projektu (Rysunek 3.2) zostały zamontowane na płytce prototypowej. Montaż części odpowiedzialnych za komunikację na jednej płytce umożliwił wyprowadzenie tego modułu na długich przewodach i umieszczenie ich w widocznym miejscu obudowy. Przewody zakończone zostały goldpinami w celu umożliwienia podpięcia ich do płytki Arduino (Rysunek 3.3).



Rysunek 3.2. Schemat poglądowy wykonany we Fritzing



Rysunek 3.3. Zmontowany moduł komunikacji

Czujniki temperatury również zostały przylutowane do długich przewodów by można było umieścić je w odpowiednich miejscach w obudowie. Zasilacz, pompa i przełącznik są elementami gotowymi. Wystarczyło je tylko odpowiednio podłączyć do płytki Arduino lub przełącznika (Rysunek 3.4) i zaimplementować ich pracę w oprogramowaniu do Arduino.





Rysunek 3.4. Podłączony układ elektryczny współpracujący z Arduino

## 3.2. Implementacja sterownika układu chłodzącego w środowisku Wiring

### 3.2.1. Projekt algorytmu głównego

Algorytm główny sterownika ma kilka zadań. Jednym z nich jest stałe monitorowanie odczytu czujników temperatury i na podstawie tych odczytów, obliczanie i wyświetlanie temperatury cieczy na ekranie. Pomiar temperatury cieczy znajdującej się w obudowie jest również wyznacznikiem momentu załączenia pompy. Gdy temperatura cieczy w obudowie przekroczy ustawioną przez użytkownika temperaturę krytyczną, Arduino ma za zadanie załączenie pompy, która przetłoczy ciecz przez chłodnicę. Gdy temperatura cieczy spadnie poniżej poziomu krytycznego, Arduino wyłączy pompę. Poziom krytyczny, o którym mowa, jest ustawiany przez użytkownika przy pomocy przycisków monostabilnych. Nasłuchiwanie wspomnianych przycisków to również jedno z zadań sterownika. Praca sterownika przedstawiona została na diagramie przepływu (Rysunek 3.5).

### 3.2.2. Implementacja algorytmu

kod razem z objaśnieniami

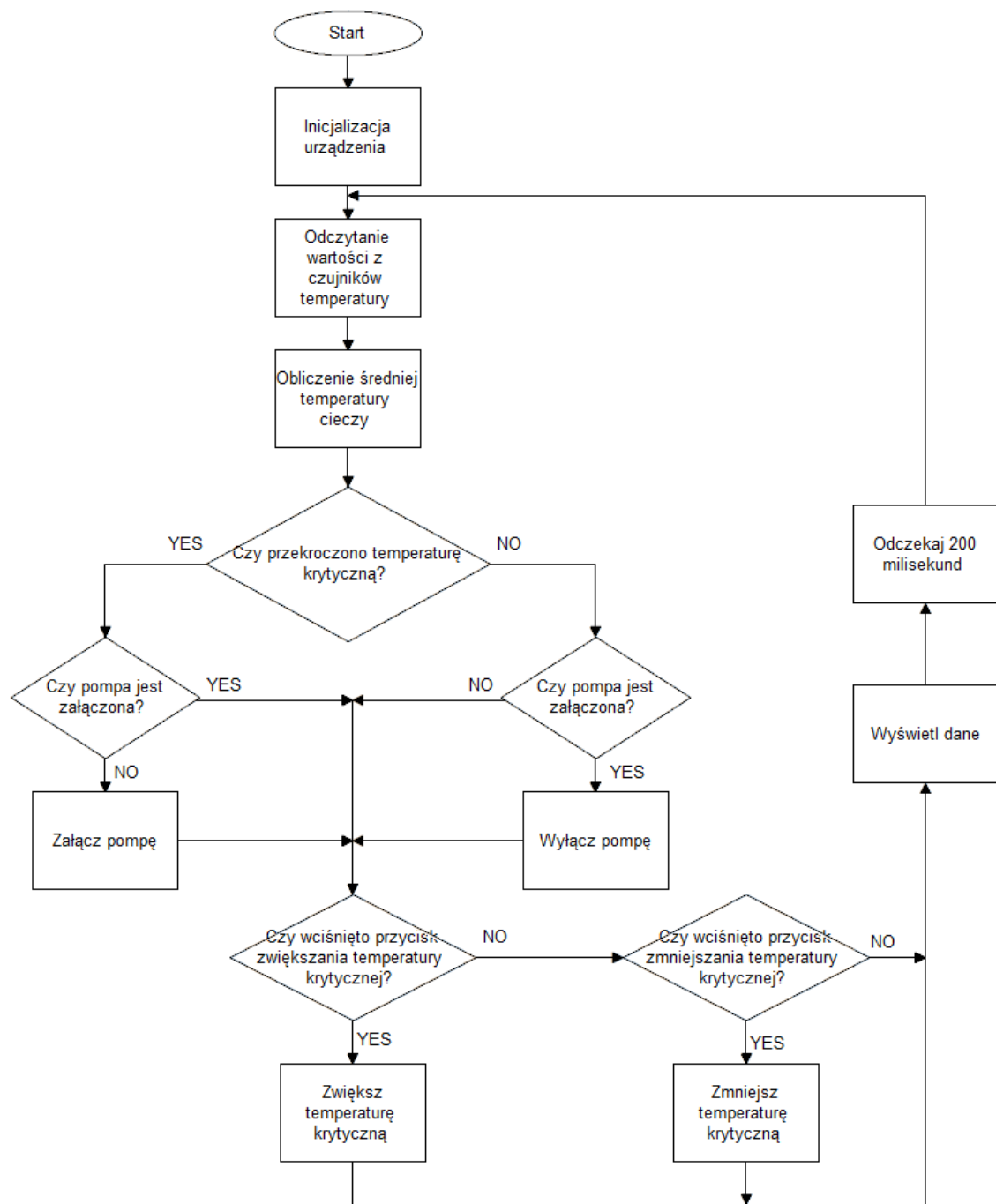
## 3.3. System chłodzący

### 3.3.1. Projekt systemu chłodzącego

projekt całej instalacji wraz z chłodnicą i tak dalej

### 3.3.2. Wykonanie systemu chłodzącego

poszczególne kroki niezbędne do wykonania instalacji przedstawione na opisanych zdjęciach



Rysunek 3.5. Diagram przepływu pracy sterownika

## Rozdział 4

# Testy wydajnościowe, optymalizacja osiągnięć stacji roboczej oraz porównanie wyników z danymi wyjściowymi

w tym rozdziale znajdują się opisy testów i ich przebieg, wykresy temperatur zmierzonych na rdzeniach procesora podczas obciążenia po wykonaniu instalacji oraz te przed, szybkie porównanie i podsumowanie wyników dotyczących temperatury

### 4.1. Opis zagadnienia overclockingu

### 4.2. Opis parametrów przed optymalizacją

### 4.3. Opis optymalizacji

### 4.4. Testy wydajnościowe

### 4.5. Analiza wyników optymalizacji

### 4.6. Analiza wyników testów wydajnościowych

## Rozdział 5

# Podsumowanie

ogólne podsumowanie pracy  
co udało się osiągnąć  
co jest zbędne a co konieczne  
czy jest zauważalna różnica

## Spis rysunków

1.1.	Gotowy projekt . . . . .	5
2.1.	Arduino UNO . . . . .	7
2.2.	Arduino IDE . . . . .	8
2.3.	Program EAGLE - wszystkie okna . . . . .	9
2.4.	Program Fritzing - edytor schematów poglądowych . . . . .	10
3.1.	Schemat elektryczny układu współpracującego z Arduino wykonany w EAGLE . . . . .	13
3.2.	Schemat poglądowy wykonany we Fritzing . . . . .	14
3.3.	Zmontowany moduł komunikacji . . . . .	14
3.4.	Podłączony układ elektryczny współpracujący z Arduino . . . . .	15
3.5.	Diagram przepływu pracy sterownika . . . . .	16

## Bibliografia