



POLITECHNIKA ŚLĄSKA
WYDZIAŁ AUTOMATYKI, ELEKTRONIKI I
INFORMATYKI

Zakład Pomiarów i Systemów Sterowania

Instytut Automatyki

PRACA DYPLOMOWA MAGISTERSKA

WIELOKANAŁOWY REGULATOR

DEDYKOWANY DO KOTŁOWNI

Dyplomant: Tomasz Dylla

Kierujący pracą: dr inż. Józef Wiora

Gliwice, czerwiec 2011

Spis treści

Spis tablic	5
Spis rysunków	7
Lista symboli	9
Rozdział 1. Wstęp	11
1.1. Wprowadzenie	11
1.2. Cel i zakres pracy	12
1.3. Stawiane wymagania	12
1.3.1. Przetwornik temperatury na napięcie	12
1.3.2. Układ mikroprocesorowy	12
Rozdział 2. Termistor i pomiar temperatury	15
2.1. Wprowadzenie teoretyczne	15
2.2. Termistor NTC	16
2.3. Układy do pomiaru temperatury z zastosowaniem termistora NTC	18
2.3.1. Połączenie szeregowo	18
2.3.2. Mostek Wheatstone’a	19
2.3.3. Cyfrowa aproksymacja charakterystyki termistora NTC	20
2.3.4. Analogowa linearyzacja charakterystyki termistora NTC	21
Rozdział 3. Projekt przetwornika temperatury na napięcie	23
3.1. Wzmacniacz operacyjny w układzie logarytmującym	23
3.2. Wzmacniacz operacyjny w układzie sumującym	24
3.3. Budowa przetwornika temperatury na napięcie	25
3.3.1. Wzmacniacz operacyjny $\mu A741$	25
3.3.2. Schemat układu	26
3.3.3. Projekt płytki PCB i wykonanie	28
3.4. Otrzymane wyniki	29
Rozdział 4. Projekt układu mikroprocesorowego	31

4.1. Wykorzystany mikrokontroler	31
4.2. Układ pracy mikrokontrolera	33
4.3. Komunikacja mikrokontrolera z komputerem	34
4.4. Budowa układu mikroprocesorowego	36
4.4.1. Schemat układu	36
4.4.2. Projekt płytki PCB i wykonanie	38
4.5. Oprogramowanie	39
4.5.1. Przetwornik analogowo cyfrowy	39
4.5.2. Komunikacja USART	44
Rozdział 5. Zakończenie	49
5.1. Podsumowanie i wnioski	49
5.2. Możliwości dalszego rozwoju	50
Bibliografia	51

Spis tablic

3.1	Wyniki badania przetwornika temperatury na napięcie	29
3.2	Błędy pomiarów uzyskane podczas badania przetwornika	30
4.1	Opis wyprowadzeń interfejsu ISP	34
4.2	Wybór napięcia odniesienia za pomocą rejestru <i>ADMUX</i>	40
4.3	Wybór kanału przetwornika analogowo cyfrowego za pomocą rejestru <i>ADMUX</i>	41
4.4	Opis bitów rejestru <i>ADCSRA</i>	41
4.5	Wybór współczynnika podziału częstotliwości za pomocą rejestru <i>ADMUX</i>	42
4.6	Opis bitów rejestru <i>UCSRA</i>	44
4.7	Opis bitów rejestru <i>UCSRB</i>	45
4.8	Opis bitów rejestru <i>UCSRC</i>	45
4.9	Opis bitów rejestru <i>UBRRH</i> i <i>UBRRL</i>	46

Spis rysunków

2.1	Zależność rezystancji od temperatury termistora NTC 10K	17
2.2	Szeregowe połączenie termistora NTC 10K	18
2.3	Przebieg napięcia wyjściowego w zależności od temperatury szeregowo połączonego termistora NTC 10K z rezystorem o wartości 10k Ω	19
2.4	Schemat połączenia termistora NTC 10K w mostku Wheatstone'a	19
2.5	Przebieg napięcia wyjściowego układu w zależności od temperatury z włączonym termistorem NTC 10K w mostek Wheatstone'a	20
2.6	Wzmacniacz operacyjny w układzie logarytmującym	21
3.1	Napięcie wyjściowe w zależności od napięcia wejściowego układu logarytmującego 24	
3.2	Wzmacniacz operacyjny w układzie sumującym	25
3.3	Wzmacniacz operacyjny $\mu A741$	25
3.4	Sposób połączenia umożliwiający kompensację wejściowego napięcia niezrównoważenia wzmacniacza $\mu A741$	26
3.5	Schemat wykonanego przetwornika temperatury na napięcie	27
3.6	Projekt wykonanego przetwornika temperatury na napięcie	28
3.7	Wykonany przetwornik temperatury na napięcie	28
3.8	Wykres napięcia wyjściowego w zależności od temperatury wykonanego przetwornika	29
4.1	Wykorzystany mikrokontroler	32
4.2	Złącze ISP	34
4.3	Wymagane połączenia odpowiednich wyprowadzeń do prawidłowego działania mikrokontrolera	35
4.4	Schemat typowego konwertera $TTL \leftrightarrow RS232$ z wykorzystaniem MAX232	35
4.5	Złącze DB9	36

4.6	Schemat wykonanego układu mikroprocesorowego	37
4.7	Projekt wykonanego układu mikroprocesorowego	38
4.8	Wykonany układ mikroprocesorowy	38
4.9	Rejestr <i>ADMUX</i> mikroprocesora ATmega8	39
4.10	Rozmieszczenie wyników pomiaru w zależności od stanu bitu <i>ADLAR</i>	40
4.11	Rejestr <i>ADCSRA</i> mikroprocesora ATmega8	41
4.12	Rejestr <i>UCSRA</i> mikroprocesora ATmega8	44
4.13	Rejestr <i>UCSRB</i> mikroprocesora ATmega8	45
4.14	Rejestr <i>UCSRC</i> mikroprocesora ATmega8	45
4.15	Rejestr <i>UBRRH</i> i <i>UBRRL</i> mikroprocesora ATmega8	46

Lista symboli

U_{ADC} – spadek napięcia na termistorze

U_{REF} – stałe napięcie odniesienia

ρ – opór właściwy

α – temperaturowy współczynnik oporu właściwego

T – temperatura

T_0 – temperatura odniesienia

R_T – rezystancja termistora

B – stała materiałowa

R_0 – rezystancja termistora w temperaturze T_0

R – opór elektryczny

S – pole przekroju poprzecznego przewodnika

l – długość przewodnika

U_{WY} – napięcie wyjściowe

U_{WE} – napięcie wejściowe

k – stała Boltzmannna

q – ładunek elektronu

I_{WE} – prąd wejściowy

I_{E0} – prąd wsteczny złącza emiterowego

Rozdział 1

Wstęp

1.1. Wprowadzenie

We współczesnym świecie utrzymanie odpowiedniej temperatury w budynkach mieszkalnych i zakładach pracy jest bardzo istotną oraz kosztowną kwestią. Automatyczna regulacja pozwala nie tylko na wygodę użytkowników centralnego ogrzewania, ale również na duże oszczędności. Obecne urządzenia wykorzystywane do sterowania piecami posiadają pojedyncze czujniki temperatury pieca oraz temperatury na zewnątrz budynku. Sterownik na podstawie zaprogramowanej krzywej grzania i odczytu temperatury na zewnątrz budynku dobiera tak temperaturę pieca, aby została utrzymana zadana temperatura wewnątrz budynku¹. W praktyce okazuje się jednak, że w poszczególnych pomieszczeniach budynków nie jest potrzebna taka sama temperatura. Za optymalną temperaturę w poszczególnych pomieszczeniach uważa się:

- w pokojach mieszkalnych 20–21°C,
- w kuchni i sypialni 18°C,
- w łazience 24°C.

Najprostszym rozwiązaniem przedstawionego problemu jest zastosowanie na grzejnikach głowic termostatycznych. Mają one niestety dwie poważne wady. Głowica steruje pracą grzejnika w funkcji temperatury powietrza, które ją owiewa, a nie w funkcji temperatury w pomieszczeniu, co jest konsekwencją dużych różnic między temperaturąadaną, a wysterowaną². Kolejną wadą głowic jest brak możliwości automatycznego sterowania. W niektórych pomieszczeniach podwyższona temperatura nie jest wymagana przez cały dzień. Przykładowo w łazience temperatura 24°C wymagana będzie podczas kąpieli użytkowników, zaś o innych porach dnia temperaturę można obniżyć do poziomu 21°C. Ręczne ustawianie temperatury jest bardzo niewygodne, dlatego najlepszym rozwiązaniem byłoby

¹ http://muratorodom.pl/instalacje/kotly-i-automatyka/automatyka-pogodowa,31_2303.html [dostęp: czerwiec 2011].

² http://termodom.pl/buduj/ogrzewanie/glowice_i_zawory_termostatyczne [dostęp: czerwiec 2011].

zastosowanie wielokanałowego regulatora, który automatycznie regulowałby temperaturę w poszczególnych pomieszczeniach w zależności od zaprogramowanych wymagań użytkownika.

1.2. Cel i zakres pracy

Celem pracy jest zbudowanie wielokanałowego regulatora, umożliwiającego pomiar temperatury oraz ustawianie odpowiednich wyjść w celu regulacji temperatury w poszczególnych pomieszczeniach. Zakres pracy obejmuje zaprojektowanie i wykonanie przetwornika do pomiaru temperatury oraz zaprojektowanie, wykonanie i zaprogramowanie układu mikroprocesorowego do odczytu temperatury oraz do regulacji otwarcia odpowiednich elektrozaworów na grzejnikach.

1.3. Stawiane wymagania

1.3.1. Przetwornik temperatury na napięcie

Do pomiaru temperatury, ze względu na bardzo niską cenę oraz łatwą dostępność, został wykorzystany termistor NTC 10K. Przetwornik ma za zadanie przetworzenie temperatury w zakresie $-20^{\circ}\text{C} - +100^{\circ}\text{C}$ na napięcie w zakresie 0–5 V. Wymagany jest tak duży zakres temperatury, ponieważ jeden z czujników będzie umieszczony w piecu i będzie podawał temperaturę ogrzewanej przez piec wody. Zakres napięcia jest wynikiem chęci uzyskania maksymalnej rozdzielczości pomiaru za pomocą przetwornika A/C wybranego mikroprocesora.

1.3.2. Układ mikroprocesorowy

Głównym elementem układu mikroprocesorowego jest mikrokontroler ATmega8 firmy Atmel. Układ został wyposażony w 6 wejść analogowo cyfrowych oraz 6 dwustanowych wyjść. Głównym zadaniem układu jest pomiar napięcia dostarczonego przez przetwornik,

konwersja napięcia na odpowiadającą mu temperaturę oraz wysłanie otrzymanej temperatury za pomocą interfejsu RS 232. Dodatkową funkcją układu jest oczekiwanie na sterowanie przesyłane za pomocą interfejsu RS 232 i zgodnie z otrzymaną informacją ustawienie odpowiednich wyjść. Układ również jest wyposażony w złącze ISP w celu ewentualnej modyfikacji programu mikrokontrolera.

Rozdział 2

Termistor i pomiar temperatury

2.1. Wprowadzenie teoretyczne

Wielkością charakterystyczną dla rodzaju materiału jest opór właściwy. Zależy on tylko od materiału, z którego jest zrobiony i wyraża liczbowo opór sześciangu o krawędzi jednego metra przy przepływie prądu od jednej ściany do przeciwległej. Dla jednorodnych materiałów izotropowych jest określony wzorem [1]:

$$(1) \quad \rho = \frac{RS}{l}$$

gdzie: R - opór elektryczny, S - pole przekroju poprzecznego przewodnika, l - długość przewodnika. Opór właściwy jest wielkością zależną od temperatury. Dla danego ciała jest on proporcjonalny do oporu elektrycznego. Zależność oporu elektrycznego od temperatury można opisać równaniem [2]:

$$(2) \quad R = R_0[1 + \alpha(T - T_0)]$$

gdzie: R_0 - opór elektryczny w temperaturze T_0 , α - średni współczynnik temperatury oporu właściwego. Po przekształceniu równania na R otrzymuje się:

$$(3) \quad \alpha = \frac{1}{R_0} \frac{dR}{dT}$$

Równanie (3) przedstawia wzór na temperaturowy współczynnik oporu właściwego. Współczynnik ten określa względną zmianę oporu właściwego przypadającego na jednostkę zmiany temperatury. Półprzewodniki o dużym współczynniku temperaturowym oporu właściwego nazywa się termistorami. Nazwa ta pochodzi od skrótu angielskiej nazwy „thermally sensitive resistors” – rezystory czułe na temperaturę. Termistory wykonuje się z tlenków: manganu, niklu, kobaltu, miedzi, glinu, wanadu i litu. Właściwości termistora zależą od rodzaju i proporcji wykorzystanych do ich produkcji tlenków. Wyróżnia się następujące typy termistorów¹:

¹ <http://pl.wikipedia.org/wiki/Termistor> [dostęp: czerwiec 2011].

- NTC – o ujemnym współczynniku temperaturowym (ang. negative temperature coefficient) – wzrost temperatury powoduje zmniejszanie się rezystancji;
- PTC – (pozystor) o dodatnim współczynniku temperaturowym (ang. positive temperature coefficient), wzrost temperatury powoduje wzrost rezystancji;
- CTR – o skokowej zmianie rezystancji (ang. critical temperature resistor) – wzrost temperatury powyżej określonej powoduje gwałtowną zmianę (wzrost/spadek) rezystancji. W termistorach polimerowych następuje szybki wzrost rezystancji, a w ceramicznych, zawierających związki baru, spadek.

2.2. Termistor NTC

Termistor NTC jest nieliniowym rezystorem, którego rezystancja jest silnie zależna od temperatury materiału oporowego. Zgodnie z jego angielską nazwą – Negative Temperature Coefficient – termistor posiada ujemny współczynnik temperaturowy, co oznacza, że ze wzrostem temperatury rezystancja maleje. W półprzewodnikach o niskiej temperaturze istnieje niewielka liczba elektronów swobodnych, ponieważ większość elektronów związana jest z atomami. Z tego powodu półprzewodniki w niskich temperaturach mają duży opór właściwy. Wzrost temperatury powoduje zwiększenie liczby elektronów swobodnych, co powoduje zmniejszenie oporu. Z drugiej strony podwyższenie temperatury wywołuje wzrost ruchu cieplnego atomów. Takie zjawisko utrudnia uporządkowany ruch elektronów, czego następstwem jest wzrost oporu. W przypadku półprzewodników wpływ związany ze wzrostem ruchu atomów na przewodnictwo okazuje się mniejszy niż wpływ zwiększonej liczby elektronów swobodnych, więc oporność maleje [2].

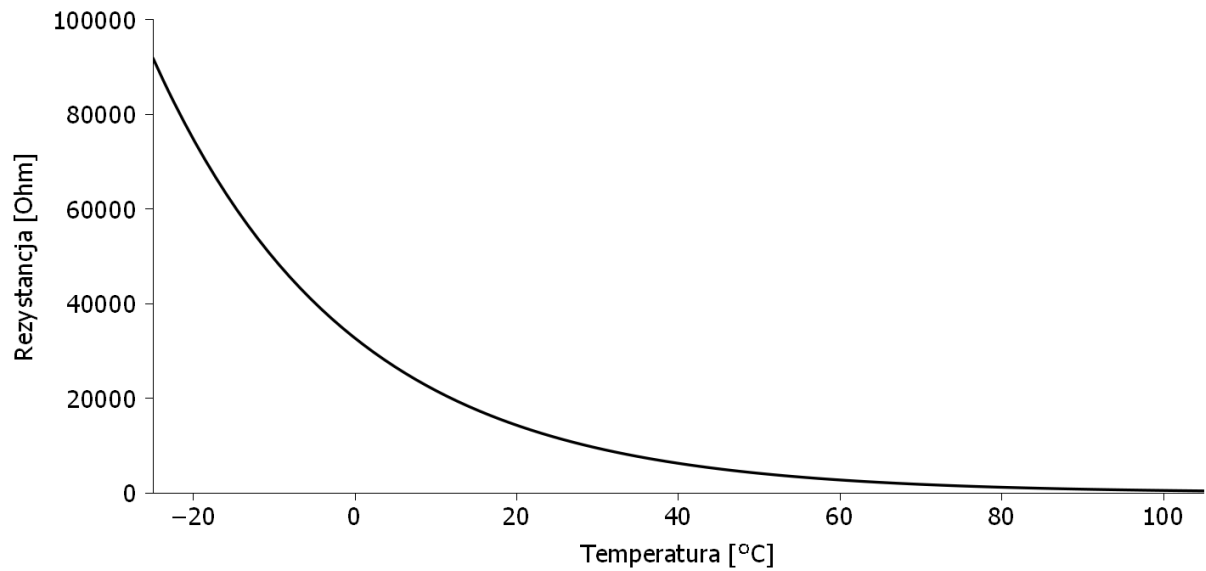
Charakterystykę $R(T)$ termistorów NTC przedstawiono na Rys. 2.1. Z wystarczającym w praktyce przybliżeniem można ją opisać następująco²:

$$(4) \quad R_T = R_0 \exp \left[B \left(\frac{1}{T} - \frac{1}{T_0} \right) \right]$$

gdzie: T jest temperaturą termistora w kelwinach (K), T_0 jest temperaturą odniesienia (zwykle 298K), R_0 jest wartością rezystancji termistora w temperaturze T_0 oraz B jest stałą materiałową termistora mającą wartość 2000–6000K. Termistory NTC są bardzo tanie, łatwo dostępne i stosuje się je głównie do:

² http://www.eres.alpha.pl/elektronika/articles.php?article_id=383 [dostęp: czerwiec 2011].

- pomiarów i regulacji temperatury,
- kompensacji temperaturowej,
- opóźnień czasowych,
- ograniczenia prądów rozruchu.



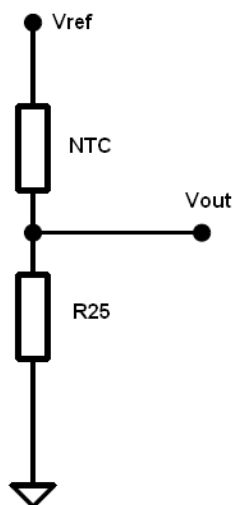
RYSUNEK 2.1. Zależność rezystancji od temperatury termistora NTC 10K

2.3. Układy do pomiaru temperatury z zastosowaniem termistora NTC

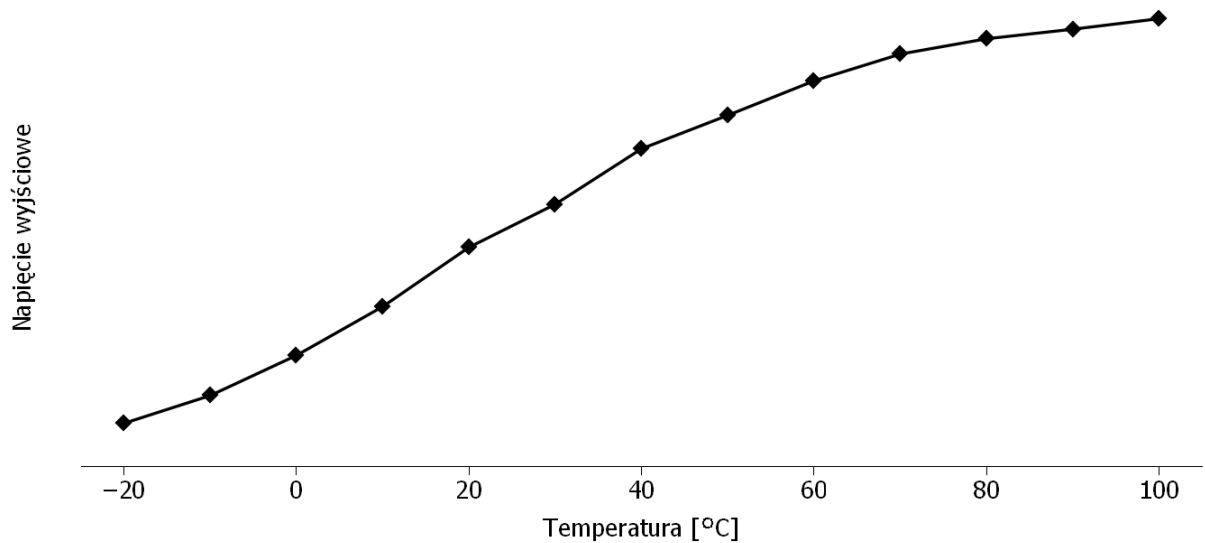
Zastosowanie odpowiedniego układu zależy od wymaganego zakresu temperatury w jakim układ pracuje. Błędny dobór może spowodować bardzo duże błędy w pomiarze. Występują układy łatwe i tanie do zrealizowania, których zakres pomiaru jest bardzo wąski oraz układy trudniejsze w realizacji, wymagające zastosowania wzmacniaczy operacyjnych, bądź skomplikowanego programowania układu mikroprocesorowego. W tym przypadku pomiary temperatury w większych zakresach będą dawały małe akceptowalne błędy.

2.3.1. Połączenie szeregowe

Na Rys.2.2 przedstawiono szeregowe połączenie termistora NTC wraz z innym rezystorem [3]. Jest to najprostszy układ do pomiaru temperatury. Wartość rezystora ma duże znaczenie. Należy go dobrać w zależności od wymaganego zakresu mierzonej temperatury. Układ taki daje wyniki z dokładnością $\pm 1^{\circ}\text{C}$ w zakresie $\pm 25^{\circ}\text{C}$ od temperatury jaka odpowiada rezystancji termistora równej rezystancji dodatkowego rezystora. Przykładowo rezystancja termistora NTC w temperaturze 25°C wynosi $10\text{k}\Omega$. W przypadku połączenia szeregowego rezystora o takiej rezystancji z termistorem NTC 10K otrzymamy linową charakterystykę w zakresie $0^{\circ}\text{C} - +50^{\circ}\text{C}$. Napięcie wyjściowe takiego układu przedstawione jest na Rys. 2.3.



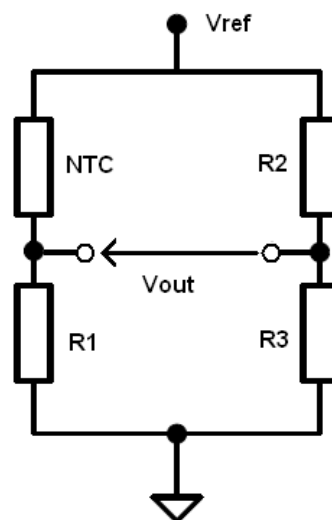
RYSUNEK 2.2. Szeregowe połączenie termistora NTC 10K



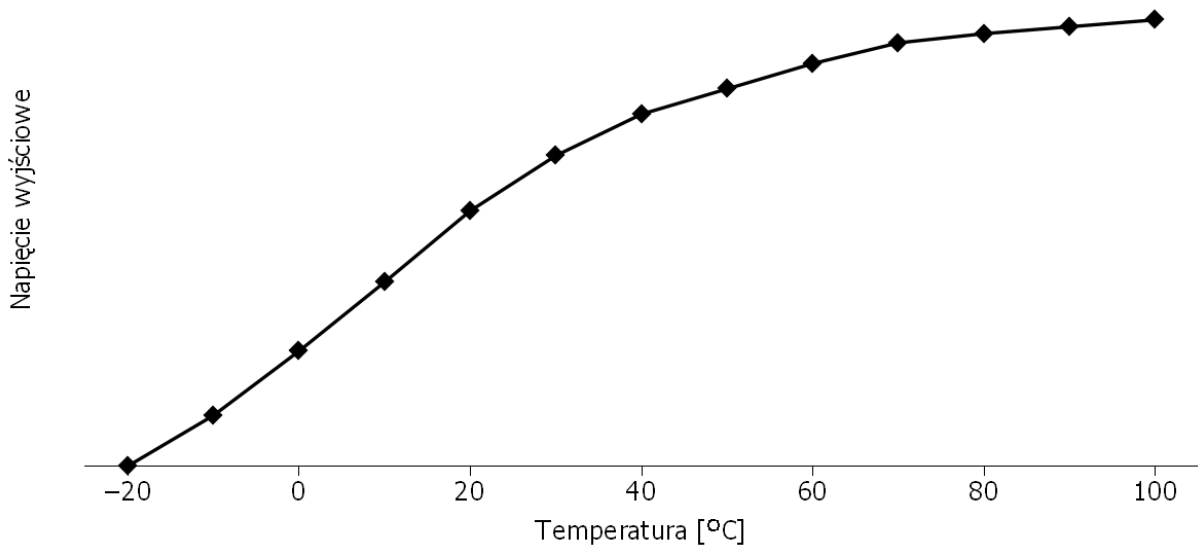
RYSUNEK 2.3. Przebieg napięcia wyjściowego w zależności od temperatury szeregowo połączonego termistora NTC 10K z rezystorem o wartości 10k Ω

2.3.2. Mostek Wheatstone'a

Na Rys.2.4 przedstawiono połączenie termistora NTC jako jednego z elementów mostka Wheatstone'a [3]. Jest to również bardzo prosty układ do pomiaru temperatury. Po przebadaniu tej konfiguracji w wymaganym zakresie temperatur otrzymano wykres przedstawiony na Rys. 2.5. Można wywnioskować, że zostały uzyskane bardzo dobre wyniki w zakresie $-20^{\circ}\text{C} - +20^{\circ}\text{C}$. W tym zakresie błędy nie przekraczają $\pm 1^{\circ}\text{C}$.



RYSUNEK 2.4. Schemat połączenia termistora NTC 10K w mostku Wheatstone'a



RYSUNEK 2.5. Przebieg napięcia wyjściowego układu w zależności od temperatury z włączonym termistorem NTC 10K w mostek Wheatstone’a

2.3.3. Cyfrowa aproksymacja charakterystyki termistora NTC

Cyfrowa aproksymacja charakterystyki termistora jest dobrą metodą pomiaru temperatury dla większych zakresów. Niestety zajmuje ona dużo pamięci FLASH w mikroprocesorze, co uniemożliwia wykorzystanie mikrokontrolera do innych funkcji. Dodatkowo wymaga ona również szeregowego połączenia termistora z rezystorem. Schemat takiego układu przedstawiono w rozdziale 2.3.1 na Rys. 2.2. W przypadku tej metody algorytm pomiaru temperatury jest następujący [4]:

- Po podłączeniu napięcia wyjściowego układu do przetwornika analogowo cyfrowego mikroprocesora należy estymować wartość rezystancji termistora na podstawie następującej zależności:

$$(5) \quad R_T = \left(\frac{U_{ADC}}{U_{REF} - U_{ADC}} \right) R_0$$

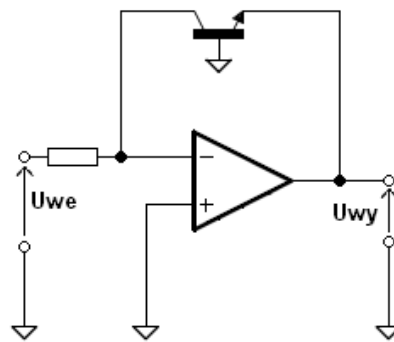
- Kolejnym krokiem jest obliczenie logarytmu według wzoru:

$$(6) \quad y = \ln(R_T)$$

- Ostatnim krokiem, który należy wykonać w każdej z metod jest aproksymacja uzyskanej charakterystyki za pomocą funkcji liniowej, bądź wielomianu.

2.3.4. Analogowa linearyzacja charakterystyki termistora NTC

Analogowa linearyzacja charakterystyki termistora NTC jest metodą optymalną. Pozwala na pomiar temperatury w dużym zakresie, równocześnie nie obciąża pamięci procesora. W porównaniu do metody cyfrowej, funkcję logarytmowania przedstawionej w równaniu (6) pełni wzmacniacz operacyjny w układzie logarytmującym. Schemat takiego układu ilustruje Rys. 2.6. Razem ze wzmacniaczem operacyjnym w układzie logarytmującym należy zastosować wzmacniacz operacyjny w układzie odwracającym, sumującym bądź odejmującym, ponieważ jego napięcie wyjściowe jest przeciwnego znaku do napięcia wejściowego. Czynność ta będzie niepotrzebna w przypadku zastosowania przetwornika analogowo cyfrowego działającego w zakresie zawierającym zakres napięcia wyjściowego z układu logarytmującego. Zastosowanie dodatkowego wzmacniacza operacyjnego, który dopasuje napięcie wyjściowe do całego zakresu działania przetwornika analogowo cyfrowego, spowoduje uzyskanie dużo lepszej dokładności pomiaru temperatury.



RYSUNEK 2.6. Wzmacniacz operacyjny w układzie logarytmującym

Rozdział 3

Projekt przetwornika temperatury na napięcie

Ze względu na wymagany szeroki zakres pomiaru temperatury oraz chęć zaoszczędzenia pamięci mikroprocesora, zdecydowano się wykonać analogowy przetwornik linearyzujący charakterystykę termistora NTC. Przetwornik składa się z dwóch wzmacniaczy operacyjnych. Jeden z nich działa w układzie logarytmującym i pełni zadanie linearyzacji wykładniczej charakterystyki termistora NTC, drugi zaś działa w układzie sumującym, który odpowiednio dopasowuje zakres napięcia wyjściowego z układu logarytmującego do zakresu działania wykorzystywanego przetwornika analogowo cyfrowego.

3.1. Wzmacniacz operacyjny w układzie logarytmującym

Wzmacniacz logarytmujący można zbudować poprzez połączenie wzmacniacza operacyjnego z elementem o charakterystyce wykładniczej. Jako element nieliniowy stosuje się diody półprzewodnikowe, bądź tranzystory bipolarne. Elementem najczęściej stosowanym jest tranzystor bipolarny. Schemat takiego układu został przedstawiony na rys. 2.6. Napięcie wyjściowe tego układu, w zależności od prądu wejściowego, opisane jest następującym wzorem [5, 6]:

$$(7) \quad U_{WY} = -\frac{kT}{q} \ln \left(\frac{I_{WE}}{I_{E0}} \right)$$

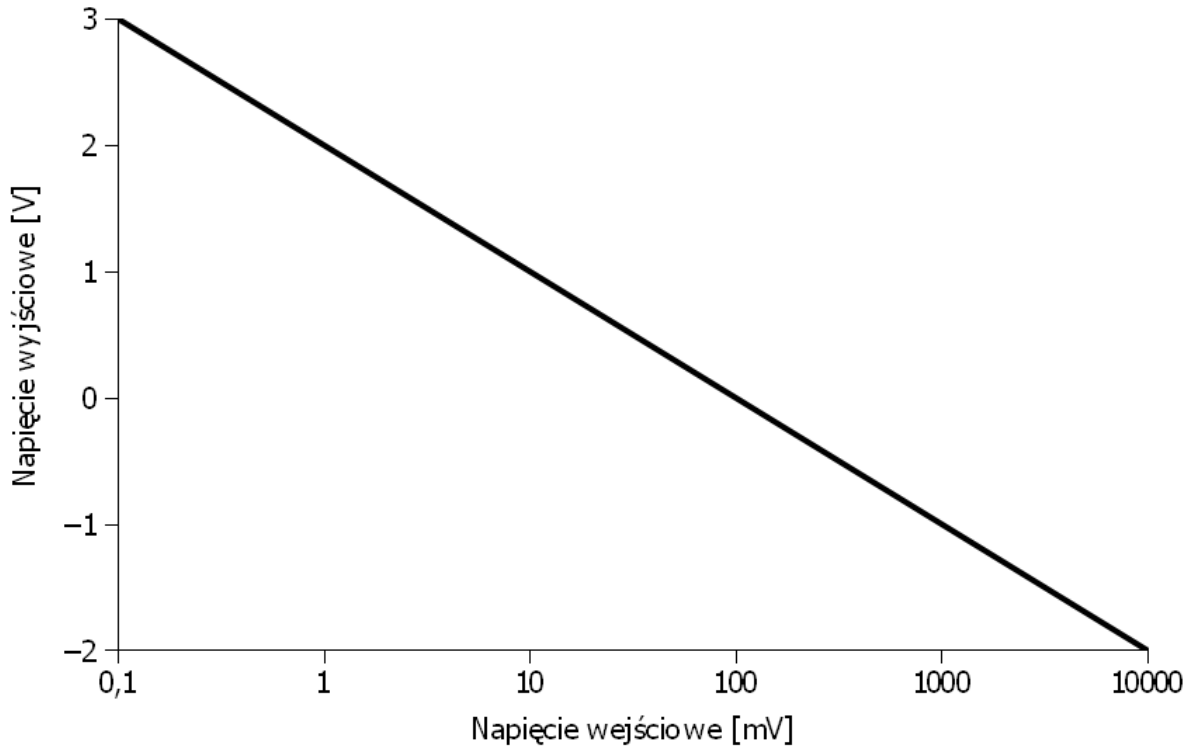
gdzie: U_{WY} - napięcie wyjściowe układu logarytmującego, k - stała Boltzmanna, T - temperatura bezwzględna, q - ładunek elektronu, I_{WE} - prąd wejściowy, I_{E0} - prąd wsteczny złącza emiterowego.

Ponieważ:

$$(8) \quad I_{WE} = \frac{U_{WE}}{R}$$

po przekształceniu wzoru (7) otrzymuje się następujący wzór przedstawiający logarytmiczną zależność napięcia wyjściowego układu od napięcia wejściowego:

$$(9) \quad U_{WY} = -\frac{kT}{q} \ln \left(\frac{U_{WE}}{RI_{E0}} \right)$$



RYSUNEK 3.1. Napięcie wyjściowe w zależności od napięcia wejściowego układu logarytmującego

3.2. Wzmacniacz operacyjny w układzie sumującym

Za pomocą wzmacniacza operacyjnego można łatwo realizować sumowanie napięć poprzez zastosowanie układu sumującego przedstawionego na Rys. 3.2. Zależność napięcia wyjściowego od napięć wejściowych można zapisać następująco [5, 6]:

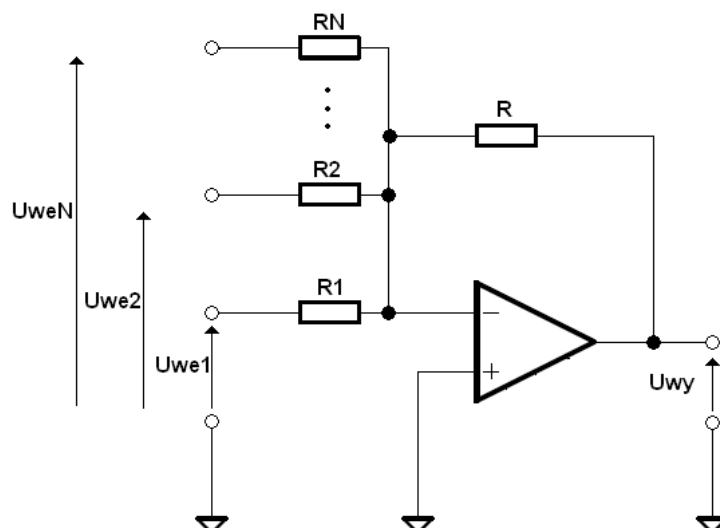
$$(10) \quad U = -R \left(\frac{U_{WE1}}{R_1} + \frac{U_{WE2}}{R_2} + \dots + \frac{U_{WE n}}{R_n} \right)$$

Analizując zależność nr (10) można zauważyć, iż w przypadku założenia:

$$(11) \quad R_1 = R_2 = \dots = R_n = R$$

układ realizuje algebraiczne sumowanie napięć.

W przypadku doboru rezystancji $R > R_1, R_2, \dots$ układ dodatkowo wzmacnia zsumowane napięcie, co zostało wykorzystane do budowy przetwornika temperatury na napięcie.

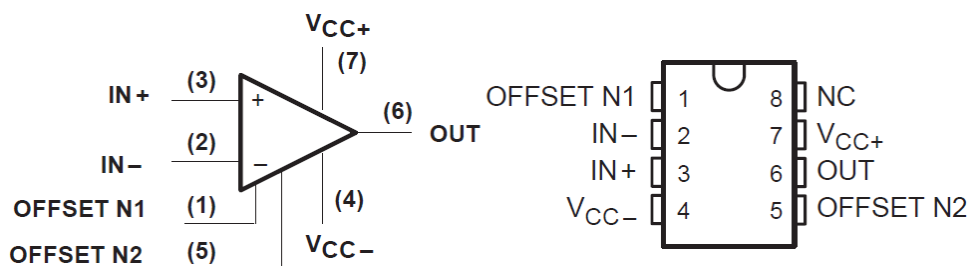


RYSUNEK 3.2. Wzmacniacz operacyjny w układzie sumującym

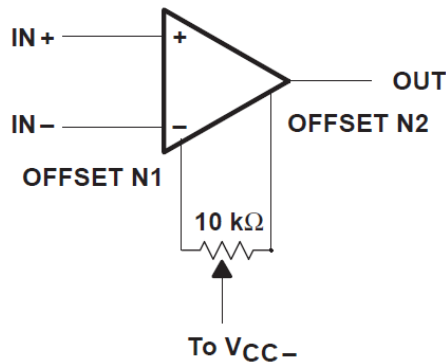
3.3. Budowa przetwornika temperatury na napięcie

3.3.1. Wzmacniacz operacyjny $\mu A741$

Do zbudowania przetwornika temperatury na napięcie został wykorzystany wzmacniacz operacyjny $\mu A741$. Jest to bardzo popularny i stosunkowo tani wzmacniacz o parametrach wystarczających do wykorzystania w przedmiotowym przetworniku. Opis wejść/wyjść wzmacniacza przedstawia Rys. 3.3 [7].

RYSUNEK 3.3. Wzmacniacz operacyjny $\mu A741$

Według noty aplikacyjnej wykorzystywanego wzmacniacza w celu kompensacji wyjściowego napięcia niezrównoważenia należy połączyć wyjścia 1 i 5 wzmacniacza (offset) za pomocą potencjometru z ujemnym zasilaniem według Rys. 3.4

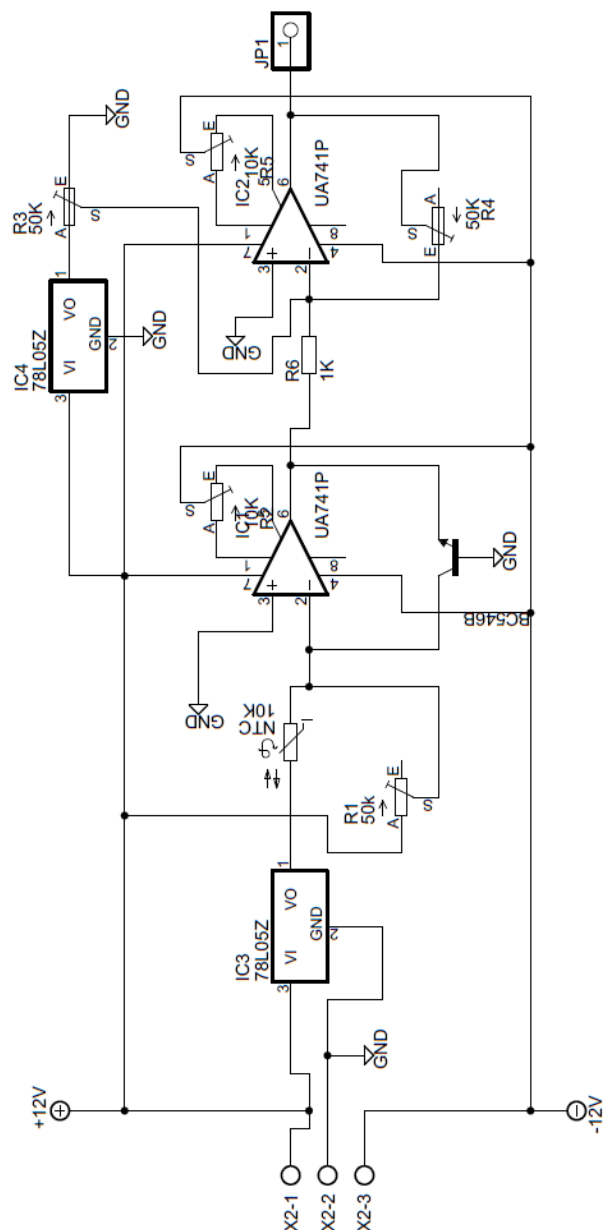


RYSUNEK 3.4. Sposób połączenia umożliwiający kompensację wejściowego napięcia niezrównoważenia wzmacniacza $\mu A741$

3.3.2. Schemat układu

Rysunek 3.5 ilustruje schemat wykonanego przetwornika. Układ składa się z dwóch stabilizatorów napięcia 78L05, dwóch wzmacniaczy operacyjnych $\mu A741$, trzech potencjometrów $50k\Omega$, dwóch potencjometrów $10k\Omega$, pojedynczego rezystora $1k\Omega$ i tranzystora npn BC546B. Cały układ zasilany jest zasilaczem symetrycznym $\pm 12V$. Takiego napięcia wymagają tylko i wyłącznie wzmacniacze operacyjne, zaś dzięki stabilizatorom napięcia termistor NTC zasilany jest napięciem równym $5V$. Ze względu na to, iż układ logarytmujący działa w odpowiednim zakresie prądu wejściowego, zaraz za termistorem podłączono ścieżkę doprowadzającą prąd, co zgodnie z pierwszym prawem Kirchhoffa powoduje dodanie stałej wartości do prądu płynącego z termistora NTC. Prąd ten można regulować za pomocą potencjometru $R1$. Następnie układ logarytmujący linearyzuje wykładniczą charakterystykę termistora NTC. Przetwornik temperatury na napięcie ma jeszcze jedną funkcję - dopasowanie napięcia wyjściowego z układu logarytmującego do zakresu przetwornika analogowo cyfrowego mikrokontrolera. W przedmiotowym przypadku zakres ten wynosi $0-5V$. Z tego powodu należy dodać odpowiednią wartość napięcia do napięcia wyjściowego i rozszerzyć zakres poprzez wzmocnienie napięcia. Taką funkcję w przetworniku pełni układ sumujący. Potencjometr $R3$ reguluje napięcie doprowadzane do tego układu. Należy go w taki sposób ustawić, aby przy najniższej temperaturze doprowadzany prąd kompensował prąd wychodzący z układu logarytmującego. W układzie sumującym rezystor w sprzężeniu zwrotnym zastąpiony został potencjometrem $R4$. Dzięki temu zachowano możliwość regulacji wzmocnienia. Pozostałe dwa potencjometry służą do kompensacji wejściowego napięcia niezrównoważenia wzmacniacza. Zastosowano tak dużą ilość elementów regulacyjnych z powodu tego, iż termistory różnych producentów różnią się parametrami.

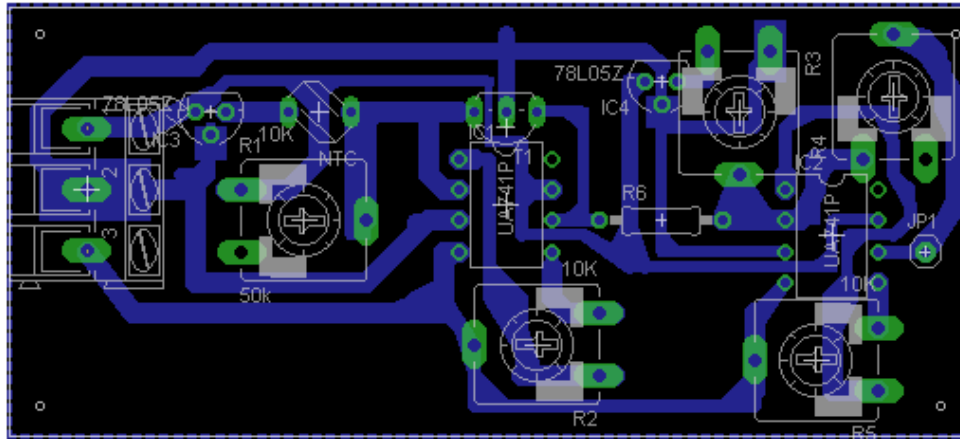
Dzięki potencjometrom można skalibrować przetwornik do działania z każdym termistorem NTC 10K.



RYSUNEK 3.5. Schemat wykonanego przetwornika temperatury na napięcie

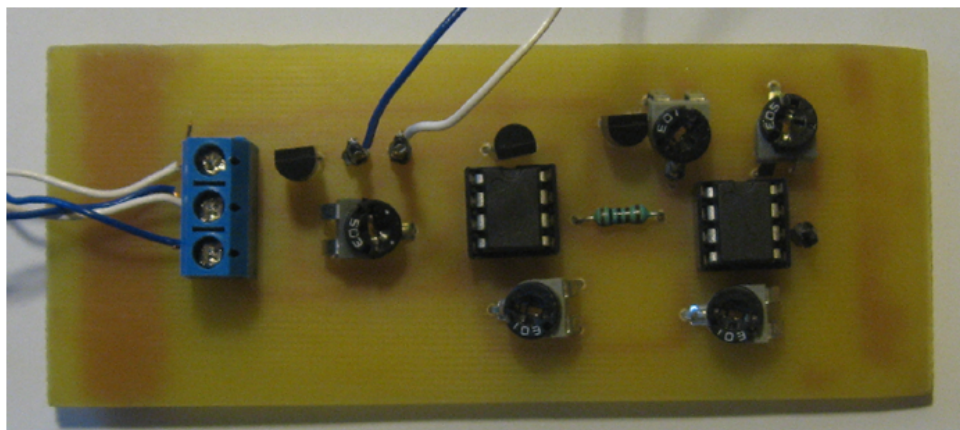
3.3.3. Projekt płytki PCB i wykonanie

Na Rys. 3.6 przedstawiono projekt płytki PCB, który został wykonany w programie Eagle. Rys. 3.7 ukazuje wykonany i działający przetwornik temperatury na napięcie.



RYSUNEK 3.6. Projekt wykonanego przetwornika temperatury na napięcie

Płytką PCB została wykonana metodą termotransferu druku laserowego na miedź pokrywającą laminat. Następnie została wytrawiona za pomocą nadsiarczuanu sodu B327.



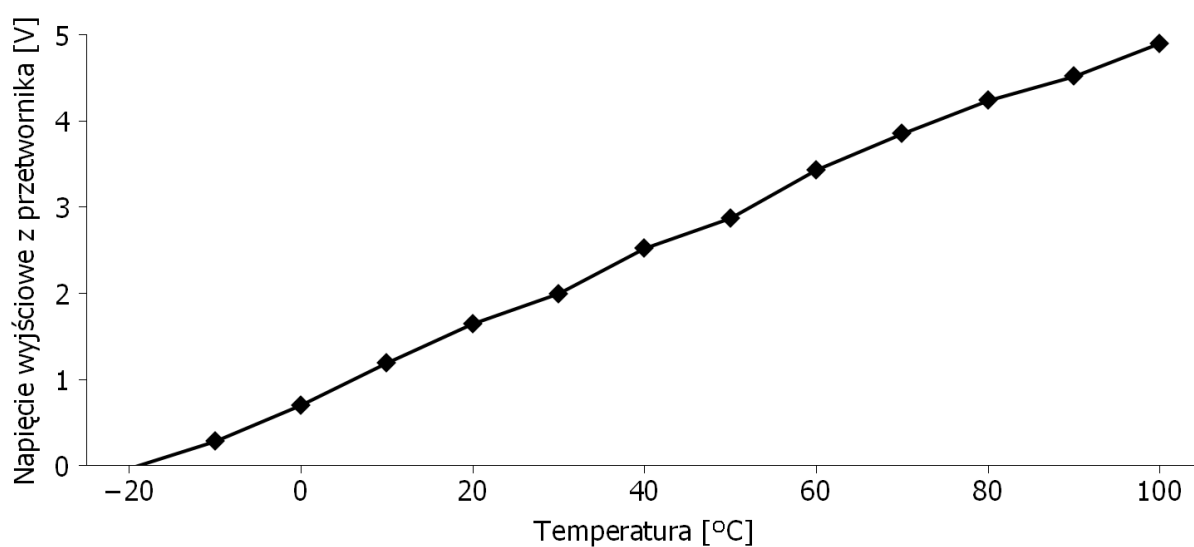
RYSUNEK 3.7. Wykonany przetwornik temperatury na napięcie

3.4. Otrzymane wyniki

Po przebadaniu przetwornika otrzymano wyniki przedstawione w tabeli 3.1 oraz na Rys. 3.8.

TABLICA 3.1. Wyniki badania przetwornika temperatury na napięcie

Temperatura [°C]	Wynik pomiaru [V]
-20	-0,035
-10	0,280
0	0,700
10	1,190
20	1,645
30	1,995
40	2,520
50	2,870
60	3,430
70	3,850
80	4,235
90	4,515
100	4,900



RYSUNEK 3.8. Wykres napięcia wyjściowego w zależności od temperatury wykonanego przetwornika

Po aproksymacji uzyskanej charakterystyki za pomocą funkcji liniowej otrzymano błędy zamieszczone w tabeli 3.2.

TABLICA 3.2. Błędy pomiarów uzyskane podczas badania przetwornika

Temperatura [°C]	Błąd pomiaru [°C]
-20	-1,1
-10	1,5
0	1,7
10	0,1
20	-0,5
30	1,3
40	-1,1
50	0,7
60	-2,5
70	-2,3
80	-1,4
90	2,1
100	3,1

Z tabeli 3.2 wynika, iż przetwornik najlepszą dokładność zachował w zakresie 10–30°C. Jest to efekt zadowalający, ponieważ w tym zakresie przetwornik będzie pracował w pomieszczeniach mieszkalnych, gdzie temperatura $\pm 1^\circ\text{C}$ jest odczuwalna. W zakresie 50–100°C przetwornik wykorzystywany będzie do pomiaru temperatury pieca bądź urządzeń grzewczych. Dla takich pomiarów dokładność $\pm 2\text{--}3^\circ\text{C}$ jest do przyjęcia.

Rozdział 4

Projekt układu mikroprocesorowego

4.1. Wykorzystany mikrokontroler

Głównym elementem układu mikroprocesorowego jest mikrokontroler ATmega8 firmy Atmel. Przy doborze mikrokontrolera kierowano się głównie kryteriami takimi jak liczba wejść/wyjść, liczba kanałów przetwornika analogowo-cyfrowego, rodzaj interfejsu komunikacyjnego, wielkość pamięci, dostępność oraz koszt mikrokontrolera. Najważniejsze parametry mikrokontrolera to [8]:

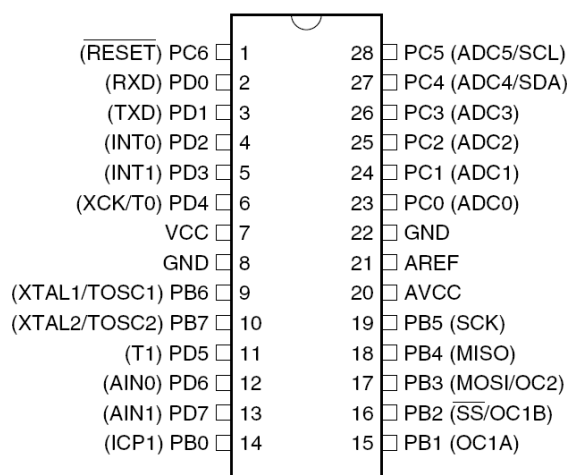
- 8kB wewnętrznej pamięci programu FLASH
- 512 B wewnętrznej pamięci danych EEPROM
- 1kB wewnętrznej pamięci SRAM
- 23 programowalne wejścia/wyjścia
- dwa 8 bit czasomierze/liczniki
- jeden 16bit czasomierz/licznik
- 6 kanałowy 10-bitowy przetwornik analogowo-cyfrowy
- programowalny interfejs komunikacji szeregowej USART
- interfejs SPI

ATmega8 (Rys.4.1) wyposażona jest w 28 wyprowadzeń, których większość może pełnić różne role, min.: wyjścia/wejścia cyfrowe, wejścia sygnałów przerwań, lub wejścia przetwornika ADC.

Wyprowadzenia wymagane do prawidłowego działania mikrokontrolera o pojedynczym zastosowaniu to:

- *VCC* - zasilanie 5V
- *GND* - masa
- *AREF* - wejście napięcia odniesienia dla przetwornika analogowo-cyfrowego
- *AVCC* - zasilanie przetwornika analogowo-cyfrowego

Pozostałe wyprowadzenia podzielone są na porty: PORTB, PORTC i PORTD. Mogą one działać jako standardowe wejścia/wyjścia. Dodatkowo każde z wyprowadzeń posiada



RYSUNEK 4.1. Wykorzystany mikrokontroler

swoją funkcję alternatywną. Funkcje alternatywne dla poszczególnych portów są następujące:

(1) PORTB

PB0(ICP1): Wejście sygnału przechwytywania stanu liczników

PB1(OC1A): Wyjście do porównań liczników

PB2(OC1B): Wyjście do porównań liczników

PB3(OC2): Wyjście do porównań liczników

PB4(MISO): Wyprowadzenie modułu SPI

PB5(SCK): Wyprowadzenie modułu SPI

PB6(XTAL1/TOSC1): Źródło zewnętrznego sygnału taktującego, sygnału taktującego zegar lub wyprowadzenie do podłączenia rezonatora kwarcowego

PB7(XTAL2/TOSC2): Źródło zewnętrznego sygnału taktującego, sygnału taktującego zegar lub wyprowadzenie do podłączenia rezonatora kwarcowego

(2) PORTC

PC0(ADC0): Wejście kanału przetwornika analogowo-cyfrowego

PC1(ADC1): Wejście kanału przetwornika analogowo-cyfrowego

PC2(ADC2): Wejście kanału przetwornika analogowo-cyfrowego

PC3(ADC3): Wejście kanału przetwornika analogowo-cyfrowego

PC4(ADC4/SDA): Wejście kanału przetwornika analogowo-cyfrowego lub wyprowadzenie modułu TWI (interfejs I2C)

PC5(ADC5/SCL): Wejście kanału przetwornika analogowo-cyfrowego lub wyprowadzenie modułu TWI (interfejs I2C)

PC6(*RESET*): Zewnętrzne źródło zerujące

(3) PORTD

PD0(*RxD*): Linia modułu USART - tor danych odbieranych

PD1(*TxD*): Linia modułu USART - tor danych nadawanych

PD2(*INT0*): Wejście przerwania zewnętrznego

PD3(*INT1*): Wejście przerwania zewnętrznego

PD4(*XCK/T0*): Wejście sygnału taktującego dla modułu USART lub wejście zliczające - zegarowe licznika TC0

PD5(*T1*): Wejście zliczające - zegarowe licznika TC1

PD6(*AIN0*): Wejście nieodwracające (+) komparatora analogowego

PD7(*AIN1*): Wejście odwracające (-) komparatora analogowego

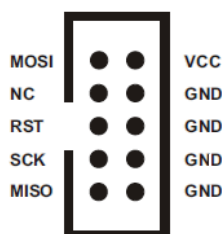
4.2. Układ pracy mikrokontrolera

Do prawidłowego działania mikrokontrolera wymagane jest wykonanie szeregu połączeń odpowiednich wyprowadzeń. Są to połączenia zasilania *VCC*, *AVCC* oraz napięcia odniesienia przetwornika analogowo-cyfrowego *AREF* za pomocą kondensatorów 100nF z masą. Połączenia te stosuje się w celu odklócenia napięcia zasilającego. Dodatkowo dokumentacja mikrokontrolera ATmega8 podaje, że wyprowadzenie *RESET* należy zewrzeć za pomocą rezystora 10k Ω z zasilaniem układu. Bardzo przydatnym rozwiązaniem konstrukcyjnym stosowanym w mikrokontrolerach jest ISP (In-System Programming). Umożliwia ono programowanie układu nawet już wewnątrz zbudowanego urządzenia. Główną zaletą, wynikającą ze stosowania ISP, jest możliwość połączenia programowania i testowania w jednej fazie, co sprzyja szybszemu projektowaniu urządzenia, ułatwia modyfikacje oraz aktualizacje oprogramowania. Z tego względu do podstawowego układu pracy zalicza się nie tylko połączenia odpowiednich wyprowadzeń mikrokontrolera, ale również gniazdo do programowania ISP przedstawione na rys. 4.2.

Rys. 4.3 przedstawia schemat gotowego do pracy układu wraz ze złączem *ISP*.

TABLICA 4.1. Opis wyprowadzeń interfejsu ISP

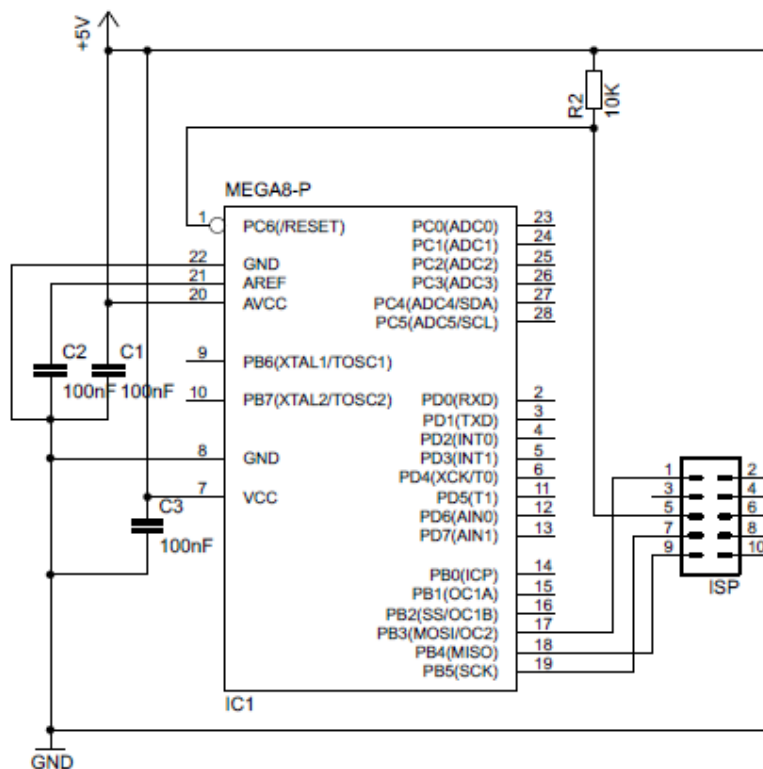
Wyprowadzenie <i>ISP</i>	Wyprowadzenie mikrokontrolera	Opis
<i>RST</i>	PC6	Reset układu, wymuszenie programowania
<i>SCK</i>	PB5	Sygnał zegarowy
<i>MISO</i>	PB4	Wyjście danych
<i>MOSI</i>	PB3	Wejście danych
<i>GND</i>		Masa
<i>VCC</i>		Zasilanie układu



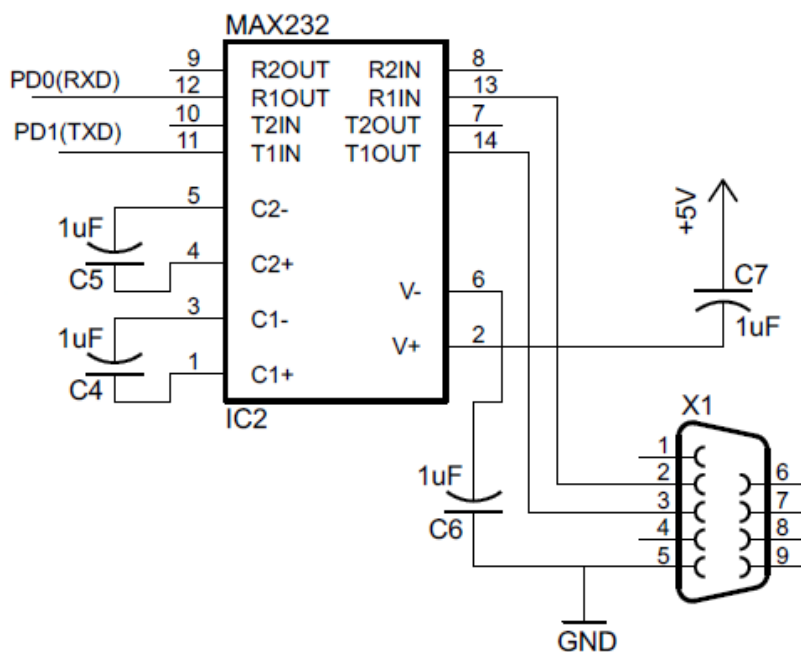
RYSUNEK 4.2. Złącze ISP

4.3. Komunikacja mikrokontrolera z komputerem

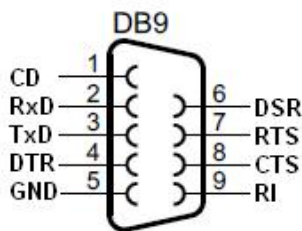
Najczęściej wykorzystywanym interfejsem do komunikacji pomiędzy mikroprocesorem a komputerem jest interfejs RS232. Do obsługi tego interfejsu wykorzystany mikrokontroler posiada moduł UART. Niestety bezpośrednie połączenie modułu UART z portem COM komputera klasy PC nie przyniesie oczekiwanych efektów. Trzeba wziąć pod uwagę, że w liniach interfejsu RS232 występują inne napięcia niż typowe dla standardu TTL. W przypadku niskonapięciowych mikrokontrolerów logiczne „0” zdefiniowane jest jako napięcie od 0V do +0,8V w odniesieniu do masy, a logiczna „1” jako napięcie +2V do +5V, zaś w przypadku standardu RS232 specyfikacja napięcia definiuje logiczną „1” jako napięcie -3V do -15V, zaś „0” jako napięcie +3V do +15V. W związku z tym, żeby uruchomić komunikację między mikroprocesorem a komputerem niezbędny jest konwerter napięć ze standardu TTL na RS232. Najbardziej popularnym scalonym konwerterem jest układ MAX232 i jego odpowiedniki, które mają wbudowaną ładunkową przetwornicę napięcia. Na rys. 4.4 przedstawiono schemat typowego układu konwertera $TTL \leftrightarrow RS232$.



RYSUNEK 4.3. Wymagane połączenia odpowiednich wyprowadzeń do prawidłowego działania mikrokontrolera



RYSUNEK 4.4. Schemat typowego konwertera $TTL \leftrightarrow RS232$ z wykorzystaniem MAX232



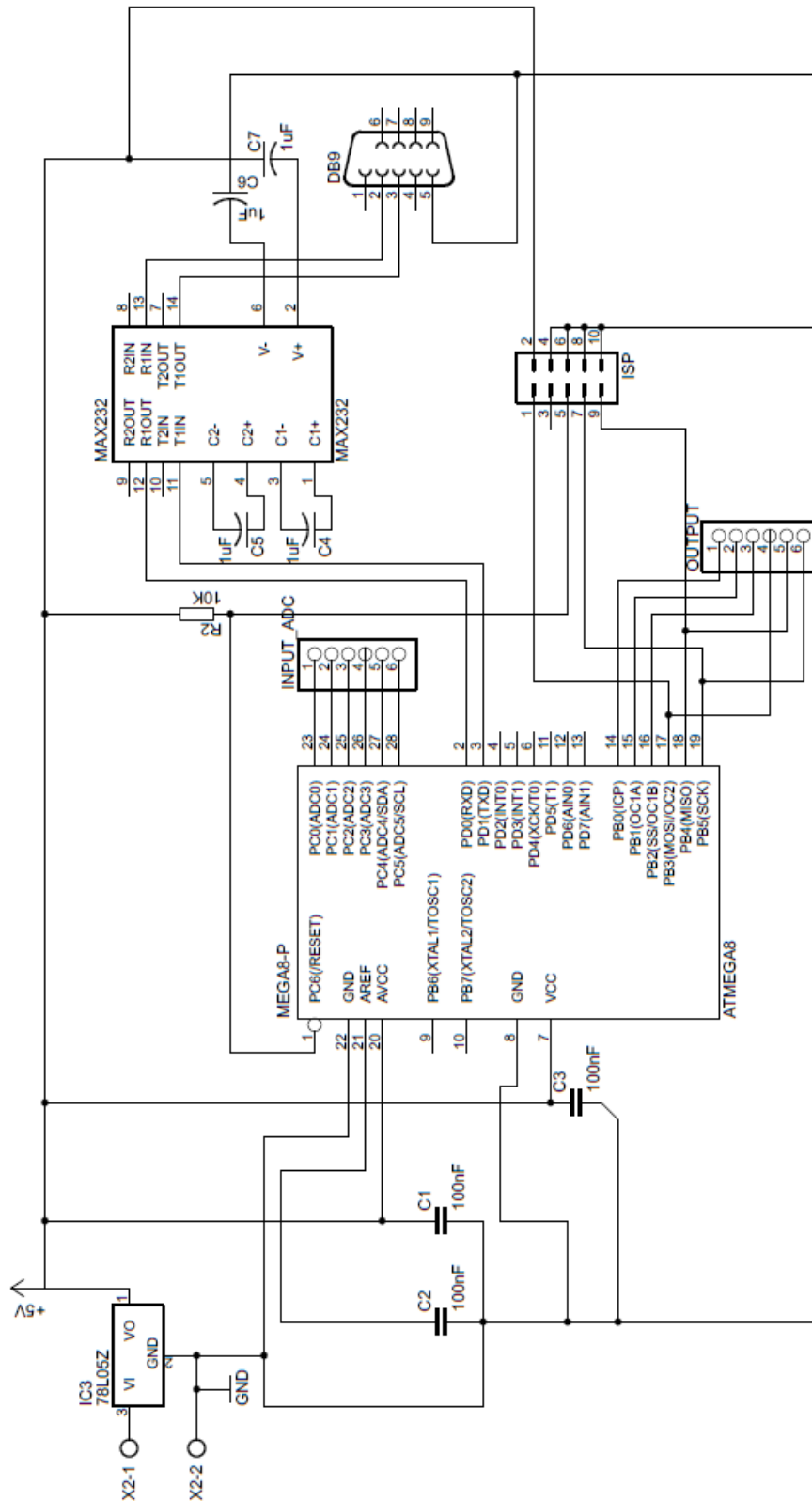
RYSUNEK 4.5. Złącze DB9

Przyglądając się rysunkom 4.4 oraz 4.5 należy zauważyć, że wyprowadzenia z układu MAX232 oznaczone TxD (*TransmitData* - dane nadawane) i RxD (*ReceiveData* - dane odebrane) połączone są ze złączem DB9 „na krzyż”. Oznacza to, iż do prawidłowej komunikacji wymagane jest, aby TxD mikrokontrolera połączyć z RxD złącza DB9, zaś RxD mikrokontrolera z TxD złącza DB9.

4.4. Budowa układu mikroprocesorowego

4.4.1. Schemat układu

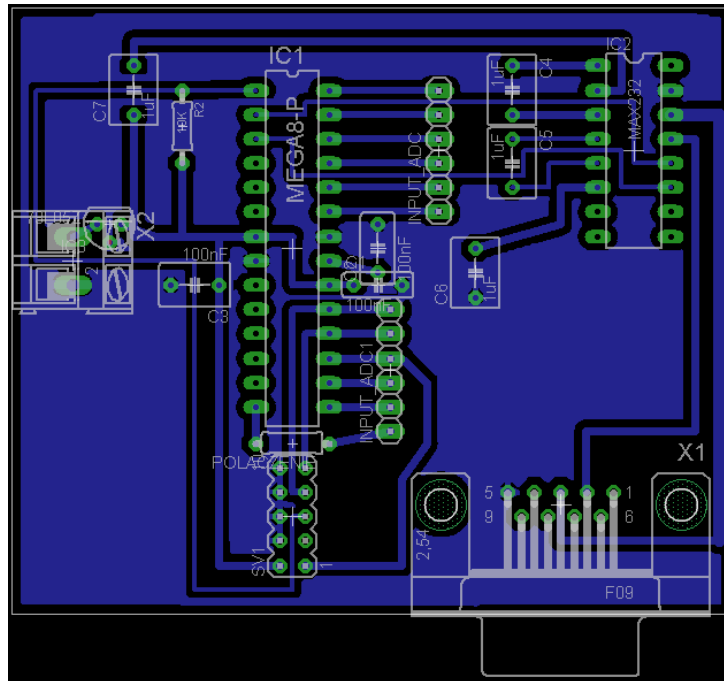
Na Rys. 4.6 przedstawiono schemat wykonanego układu mikroprocesorowego. Układ zawiera mikroprocesor ATmega8 ze złączem ISP i wymaganymi połączeniami zawartymi w rozdziale 4.2 oraz układ MAX232 wraz ze złączami opisany w rozdziale 4.3. Dodatkowo z portu *PORTC* oraz *PORTB* zostały wyprowadzone dwie listwy sześciopinowe do podłączenia wejść i wyjść. Wyprowadzenia z portu *PORTC* pełnią funkcje wejść przetwornika analogowo cyfrowego, dlatego zostały opisane jako INPUT_ADC, zaś *PORTB* oprogramowano jako dwustanowe wyjścia (OUTPUT). Do zasilania układu mikroprocesorowego oraz przetworników temperatury na napięcie został wykorzystany ten sam zasilacz. Ze względu na to, iż przetworniki wymagają napięcia $\pm 12V$, a mikroprocesor 5V, układ mikroprocesorowy wyposażony jest w stabilizator napięcia 78L05.



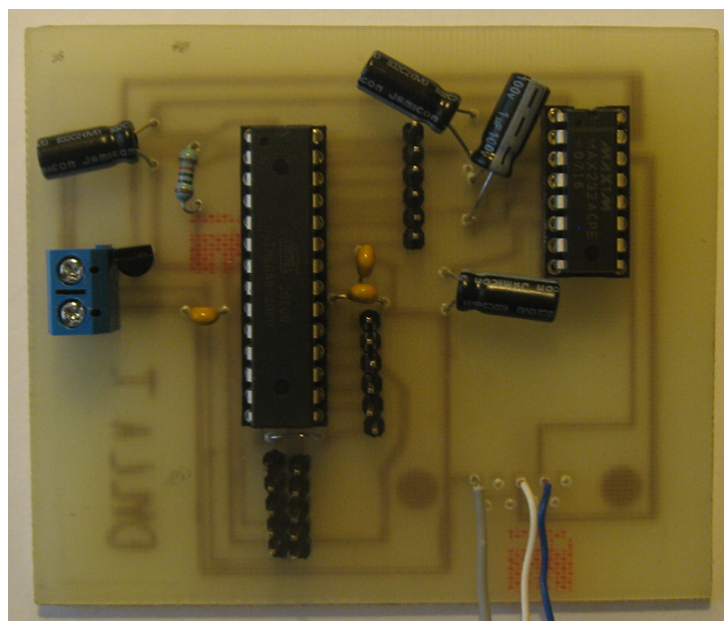
RYSUNEK 4.6. Schemat wykonanego układu mikroprocesorowego

4.4.2. Projekt płytki PCB i wykonanie

Na rys. 4.7 przedstawiono projekt płytki PCB, który został wykonany w programie Eagle. Rys. 4.8 ukazuje samodzielnie wykonany i działający układ mikroprocesorowy.



RYSUNEK 4.7. Projekt wykonanego układu mikroprocesorowego



RYSUNEK 4.8. Wykonany układ mikroprocesorowy

4.5. Oprogramowanie

Oprogramowanie dla mikrokontrolera ATmega8 można napisać w językach : C, BASCOM lub assembler. W przedmiotowym projekcie został zastosowany język C. W porównaniu do pozostałych jest najbardziej popularnym i rozpowszechnionym językiem wysokiego poziomu, stosowanym do programowania prostych mikroprocesorów jak i skomplikowanych superkomputerów. Dodatkowo jest bardzo czytelny i uniwersalny, ponieważ nie wymaga ukierunkowania się na konkretny sprzęt. Do oprogramowania mikrokontrolera w wybranym języku zostało zastosowane darmowe środowisko WinAVR. Zawiera ono kompilator języka C wraz z kompletem narzędzi do tworzenia oprogramowania dla 8-bitowych mikroprocesorów AVR. W celu łatwiejszej analizy programu, napisany kod można podzielić na dwie części: obsługa przetwornika analogowo cyfrowego oraz obsługa komunikacji szeregowej USART.

4.5.1. Przetwornik analogowo cyfrowy

4.5.1.1. Wprowadzenie. Mikrokontroler ATmega8 wyposażony jest w sześciokanałowy, dziesięciobitowy przetwornik analogowo cyfrowy. Pomiar wartości napięcia względem masy określa się według następującego wzoru [8]:

$$(12) \quad ADC = \frac{V_{IN} \cdot 1024}{V_{REF}}$$

gdzie: ADC - rejestr przechowujący wynik pomiaru, V_{IN} - mierzone napięcie wejściowe, V_{REF} - programowalne napięcie odniesienia przetwornika analogowo cyfrowego.

Przed wykorzystaniem przetwornika należy go odpowiednio skonfigurować poprzez ustawienie odpowiednich bitów w rejestrze $ADMUX$ oraz $ADCSRA$.

7	6	5	4	3	2	1	0	
REFS1	REFS0	ADLAR	–	MUX3	MUX2	MUX1	MUX0	ADMUX

RYSUNEK 4.9. Rejestr $ADMUX$ mikroprocesora ATmega8

Bit y przedstawionego na Rys. 4.9 rejestru $ADMUX$ określają:

- $REFS1$, $REFS2$ – źródło napięcia odniesienia

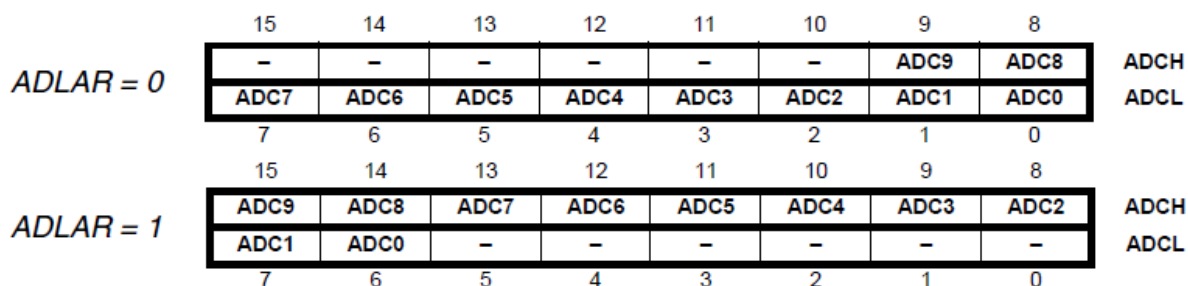
Napięcie odniesienia wewnętrzne równe 2,56 V

TABLICA 4.2. Wybór napięcia odniesienia za pomocą rejestru *ADMUX*

<i>REFS1</i>	<i>REFS2</i>	Napięcie odniesienia
0	0	zewnętrzne, podane na wyprowadzenie <i>AREF</i>
0	1	napięcie zasilania
1	0	wartość zarezerwowana
1	1	wewnętrzne równe 2,56 V

- *ADLAR* – sposób prezentacji wyników

Rejestr przechowujący wynik pomiaru składa się z dwóch bajtów - *ADCH* oraz *ADCL*. Rejestr *ADLAR* określa, czy wynik zostanie wyrównany do lewej strony, co oznacza, iż 8 bitów znajdzie się w bajcie *ADCH* i 2 najmłodsze bity w bajcie *ADCL*, bądź do prawej strony. W przypadku wyrównania do prawej strony bajt *ADCH* będzie zawierał 2 najstarsze bity, zaś bajt *ADCL* pozostałe 8 bitów. Rys 4.10 ilustruje rozmieszczenie wyników w zależności od stanu bitu *ADLAR*.

RYSUNEK 4.10. Rozmieszczenie wyników pomiaru w zależności od stanu bitu *ADLAR*

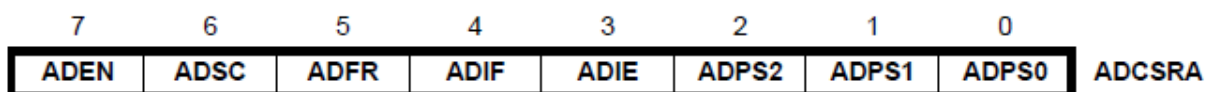
- *MUX3...0* – wybór kanału przetwornika

Ustawienie odpowiednich bitów *MUX* powoduje wybór odpowiedniego kanału zgodnie z tabelą 4.3.

TABLICA 4.3. Wybór kanału przetwornika analogowo cyfrowego za pomocą rejestru *ADMUX*

<i>MUX3...0</i>	Kanał
0000	<i>ADC0</i>
0001	<i>ADC1</i>
0010	<i>ADC2</i>
0011	<i>ADC3</i>
0100	<i>ADC4</i>
0101	<i>ADC5</i>
0110	<i>ADC6</i>
0111	<i>ADC7</i>

Kolejnym ważnym rejestrem do skonfigurowania przetwornika analogowo cyfrowego jest przedstawiony na Rys. 4.11 rejestr *ADCSRA*.



RYSUNEK 4.11. Rejestr *ADCSRA* mikroprocesora ATmega8

Opis odpowiednich bitów rejestru został przedstawiony w tabeli 4.4. Na bitach *ADPS2...0* ustawia się zgodnie z tabelą 4.5 współczynnik podziału, który odpowiednio dzieli częstotliwość zegaru systemowego. Następnie podzieloną częstotliwością jest taktowany przetwornik analogowo cyfrowy.

TABLICA 4.4. Opis bitów rejestru *ADCSRA*

Bity rejestru <i>ADCSRA</i>	Opis
<i>ADEN</i>	włączenie przetwornika
<i>ADSC</i>	start przetwarzania
<i>ADATE</i>	włączenie automatycznego trybu wyzwalania
<i>ADIF</i>	znacznik przerwania, ustawiany gdy przetwarzanie jest zakończone
<i>ADIE</i>	włączenie przerwania
<i>ADPS2...0</i>	ustawienie częstotliwości taktowania przetwornika

TABLICA 4.5. Wybór współczynnika podziału częstotliwości za pomocą rejestru *ADMUX*

<i>ADPS2...0</i>	Współczynnik podziału
000	2
001	2
010	4
011	8
100	16
101	32
110	64
111	128

4.5.1.2. Oprogramowanie przetwornika analogowo cyfrowego. W pierwszej kolejności po uruchomieniu mikroprocesora, inicjalizowany jest przetwornik. W funkcji inicjalizującej zawarte są komendy ustawiające odpowiednie bity w rejestrach *ADMUX* oraz *ADCSRA*. Wygląda to następująco:

```
// Napięcie odniesienia wewnętrzne równe 2,56 V
ADMUX |= _BV(REFS0);
// Wyrównanie wyniku do lewej
ADMUX |= _BV(ADLAR);
// Zezwolenie na konwersję
ADCSRA |= _BV(ADEN);
// Wybicie częstotliwości dla taktowania przetwornika
ADCSRA |= _BV(ADPS0);
ADCSRA |= _BV(ADPS1);
ADCSRA |= _BV(ADPS2);
```

Następnym krokiem jest ustawienie odpowiedniego kanału i wyzwolenie przetwarzania. Początkowa wartość w rejestrach *ADMUX* oraz *ADCSRA* wszystkich bitów jest równa 0, dlatego w pierwszym kroku przetwornik mierzy napięcie na kanale 0. Po zakończeniu pomiaru wynik zapisywany jest do zmiennej. Następnie wybierany jest kolejny kanał i realizowany jest analogicznie do kanału 0 pomiar. Poniżej przedstawiono kod tego procesu w języku C.

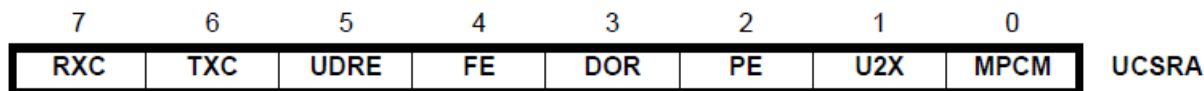
```
// Rozpoczęcie przetwarzania
    ADCSRA |= _BV(ADSC);
// Oczekiwanie na zakończenie przetwarzania
    while( bit_is_set (ADCSRA,ADSC))
        {};
// Zapisanie starszych 8 bitów wyniku konwersji
    pomiar1=ADCH;
```

```
// Wybranie kanału 1
    ADMUX |= _BV(MUX0);
// Rozpoczęcie przetwarzania
    ADCSRA |= _BV(ADSC);
// Oczekiwanie na zakończenie przetwarzania
    while( bit_is_set (ADCSRA,ADSC))
        {};
// Zapisanie starszych 8 bitów wyniku konwersji
    pomiar2=ADCH;
```

```
// Wybranie kanału 2
    ADMUX &= ~_BV(MUX0);
    ADMUX |= _BV(MUX1);
// Rozpoczęcie przetwarzania
    ADCSRA |= _BV(ADSC);
// Oczekiwanie na zakończenie przetwarzania
    while( bit_is_set (ADCSRA,ADSC))
        {};
// Zapisanie starszych 8 bitów wyniku konwersji
    pomiar3=ADCH;
```

4.5.2. Komunikacja USART

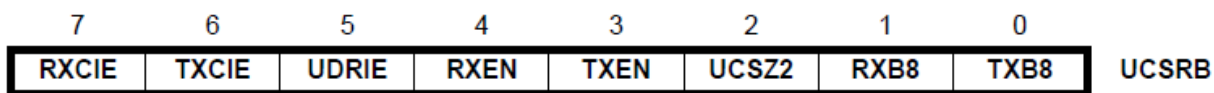
4.5.2.1. Wprowadzenie. USART (*Universal Synchronous and Asynchronous Receiver and Transmitter*) jest to uniwersalny układ transmisji szeregowej mogący pracować synchronicznie i asynchronicznie. Praca synchroniczna polega na przesyłaniu kolejnych bitów w takt zegara sterującego transmisją, zaś w trybie pracy asynchronicznej odbiornik USART wykrywa początek ramki transmisyjnej, po czym przez ustalony czas trwania bitu próbuje sygnal wejściowy. Podobnie jak przetwornik analogowo cyfrowy, moduł *USART* do poprawnego działania wymaga konfiguracji odpowiednich rejestrów. Są to trzy rejestry kontroli i statusu (*UCSRA*, *UCSRB*, *UCSRC*) oraz dwa rejestry szybkości transmisji USART (*UBRRL*, *UBRRH*) [8].



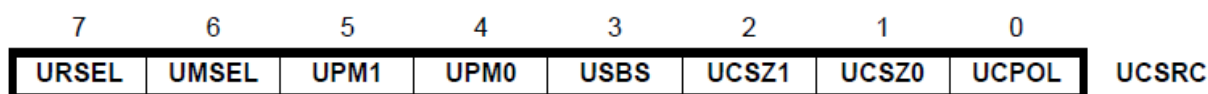
RYSUNEK 4.12. Rejestr *UCSRA* mikroprocesora ATmega8

TABLICA 4.6. Opis bitów rejestru *UCSRA*

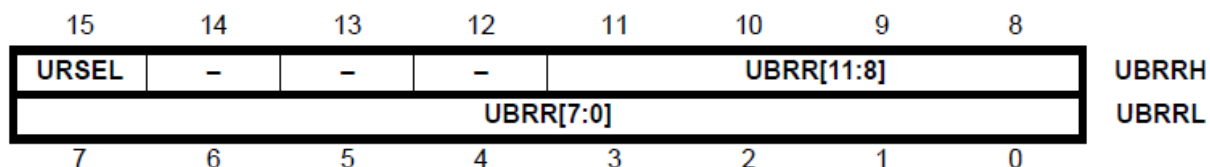
Bity rejestru <i>UCSRA</i>	Opis
<i>RXC</i>	zakończona transmisja USART
<i>TXC</i>	zakończone nadawanie USART
<i>UDRE</i>	pusty rejestr danych
<i>FE</i>	błąd ramki
<i>DOR</i>	przekroczenie danych
<i>UPE</i>	błąd parzystości
<i>U2X</i>	podwójna prędkość transmisji danych
<i>MPCM</i>	tryb komunikacji wieloprocessorowej

RYSUNEK 4.13. Rejestr *UCSRB* mikroprocesora ATmega8TABLICA 4.7. Opis bitów rejestru *UCSRB*

Bity rejestru <i>UCSRB</i>	Opis
<i>RXCIE</i>	włączenie przerwania RX
<i>TXCIE</i>	włączenie przerwania TX
<i>UDRIE</i>	włączenie przerwania zgłaszającego pusty rejestr danych
<i>RXEN</i>	włączenie odbiornika
<i>TXEN</i>	włączenie nadajnika
<i>UCSZ2</i>	rozmiar znaku(ilość bitów danych w ramce)
<i>RXB8</i>	8 bitów danych odbiornika
<i>TXB8</i>	8 bit danych nadajnika

RYSUNEK 4.14. Rejestr *UCSRC* mikroprocesora ATmega8TABLICA 4.8. Opis bitów rejestru *UCSRC*

Bity rejestru <i>UCSRC</i>	Opis
<i>URSEL</i>	wybór rejestru <i>UCSRC</i> lub <i>UBRRH</i>
<i>UMSEL</i>	wybór trybu synchronicznego lub asynchronicznego
<i>UPM1...0</i>	włączenie i ustawienie trybu sprawdzania parzystości
<i>USBS</i>	określenie liczby bitów stopu
<i>USZ1...0</i>	rozmiar znaku(ilość bitów danych w ramce)
<i>UCPOL</i>	polaryzacja zegara

RYSUNEK 4.15. Rejestr *UBRRH* i *UBRRL* mikroprocesora ATmega8TABLICA 4.9. Opis bitów rejestru *UBRRH* i *UBRRL*

Bit y rejestru <i>UBRRH</i> i <i>UBRRL</i>	Opis
<i>URSEL</i>	wybór rejestru <i>UCSRC</i> lub <i>UBRRH</i>
– – 14...12	bity zarezerwowane
<i>UBRR11...0</i>	rejestr szybkości transmisji

4.5.2.2. Oprogramowanie modułu *USART*. Oprogramowanie modułu *USART* składa się z 3 funkcji:

- Inicjalizacja,

Przed napisaniem funkcji inicjalizującej należy obliczyć wartość rejestru szybkości transmisji *UBRR* zgodnie z następującym wzorem:

$$(13) \quad UBRR = \frac{f_{osc}}{16BAUD} - 1$$

gdzie: *UBRR* - wartość rejestru szybkości transmisji, f_{osc} - częstotliwość zegara systemowego, *BAUD*- prędkość transmisji.

Najłatwiejszym i bardzo przejrzystym sposobem na obliczenie powyższego równania jest zdefiniowanie go na początku programu w następujący sposób.

```
//częstotliwość zegara systemowego
#define FOSC 1 000 000
//prędkość transmisji
#define BAUD 2400
//obliczenie współczynnika UBRR
#define UBRR FOSC/16/BAUD-1
```

Po napisaniu powyższego kodu można przejść do funkcji inicjalizującej. Tutaj przepisywana jest obliczona wartość *UBRR* do rejestru szybkości transmisji w

taki sposób, że cztery najstarsze bity wpisywane są do rejestru *UBRRH* zaś pozostałe osiem młodszych bitów do rejestru *UBRRL*. Następnie konfigurowany jest rejestr *UCSRB* oraz *UCSRC*. W rejestrze *UCSRB* ustawiane są bity odpowiedzialne za włączenie nadajnika, odbiornika oraz obsługi przerwań od odbiornika, zaś w rejestrze *UCSRC* określany jest format przesyłanej ramki. W przedmiotowym przypadku poniższy kod wskazuje na ustawienie 8 bitów danych i 2 bity stopu.

```
void USART_Init( unsigned int ubrr )
{
    //Ustawienie prędkości
    UBRRH = (unsigned char)(ubrr>>8);
    UBRRL = (unsigned char)ubrr;
    //Włączenie nadajnika, odbiornika, zezwolenie na przerwania
    UCSRB = (1<<RXEN)|(1<<TXEN)|(1<<RXCIE);
    //Ustawienie formatu ramki: 8 bitów danych, 2 bity stopu
    UCSRC = (1<<URSEL)|(1<<USBS)|(3<<UCSZ0);
}
```

- Wysyłanie znaków,

Funkcja wysyłająca znaki jest bardzo prosta. W pierwszym kroku oczekuje na pusty bufor nadajnika, co sygnalizuje bit *UDRE* w rejestrze *UCSRA*. Gdy bit ten ustawiony jest na 1, nadajnik gotowy jest do wysłania danych. Następnie dane wprowadzane są do bufora *UDR* i wysyłane.

```
void USART_Transmit( unsigned char data )
{
    //Oczekiwanie na pusty bufor
    while ( !( UCSRA & (1<<UDRE)) );
    //Umieszczenie danych w buforze i wysłanie
    UDR = data;
}
```

- Odbieranie znaków. Funkcja odbierania znaku oczekuje na ustawienie na 1 bitu *RXC* rejestru *UCSRA*. Oznacza to, że w buforze są nieprzeczytane dane. Gdy taki moment nastąpi, pobierane są i zwracane odebrane dane z bufora. Nasłuchiwanie i oczekiwanie na nadchodzące dane w każdym cyklu zegara jest niepotrzebne, dlatego w przedmiotowym przypadku funkcja ta wywoływana jest w przerwaniu. Dzięki temu mikroprocesor w każdym cyklu pracy tylko wysyła dane, zaś w momencie wystąpienia przerwania od odbiornika, blokowane jest wykonywanie programu i odbierane są dane.

```
unsigned char USART_Receive( void )
{
    //Oczekiwanie na dane przeznaczone do odebrania
    while ( !(UCSRA & (1<<RXC)) );
    // Pobranie i zwrócenie odebranych danych z bufora
    return UDR;
}
```


Rozdział 5

Zakończenie

5.1. Podsumowanie i wnioski

Wynikiem końcowym wykonanej pracy jest wielokanałowy regulator, umożliwiający pomiar temperatury za pomocą czujników NTC 10K w zakresie -20°C – $+100^{\circ}\text{C}$ oraz ustawianie odpowiednich wyjść, zgodnie z otrzymanym za pomocą magistrali RS232 sterowaniem, w celu regulacji temperatury w poszczególnych pomieszczeniach. W pracy zaprojektowano i wykonano przetwornik do pomiaru temperatury oraz zaprojektowano, wykonano i zaprogramowano układ mikroprocesorowy. Praca została wykonana zgodnie z wyznaczonymi celami i wymaganiami.

W pierwszej części pracy został opisany termistor łącznie z wykresem zależności rezystancji od temperatury oraz układami do pomiaru temperatury. Dla każdego z układów został przedstawiony schemat oraz dokładny opis układu.

W rozdziale 3 opisano wykonany przetwornik temperatury na napięcie. Ponadto zaprezentowane zostały elementy przetwornika wraz z układami, w których pracują, schemat przetwornika z dokładnym opisem zasady działania oraz projekt płytki PCB łącznie z fotografią wykonanego przetwornika. Rozdział ten zawiera również otrzymane wyniki badania przetwornika wspólnie z uzyskanymi w pełni akceptowalnymi błędami.

Rozdział 4 przedstawia relację z wykonania i zaprogramowania układu mikroprocesorowego. Zawiera on również dokładny opis mikrokontrolera ATmega8 firmy Atmel wraz z przedstawieniem sposobu programowania przetwornika analogowo cyfrowego i modułu USART. Co więcej, przedstawiono schematy połączeń wymaganych do poprawnej pracy mikrokontrolera, do komunikacji szeregowej oraz do programowania mikrokontrolera.

Wykonana praca pozwoliła nabyć autorowi wiele umiejętności i zagadnień związanych z elektroniką analogową oraz działaniem i programowaniem mikrokontrolerów. Zawarte opisy mogą zostać wykorzystane do nauki programowania mikroprocesorów z rodziny AVR firmy Atmel oraz pomóc w wyborze łatwego i taniego sposobu na pomiar temperatury.

5.2. Możliwości dalszego rozwoju

Do wykorzystania wykonanego projektu na rzeczywistym obiekcie w kotłowni wymagane jest skonstruowanie sterownika. Rolę sterownika może pełnić komputer klasy PC, iPod bądź inne urządzenia posiadające port komunikacji szeregowej RS232, na które można napisać aplikację sterującą urządzeniem. W wykonanym regulatorze została zachowana duża ilość miejsca w pamięci FLASH mikroprocesora. Miejsce to może być również wykorzystane do sterowania odpowiednimi wyjściami w zależności od temperatury. W takim przypadku układ mógłby pełnić rolę samodzielnego sterownika i kontrolować temperaturę w poszczególnych pomieszczeniach budynku, w którym zainstalowano regulator.

Bibliografia

- [1] M. Skorko: *Fizyka*. PWN. Warszawa 1976.
- [2] H. Szydłowski: *Pracownia fizyczna*. PWN. Warszawa 1973.
- [3] Bonnie C. Baker: Microchip Technology Inc.: Application Note - AN685. *Thermistors in Single Supply Temperature Sensing Circuits*. 07.27.2004.
- [4] P. Kluk, M. Bączyk: *Aproksymacja charakterystyki przetwarzania termistora ACCU-CURVE TM*. Elektronika - konstrukcje, technologie, zastosowania nr.7/2009.
- [5] M. Nadachowski, Z. Kulka: *Analogowe układy scalone*. WKŁ. Warszawa 1985.
- [6] A. Chwaleba, B. Moeschke, G. Płoszajski: *Elektronika wyd. 7*. WSiP. Warszawa 1999.
- [7] Texas Instruments, DALLAS, TEXAS 75265: *$\mu A741$ GENERAL-PURPOSE OPERATIONAL AMPLIFIERS*. 2000.
- [8] Atmel Corporation, 2325 Orchard Parkway, San Jose, USA: *8-bit AVR with 8K Bytes In-System Programmable Flash - ATmega8/ATmega8L*. 2010.