

Adam Głowacz*, Witold Głowacz*

Automatyczne sterowanie temperaturą z zastosowaniem mikrokontrolera Motorola MC68HC908QT4CP

1. Wprowadzenie

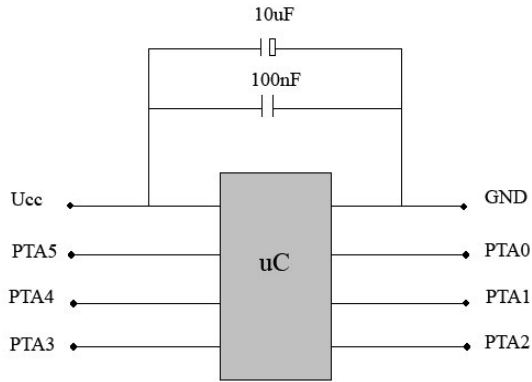
Obecnie istnieje wiele sposobów sterowania temperaturą [1, 3, 7, 10]. Szeroko stosowane jest utrzymywanie temperatury przy użyciu bezprzewodowego ogrzewania podłogi wodą [3]. Sterowanie temperaturą jest ważne [2]. Większość z tych sposobów opartych jest na mierzeniu temperatury i odpowiedniej reakcji systemu [5, 8]. Systemy oparte na sterowaniu temperaturą są stosowane w utrzymywaniu temperatury pomieszczeń [3, 10, 11]. Mogą być również zastosowane do regulacji pracy pieców [9]. Celem niniejszej pracy jest projekt urządzenia umożliwiającego regulację temperatury w sztucznie zbudowanym pomieszczeniu z zastosowaniem mikrokontrolera Motorola MC68HC908QT4CP [4]. Przyrząd ten ma za zadanie utrzymać w komorze nastawioną temperaturę za pomocą żarówki wytwarzającej ciepło oraz wiatraka odpowiedzialnego za chłodzenie. Temperatura powietrza przy wymuszonej konwekcji będzie nie wyższa od założonej temperatury docelowej.

2. Budowa sterownika temperatur

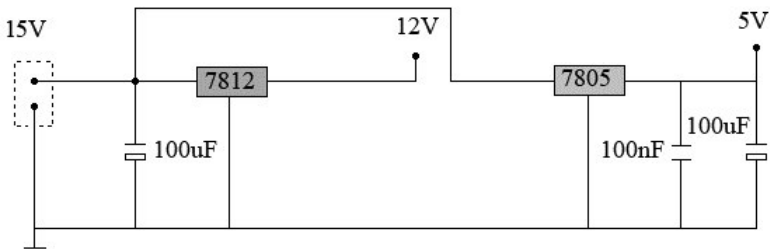
Sterownik temperatur zbudowany został dla pomieszczenia w kształcie prostopadłościanu o wymiarach 10 cm × 10 cm × 15 cm. Ściany pomieszczenia wykonano z laminatu, który ma dobre własności izolacyjne i wytrzymałościowe. Sterownik temperatur pracuje w zakresie od 0 do 100°C. Zasilanie całego układu pochodzi z zasilacza komputerowego 240 W, użycie zasilacza o mniejszej mocy może spowodować niewłaściwe działanie układu. Wentylator, który chłodzi komorę, ma napięcie zasilania 12 V i stałą prędkość obrotową. Układ wyposażony jest w mostek H, który odpowiedzialny jest za przesuwanie tacki z CD-ROM w jedną i w drugą stronę. Mostek H jest układem często stosowanym do sterowania silnikami. Kłapa została zamocowana na drucie. Przesuwana jest za pomocą mechanizmu wysuwania tacki z CD-ROM. Układ do obsługi grzałki i wentylatora został wykonany tak, aby zapewnić odpowiednią pracę tych urządzeń. Zauważyć można, że wejście mikrokontrolera jest przystosowane do obsługi żarówki i działa w tym przypadku jako

* Katedra Automatyki, Akademia Górniczo-Hutnicza w Krakowie

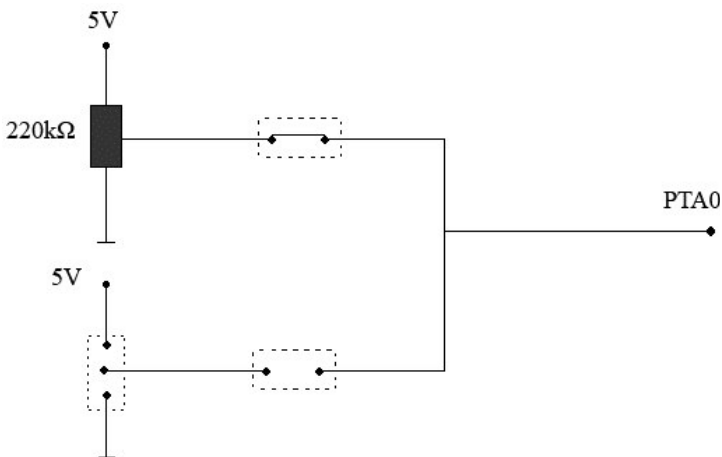
Rysunki 3–9 przedstawiają sposób wyprowadzeń nóżek mikrokontrolera. Każde wyprowadzenie mikrokontrolera zostało wykorzystane w sterowaniu układem.



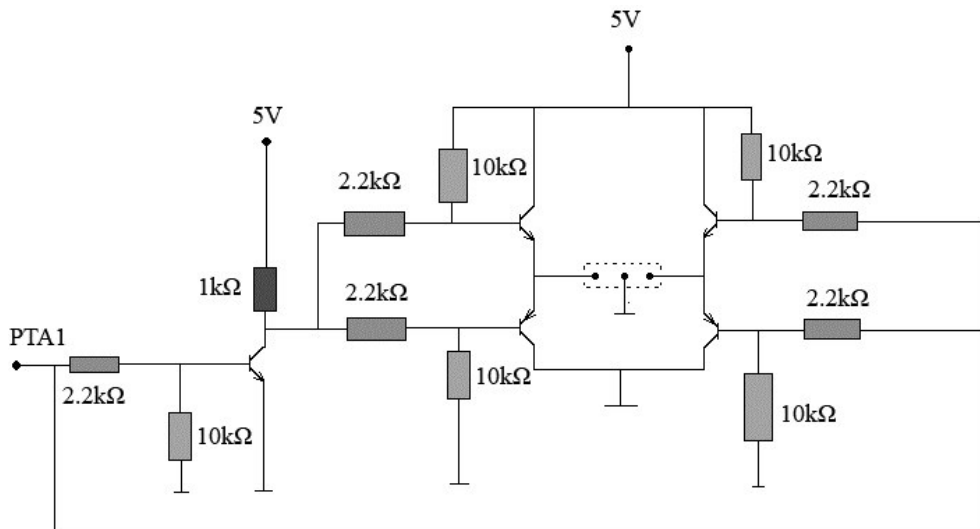
Rys. 3. Układ mikrokontrolera firmy Motorola MC68HC908QT4CP



Rys. 4. Układ zasilania



Rys. 5. Układ potencjometru



Rys. 9. Układ mostka H

3. Sposób działania sterownika temperatur

Sterownik temperatury utrzymuje temperaturę zadaną przez użytkownika. Użytkownik za pomocą potencjometru nastawia określoną temperaturę. W przypadku ustawienia temperatury wyższej niż temperatura otoczenia, zapala się żarówka 45 W, która ogrzewa wnętrze komory. Wentylator wówczas nie pracuje. Kłapa komory jest zamknięta. Dodatkowo zapala się czerwona dioda, aby poinformować, że komora się grzeje. Natomiast w przypadku nastawienia temperatury poniżej temperatury otoczenia, otwiera się kłapa i wentylator zaczyna chłodzić komorę. Żarówka wówczas nie świeci.

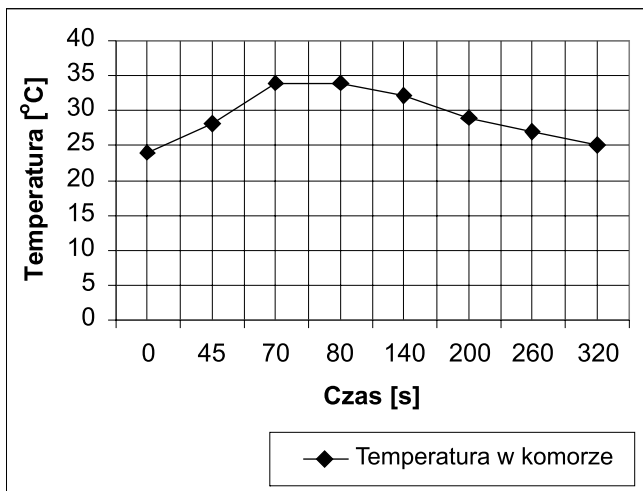
4. Implementacja oprogramowania dla mikrokontrolera

Program umieszczony w mikrokontrolerze firmy Motorola MC68HC908QT4CP zawiera obsługę mechanizmu wysuwania tacki z CD-ROM, wentylatora, żarówki oraz czujnika LM35CZ. Obsługa mechanizmu wysuwania tacki z CD-ROM zaimplementowana jest tak, aby kłapka pudełka mogła się zamykać i otwierać w zależności od tego, czy następuje grzanie lub chłodzenie. Dodatkowo zostało to zaimplementowane w taki sposób, by małe zmiany temperatur nie powodowały natychmiastowego przejścia stanu chłodzenia w grzanie i odwrotnie. Programy dla układu zostały napisane w języku C przy zastosowaniu narzędzia CodeWarrior Development Tools. Kolejny kod programu został zamieniony na od-

powiednią postać w języku Asembler. Wszystkie nóżki mikrokontrolera zostały wykorzystane. Dlatego mikrokontroler firmy Motorola MC68HC908QT4CP jest odpowiedni dla tego układu.

5. Wyniki badań

Badania zostały przeprowadzone dla sterownika temperatur, który znajdował się w pomieszczeniu z temperaturą zewnętrzną 24°C . Badanie oparte było na sprawdzeniu działania sterownika (rys. 10). Na początku nastawiono temperaturę 34°C , wówczas nastąpiło zamknięcie klapki i grzanie żarówki (70 s). Sterownik temperatury utrzymywał stałą temperaturę w komorze (10 s). Gdy temperatura nieznacznie spadła, mikrokontroler włączył żarówkę. Kolejnym krokiem w badaniu było nastawienie chłodzenia. W tym celu nastawiono temperaturę 25°C ze względu na temperaturę otoczenia. W tym przypadku urządzenie otworzyło klapkę i uruchomiło wentylator (240 s). Żarówka w tym czasie nie świeciła.



Rys. 10. Zależność temperatury w komorze od czasu trwania badania

6. Podsumowanie

W pracy zbudowano i badano przyrząd do sterowania temperaturą w pomieszczeniu wykonanym z laminatu. Podobne rozwiązania można wdrażać w innych większych pomieszczeniach przy zastosowaniu innych modułów. Badane urządzenie prawidłowo spełniało swoje zadania. Sterownik temperatur zawsze dążył do uzyskania temperatury zadanej przez użytkownika. Całym układem sterował mikrokontroler Motorola MC68HC908QT4CP.

Dodatek A

Dodatek zawiera kod programu w języku C do obsługi mikrokontrolera Motorola MC68HC908QT4CP.

Kod programu z klasą główną *main.c*

```
#include <hidef.h>
#include <MC68HC908QT4.h>
#include „ podstawowy.h „

void main(void) {
    unsigned char temperatura_zewnetrzna=0;
    unsigned char temperatura_wewnetrzna=0;
    unsigned char threshold=0;
    unsigned char histereza = 12;

    konfiguracja();

    for(;;){
        temperatura_zewnetrzna=przetwornik_AC(channel_3);
        temperatura_wewnetrzna=przetwornik_AC(channel_2); //temperatura w komorze
        threshold=przetwornik_AC(channel_0); //temperatura zadana
        sterowanie(temperatura_zewnetrzna, temperatura_wewnetrzna, threshold, histereza);
        opoznienie(1000);
    }
}
```

Kod pliku ustawień sprzętu po resecie *ustawienia.c*

```
#include <hidef.h>
#include „ podstawowy.h „
#include <MC68HC908QT4.h>

void konfiguracja(void){
    DDRA=0b001010;
    PTA=0b000000;
    PTAPUE=0b00000000;
    CONFIG2=0b10000000;
    CONFIG1=0b00011001;
    ADICLK=0b01000000;
    ADSCR=0b00100000;
    TSC=0b00000000;
    TMOD=200;
    TSC1=0b00011010;
    TCH1=100;
}
```

Kod pliku *podstawowy.h*

```
#include <MC68HC908QT4.h>

#define rejestr_danych_adc ADR;
#define channel_0 0
#define channel_2 2
#define channel_3 3
#define set_channel_0() {ADSCR=0b00100000;}
#define set_channel_2() { ADSCR=0b00100010;}
#define set_channel_3() { ADSCR=0b00100011;}
#define WLACZ_GRZALKE() {PTAPUE|=0b00000100;}
#define WYLACZ_GRZALKE() {PTAPUE&=0b11111011;}
#define WLACZ_WENTYLATOR(){PTA_PTA3=1;}
#define WYLACZ_WENTYLATOR(){PTA_PTA3=0;}
#define wypelnienie TCH1

void konfiguracja(void);
void opoznienie(unsigned int delay);
void sterowanie(unsigned char temperatura_zewnetrzna, unsigned char temperatura_wewnetrzna,
unsigned char threshold, unsigned char histereza);
unsigned char przetwornik_AC (unsigned char kanal);
```

Kod do otwierania klapki pudełka *otwierania-klapki.c*

```
#include <hidef.h>
#include <MC68HC908QT4.h>
#include <stdlib.h>
#include „podstawowy.h”

extern volatile unsigned char licznik_przerwan;
extern unsigned char flaga_klapka_zamknieta;
void otworz_klapke (void){
static unsigned int pomocny_licznik;
static unsigned char flaga_klapka_otwarta;
if (!flaga_klapka_otwarta){
while (pomocny_licznik<50000){
pomocny_licznik++;
if(licznik_przerwan<12){
klapka=0;
} else{
klapka=1;
}
}
}
pomocny_licznik=0;
flaga_klapka_otwarta=1;
flaga_klapka_zamknieta=0;
}
}
```


Kod do otwierania klapki pudełka *zamykania-klapki.c*

```
#include <hidef.h>
#include <MC68HC908QT4.h>
#include <stdlib.h>
#include „podstawowy.h”

extern volatile unsigned char licznik_przerwan;
extern unsigned char flaga_klapka_otwarta;

void zamknij_klapke (void){

static unsigned int pomocny_licznik;
static unsigned char flaga_klapka_zamknieta;

if(!flaga_klapka_zamknieta){
while (pomocny_licznik<50000){
pomocny_licznik++;
if(licznik_przerwan<4){
klapka=0;
} else{
klapka=1;
}
}
pomocny_licznik=0;
flaga_klapka_zamknieta=1;
flaga_klapka_otwarta=0;
}
}
```

Kod programu gdy klapka pudełka jest nieruchoma *klapka-w-soczynku.c*

```
#include <hidef.h>
#include <MC68HC908QT4.h>
#include <stdlib.h>
#include „podstawowy.h”

extern volatile unsigned char licznik_przerwan;

void klapka_w_soczynku (void){

if(licznik_przerwan<6){
klapka=0;
} else{
klapka=1;
}
}
```

Kod programu do ustawienia przetwornika AC *przetwornik-ac.c*

```
#include „podstawowy.h”

unsigned char przetwornik_AC (unsigned char kanal){
static unsigned char FIFO_0[8];
static unsigned char FIFO_1[8];
static unsigned char FIFO_2[8];
static unsigned char i=0;
static unsigned char j=0;
static unsigned char k=0;
unsigned int suma=0;
unsigned char tmp;
if (kanal==0){
set_channel_0()
FIFO_0[i]=rejestr_danych_adc;
i++;
if (i==8){
i=0;
}
for (tmp=0;tmp<8;tmp++){
suma+=FIFO_0[tmp]; //wynik to suma wszystkich el. kolejki FIFO
}
}

if (kanal==2){
set_channel_2()
FIFO_1[j]=rejestr_danych_adc;
j++;
if (j==8){
j=0;
}
for (tmp=0;tmp<8;tmp++){
suma+=FIFO_1[tmp];
}
}

if (kanal==3){
set_channel_3()
FIFO_2[k]=rejestr_danych_adc;
k++;
if (k==8){
k=0;
}
for (tmp=0;tmp<8;tmp++){
suma+=FIFO_2[tmp];
}
}
return ((unsigned char)(suma/8));
}
```

Program do sterowania całym układem *sterowanie.c*

```

#include <hidef.h>
#include <MC68HC908QT4.h>
#include <stdlib.h>
#include „podstawowy.h”

static unsigned char flaga_klapka; //0:otwarta,1:zamknięta

void sterowanie (unsigned char temperatura_zewnetrzna, unsigned char temperatura_wewnetrzna,
unsigned char threshold, unsigned char histereza){
    unsigned char z;
    if ((temperatura_wewnetrzna>=(threshold+histereza))){
        WYLACZ_GRZALKE()
        WYLACZ_WENTYLATOR()
        if(flaga_klapka){ //zamykanie klapki, jesli otwarta
            flaga_klapka=0;
            wypelnienie = 2;
            for (z=0;z<8;z++){
                opoznienie(60000);
            }
        }
        wypelnienie=100; //po zamknięciu klapki wylaczamy silniczek od klapki: pwm=50%, U na
silniczku=0
    }
    if (temperatura_wewnetrzna<=(threshold-histereza)){
        WYLACZ_GRZALKE()
        WYLACZ_WENTYLATOR()
        if(!flaga_klapka){ //otwieranie klapki, jesli zamknięta
            flaga_klapka=1;
            wypelnienie = 183; //pwm~ 91,5%, U na silniczku ~4.5
            for (z=1;z<10;z++){
                opoznienie(60000);
            }
        }
        wypelnienie=100; //po otwarciu klapki wylaczamy silniczek od klapki: pwm=50%
    }
    if((abs(temperatura_wewnetrzna-threshold))<(histereza-6)){ //w pewnym przedziale temp,
mniejszym od podwojonej histerezy, nie wentylujemy
        WYLACZ_GRZALKE()
        WYLACZ_WENTYLATOR()
    }
}

```

Literatura

- [1] Carlos A. González Lozano: *Temperature Control of Solar Air Conditioning Systems*. Master Thesis, University of Puerto Rico, 2004
- [2] Dae-Sik Lee, Hyoung Gil Choi, Kwang Hyo Chung, Bun Yeoul Lee, Hyeon-Bong Pyo, and Hyun C. Yoon: *A Temperature-Controllable Microelectrode and Its Application to Protein Immobilization*. ETRI Journal, vol. 29, no. 5, Oct. 2007, s. 667–669
- [3] Danfoss A/S 99.06.APH.FWR: *Urządzenia do automatycznej regulacji – Projektowanie i zastosowanie*. The Danfoss Journal, nr 3, 1999, s. 8–9
- [4] Freescale Semiconductor, MC68HC908QY/QT Family Data Sheet ,Rev. 5, 07/2005, http://www.freescale.com/files/microcontrollers/doc/data_sheet/MC68HC908QY4.pdf
- [5] Hang Kuang: *A Neural Network-Based Steam Temperature Control System*. First International Workshop on Knowledge Discovery and Data Mining (WKDD 2008), 2008, s. 314–317
- [6] Kuta S.: *Elementy i układy elektroniczne, Część I-II*. Wydawnictwa AGH 2000
- [7] Mitkowski W., Oprzędkiewicz K.: *Sterowanie procesem nagrzewu z wykorzystaniem regulatora rozmytego*. Automatyka (półrocznik AGH), 2001, s. 429–437
- [8] Mukaro R., Gasseller M., Kufazvinei C., Olumekor L., Tael B. M.: *Microcontroller-based multi-sensor apparatus for temperature control and thermal conductivity measurement*. Measurement Science and Technology, vol. 14, 2003, s. 45–49
- [9] Ordyłowski M., Łobodziński W., Rudolf Z., Kałużniacki T., Nowicki G., Sawicki J.P., Zych M.: *Diagnostyka stanu procesu syntezy związku półprzewodnikowego na podstawie pomiarów pośrednich*. Pomiary Automatyka Robotyka, 7–8/2004, s. 108–111
- [10] Sowiński K.: *Wykorzystanie systemu EIB do sterowania ogrzewaniem w budynkach*. Elektroinstalator, nr 9/2003
- [11] Taylor J.: *Recommendations on the control and monitoring of storage and transportation temperatures of medicinal products*. The Pharmaceutical Journal, vol. 267, July 2001, s. 128–131