

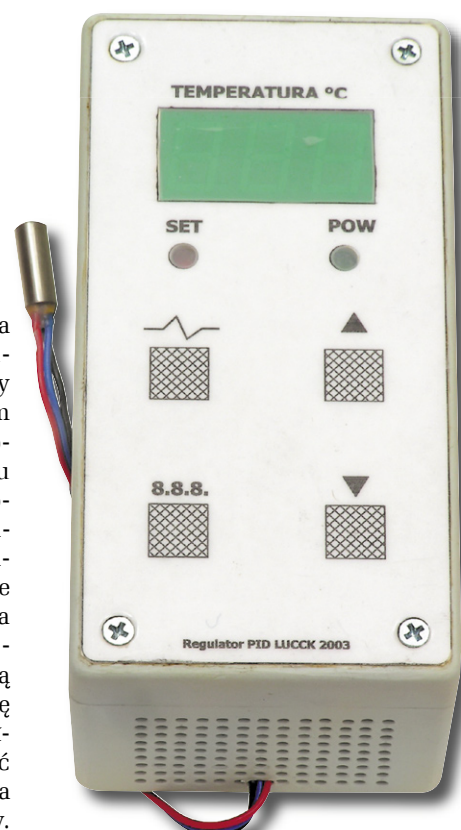
Mikroprocesorowy regulator temperatury PID z interfejsem MODBUS, część 1

Podstawy teoretyczne AVT-5113

Regulatory PID (Proportional-Integral-Derivative – Proporcjonalno-Całkująco-Różniczkujące) wykorzystywane są powszechnie w systemach regulacji parametrów technicznych takich jak: temperatura, ciśnienie, siła, prędkość itp. Dzięki zasadzie działania zapewniają dużą dokładność regulacji.

Rekomendacje:
wykonanie regulatora polecamy szczególnie tym, którzy planują generalny remont swoich mieszkań uwzględniający instalację ogrzewania podłogowego.

W artykule zostanie opisana konstrukcja regulatora temperatury, który może być wykorzystany np. do sterowania elektrycznym ogrzewaniem podłogowym. Dostępne na rynku urządzenia tego typu to najczęściej regulatory dwustanowe (załącz-wyłącz) nie zapewniające dobrej dokładności stabilizacji temperatury oraz pozbawione najczęściej możliwości sterowania z komputera. W opisywanym projekcie zastosowano regulację ciągłą PID, zapewniającą płynną regulację mocy grzałki. Regulacja mocy grzałki, z uwagi na dużą bezwładność termiczną obiektów, odbywa się na zasadzie grupowej regulacji mocy. Nazwa „regulacja grupowa” pochodzi stąd, że układ zasilania odbiornika energii elektrycznej za pomocą grup przebiegów sinusoidalnych włączanych zawsze przy napięciu bliskim zeru. Sterownie grupowe możemy z dużym przybliżeniem porównać do regulacji PWM stosowanej w obwodach prądu stałego, gdzie średnie napięcie na elemencie wykonawczym zależy od stosunku czasu włączenia do czasu wyłączenia w danym okresie. Rolę elementu wykonawczego sterującego grzałką pełni triak. W regulatorze zaimplementowano algorytm samostrojenia, który służy do automatycznego doboru nastaw regulatora w zależności od parametrów obiektu. Urządzenie wyposażono w panel operatorski umożliwiający zmianę niektórych parametrów bezpośrednio z urządzenia. Można to robić również z komputera PC poprzez interfejs RS485 (transmisja 1200, 8, e, 1).

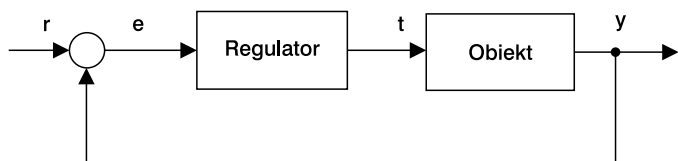


W ten sposób można połączyć w sieć większą liczbę sterowników. Regulator komunikuje się z innymi urządzeniami w takiej sieci wykorzystując protokół RTU-MODBUS firmy MODICON. Jest to standard komunikacyjny dla sterowników przemysłowych.

Sterownik tego typu wykorzystuję w swoim domku letniskowym, gdzie zostały zainstalowane cztery regulatory ustalające temperaturę w różnych pomieszczeniach. Wszystkie sterowniki są połączone ze sobą za pomocą magistrali RS485, do której za pomocą konwertera RS485<->RS232 (AVT530) został podłączony komputer PC (Master w protokole MODBUS). Umożliwia to kontrolę temperatury w całym domu z jednego komputera. Nic nie stoi na przeszkodzie, aby napisać odpowiedni program umożliwiający sterowanie temperaturą np. za pomocą SMS-ów. Regulator z triakiem

PODSTAWOWE PARAMETRY

- Płytki o wymiarach: 122x57 mm
- Zasilanie 230 V
- Zakres pomiaru temperatury: -10°C...+85°C
- Dokładność pomiaru temperatury: 0,5°C
- Moc grzałki dołączonej do regulatora 2,5 kW
- Interfejs: RS485
- Parametry transmisji: 1200, 8,e,1



Rys. 1. Schemat typowego obiektu regulacji

BT139 umożliwia sterowanie grzałką o mocy do 2,5 kW.

Trochę teorii sterowania i protokół MODBUS

Do zrozumienia zasady działania regulatora PID konieczny będzie „łyk” matematyki. Wiedza ta w możliwie minimalnym zakresie zostanie przytoczona poniżej, przy czym nie jest ona niezbędna do praktycznej obsługi regulatora.

Najprostszemu układowi regulacji (rys. 1) składa się z obiektu (w naszym przypadku pomieszczenia ogrzewanego za pomocą mat grzewczych), członu regulacyjnego, członu porównującego (regulator) oraz toru pomiarowego – pętli sprzężenia zwrotnego (czujnik temperatury). Zadaniem takiego układu regulacji jest utrzymanie wybranego parametru w procesie (tutaj temperatury) na zadanym poziomie. Jeżeli przez r oznaczymy wielkość zadaną (temperaturę ustawioną), zaś przez y zmienną procesową (pomiar wartości wyjściowej z obiektu, czyli temperaturę bieżącą), to wartość uchybu

powiedni regulator i jego parametry, musimy znać charakterystykę obiektu. Najpierw na podstawie obserwacji należy zidentyfikować obiekt. Najprostszą i najbardziej rozpowszechnioną metodą identyfikacji obiektu jest analiza odpowiedzi na ściśle określone i znane wymuszenie. Podając na wejście określone sygnały testowe, możemy wyznaczyć właściwości obiektu. Właśnie na tej zasadzie działa algorytm regulatora służący do samostrojenia. W naszym regulatorze sygnałem testowym jest załączenie grzałki na pełną moc, a odpowiedzią jest zmiana temperatury otoczenia. Obiekty cieplne to obiekty inercyjne I rzędu z opóźnieniem, charakteryzujące się czasem opóźnienia odpowiedzi τ , stałą czasową obiektu T_z oraz wzmocnieniem k_o . Transmitancja i przykładowa charakterystyka odpowiedzi na wymuszenie skokowe (nagle załączenie sterowania) jest przedstawiona na rys. 2.

Na osi x zaznaczono upływ czasu, natomiast oś y jest wyskalowana w procentach, gdzie w przypadku sterowania (r) 0% (0,0)

oznacza wyłączenie grzałki, natomiast 100% (1,0) pracę z pełną mocą. Odpowiedź obiektu (Y_u) 0% (0,0) oznacza założoną temperaturę minimalną natomiast 100% (1,0) założoną temperaturę maksymalną. W chwili $t=0$ zostaje załączona grzałka na 100% mocy (linia A). W odpowiedzi na wymuszenie (załączenie grzałki) obiekt po czasie τ odpowiada przyrostem temperatury (linia B), która zaczyna gwałtownie rosnąć, następnie rośnie coraz wolniej, aby po pewnym czasie ustabilizować się na praktycznie stałym poziomie. Transmitancja obiektu cieplnego ma postać:

$$G_m(s) = \frac{k_o}{T_z \cdot s + 1} \cdot e^{-\tau \cdot s}$$

gdzie:

k_o – wzmocnienie obiektu,
 T_z – stała czasowa obiektu,
 τ – opóźnienie obiektu.

Wszystkie parametry odczytuje się bezpośrednio z wykresu, prowadząc styczną w punkcie przegięcia P wykresu odpowiedzi obiektu w czasie. Wzmocnienie obiektu (k_o) wyznaczamy zgodnie ze wzorem:

$$k_o = \Delta y_u / \Delta r$$

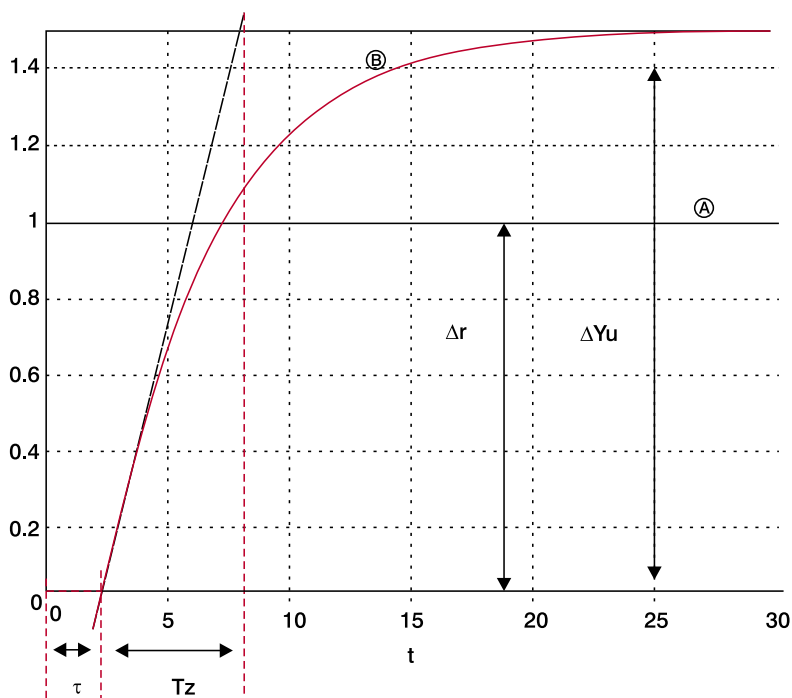
gdzie:

Δy_u – przyrost odpowiedzi (temperatury) obiektu od czasu $t=0$ do $t=t_1$

Δr – przyrost sterowania (mocy grzałki) od czasu $t=0$ do $t=t_1$

Na podstawie przebiegu zmian temperatury w czasie, w odpowiedzi na załączenie grzałki, regulator wyznacza nieznanne parametry: opóźnienie τ , stałą czasową T_z oraz wzmocnienie obiektu k_o . Parametry te są następnie wykorzystane do wyznaczenia nastaw regulatora PID.

Regulator jest urządzeniem przetwarzającym sygnał uchybu regulacji e na sygnał sterujący u . Uchyb jest różnicą między wartością zadaną a wartością mierzoną (pochodzącą z procesu) $e=r-y$. Sposób przetwarzania sygnału uchybu e na sterowanie nazywany jest algorytmem regulacji. Dla obiektów cieplnych najodpowiedniejszym regulatorem jest regulator PI lub PID. My wykorzystujemy regulator PID. Algorytm PID opiera się na wzorze określającym odpowiedź regulatora $u(t)$ na wymuszenie $e(t)$:



Rys. 2. Odpowiedź na wymuszenie skokowe obiektu inercyjnego I rzędu z opóźnieniem

START	NUMER STEROWNIKA	NR FUNKCJI	DANE	CRC16	KONIEC
T1-T2-T3-T4	8 bitów	8 bitów	N x 8 bitów	16 bitów	T1-T2-T3-T4

Rys. 3. Ramka MODBUS-RTU

$$u(t) = K_p \cdot$$

$$\left[e(t) + \frac{1}{T_i} \int e(t)dt + T_d \frac{de(t)}{dt} \right]$$

Wzór ten w zapisie operatorowym określa transmitancję G (s) regulatora:

$$G_r(s) = K_p \cdot \left[1 + \frac{1}{T_i s} + T_d s \right]$$

gdzie:

K_p – wzmacnienie regulatora,

T_i – stała czasowa całkowania,

T_d – stała czasowa różniczkowania.

Stosując metodę prostokątów otrzymamy cyfrowy algorytm regulatora PID, wygodny do realizacji mikroprocesorowej. Odpowiednia zależność ma postać:

$$u[n] = K_p \cdot (e[n] + \frac{T_p}{T_i} \cdot \sum e[i-1] + \frac{T_d}{T_o} * (e[n] - e[n-1]))$$

gdzie:

T_p – czas próbkowania,

$e[n]$ – uchyb w n-tym okresie próbkowania.

Na tej podstawie możemy przejść wprost do implementacji algorytmu w języku C, który zostanie przedstawiony w dalszej części artykułu. Aby regulator mógł działać w sposób prawidłowy musimy odpowiednio ustawić jego parametry, czyli K_p , T_i , T_d , T_p w zależności od charakterystyki sterowanego obiektu. Złe nastawy regulatora mogą spowodować jego nieprawidłowe działanie (np. powstawanie oscylacji) i w efekcie brak stabilizacji temperatury. Dlatego dobór nastaw regulatora do charakterystyki obiektu jest zagadnieniem bardzo istotnym. Dla obiektów jednoinercyjnych z opóźnieniem, regulator PID można nastawić wg następujących wzorów empirycznych:

$$\beta = \frac{k_o \cdot K_p \cdot \tau}{T_z}$$

gdzie:

$\beta = 1$,

$T_i = 2,4 * \tau$,

$T_d = 0,4 * \tau$,

Aby przeprowadzić proces samostrojzenia, musimy najpierw wyznaczyć parametry obiektu na podstawie eksperymentu ze skokiem jednostkowym, a następnie na podstawie tych parametrów wyznaczyć wg powyższych wzorów nastawy regulatora. Po wyznaczeniu parametrów, regulator przepisuje te parametry do zmiennych roboczych w pamięci RAM oraz do pamięci konfiguracji EEPROM. Możemy również pokusić się o ręczne wyznaczenie parametrów K_p , T_i , T_d , a następnie wprowadzenie ich do regulatora za pomocą programu serwisowego *rt1_service.exe*

Jak już wiemy, w naszym sterowniku do komunikacji z otoczeniem wykorzystano protokół MODBUS. Jest on stosowany powszechnie w sterownikach przemysłowych. Protokół MODBUS działa na zasadzie pytań i odpowiedzi. Do jednej magistrali może być podłączonych wiele sterowników (Slave), które odbierają zapytania pochodzące od sterownika nadrzędnego (Master). Protokół dopuszcza tylko jedno urządzenie Master na magistrali. Sterownik Slave może wysłać dane na magistralę tylko na zapytanie urządzenia Master, odpada więc problem arbitrażu. Do magistrali RS485 pracującej w standardzie MODBUS bez żadnych dodatkowych układów wzmacniających może być podłączonych do 32 urządzeń, natomiast sam protokół MODBUS dopuszcza możliwość użycia 255 urządzeń. Ponadto istnieją dwie odmiany protokołu MODBUS, mianowicie MODBUS-ASCII, który charakteryzuje się tym, że dane są przesyłane w postaci znaków ASCII oraz MODBUS-RTU, w którym dane są przesyłane w postaci binarnej. W naszym przypadku, ze względu na mniejszą ilość przesyłanych danych oraz większą powszechność, wykorzystano transmisję MODBUS-RTU. Wybrano prędkość transmisji 1200 bitów/s.

Działanie protokołu MODBUS polega na wysyłaniu ramek z zapytania. Na prawidłowe zapytania sterownik musi przesłać ramkę odpowiedzi. Ramka komunikacyjna MODBUS przesyłana poprzez interfejs RS485 ma format przedstawiony na rys. 3.

Przed rozpoczęciem transmisji musi wystąpić przerwa trwająca minimum 4 znaki, po tym czasie następuje transmisja danych, które składają się z 8-bitowego adresu urządzenia, do którego jest kierowane zapytanie, numeru funkcji (tzn. kod operacji), danych wysyłanych, a następnie 16-bitowego kodu CRC służącego do weryfikacji przesłanych danych. W naszym sterowniku wykorzystano tylko funkcję 3 (*READ HOLDING REGISTER*), która służy do odczytu danych rejestrów analogowych oraz funkcję 6 (*PRESET SINGLE REGISTER*), która służy do ustawiania danych w rejestrach analogowych.

Przykładowe zapytanie o stan rejestrów analogowych A8...A10 skierowane do urządzenia Slave o numerze 0x11, wysłane przez urządzenie Master, ma następującą postać:

```
Adres sterownika docelowego: 0x11
Numer Funkcji: 0x03 (READ HOLDING REGISTER)
Adres pierwszego rejestru (HI): 0x00
Adres pierwszego rejestru (LOW): 0x07
Liczba rejestrów (HI): 0x00
Liczba rejestrów (LOW): 0x03
Kontrola CRC: (HI): -
Kontrola CRC: (LOW): -
Przykładowa odpowiedź urządzenia Slave o adresie 0x11:
Adres sterownika docelowego: 0x11
Numer Funkcji: 0x03 (READ HOLDING REGISTER)
Liczba bajtów danych: 0x06
Dane (HI) (Rejestr A8): 0x12
Dane (LOW) (Rejestr A8): 0x45
Dane (HI) (Rejestr A9): 0x17
Dane (LOW) (Rejestr A9): 0x12
Dane (HI) (Rejestr A10): 0x55
Dane (LOW) (Rejestr A10): 0x77
Kontrola CRC: (HI): -
Kontrola CRC: (LOW): -
```

Przykładowe żądanie wysłane przez urządzenie Master do urządzenia Slave o numerze 0x11, żądające ustawienia wartości 259 w rejestrze analogowym A8, ma następującą postać:

```
Adres sterownika docelowego: 0x11
Numer Funkcji: 0x03 (READ HOLDING REGISTER)
Adres rejestru (HI): 0x00
Adres rejestru (LOW): 0x07
Zawartość rejestru (HI): 0x10
Zawartość rejestru (LOW): 0x03
Kontrola CRC: (HI): -
Kontrola CRC: (LOW): -
```

Sterownik po ustawieniu żądanego rejestru musi do urządzenia Master odesłać ramkę o identycznej treści, jak zapytanie. Jak więc widać, protokół MODBUS jest stosunkowo prosty i warto go stosować w swoich urządzeniach. Dodatkową korzyścią jest możliwość korzystania z licznych programów obsługi dostępnych w Internecie.

Lucjan Bryndza, EP
lucjan.bryndza@ep.com.pl