

Konstrukcja systemu telemetrycznego z zastosowaniem technologii internetowych

*wykorzystanie mikrokontrolera do realizacji
akwizycji danych oraz zaimplementowanie
w nim serwera WWW i serwera bazy danych jako
podstawy do zdalnego monitorowania obiektów*

- mikrokontroler 8-bitowy nadzorujący pracę systemu
- autonomiczność systemu (możliwość samodzielnej pracy bez komputera)
- zestaw czujników do wielopunktowego pomiaru temperatury
- sterowanie gradientem temperatury
- możliwość akwizycji danych pomiarowych wewnątrz systemu
- dostęp do systemu poprzez interfejs ethernetowy
- mini-serwer WWW
- wyświetlacz LCD używany do podglądu stanu systemu
- diagnostyczne złącze RS232
- końcówki mocy umożliwiające sterowanie układami wykonawczymi (grzałki) – sterownik PWM
- możliwość współpracy systemu z pakietem MATLAB

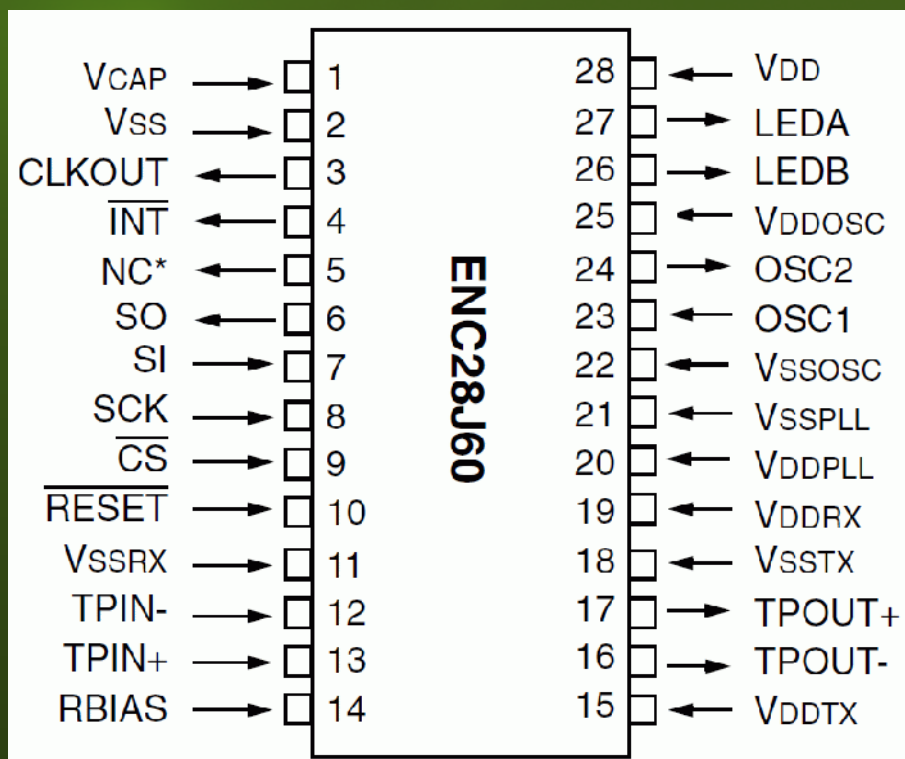
Elektronika: μ CATmega32

PDIP

(XCK/T0) PB0	1	40	PA0 (ADC0)
(T1) PB1	2	39	PA1 (ADC1)
(INT2/AIN0) PB2	3	38	PA2 (ADC2)
(OC0/AIN1) PB3	4	37	PA3 (ADC3)
($\overline{SS}$) PB4	5	36	PA4 (ADC4)
(MOSI) PB5	6	35	PA5 (ADC5)
(MISO) PB6	7	34	PA6 (ADC6)
(SCK) PB7	8	33	PA7 (ADC7)
$\overline{RESET}$	9	32	AREF
VCC	10	31	GND
GND	11	30	AVCC
XTAL2	12	29	PC7 (TOSC2)
XTAL1	13	28	PC6 (TOSC1)
(RXD) PD0	14	27	PC5 (TDI)
(TXD) PD1	15	26	PC4 (TDO)
(INT0) PD2	16	25	PC3 (TMS)
(INT1) PD3	17	24	PC2 (TCK)
(OC1B) PD4	18	23	PC1 (SDA)
(OC1A) PD5	19	22	PC0 (SCL)
(ICP1) PD6	20	21	PD7 (OC2)

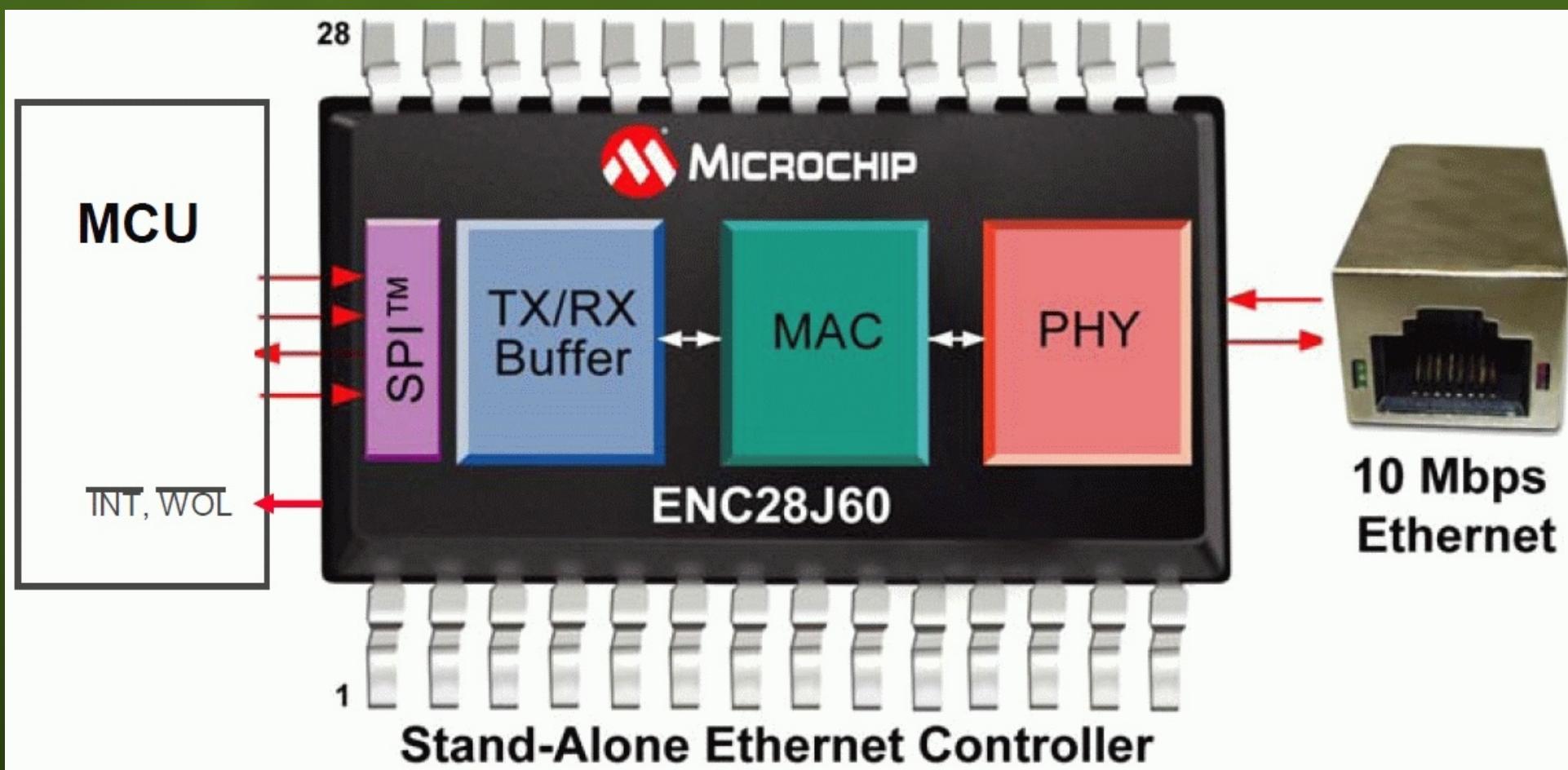
- 8-bitowa architektura RISC
- taktowanie do 16MHz
- pamięć programu (flash): 32 kB
- pamięć SRAM: 2 kB
- pamięć EEPROM: 1 kB
- linie I/O: 32
- interfejsy:
 - USART
 - SPI
 - I²C
- zasilanie: 5V
- pobór mocy: 110mW (@16MHz)

Interfejs ethernetowy: enc28j60



- kompletny sterownik Ethernetu 10Base-T (do 10Mbit/s)
- sprzętowa implementacja warstwy PHY i MAC
- interfejs SPI (do 20MHz)
- zgodność z IEEE802.3
- 8 kB pamięci SRAM na pakiecie (odbiór/wysyłka)
- programowalna filtracja pakietów
- proste interfejsowanie:
 - ✓ gniazdo ethernetowe RJ45 z transformatorem
 - ✓ magistrala SPI

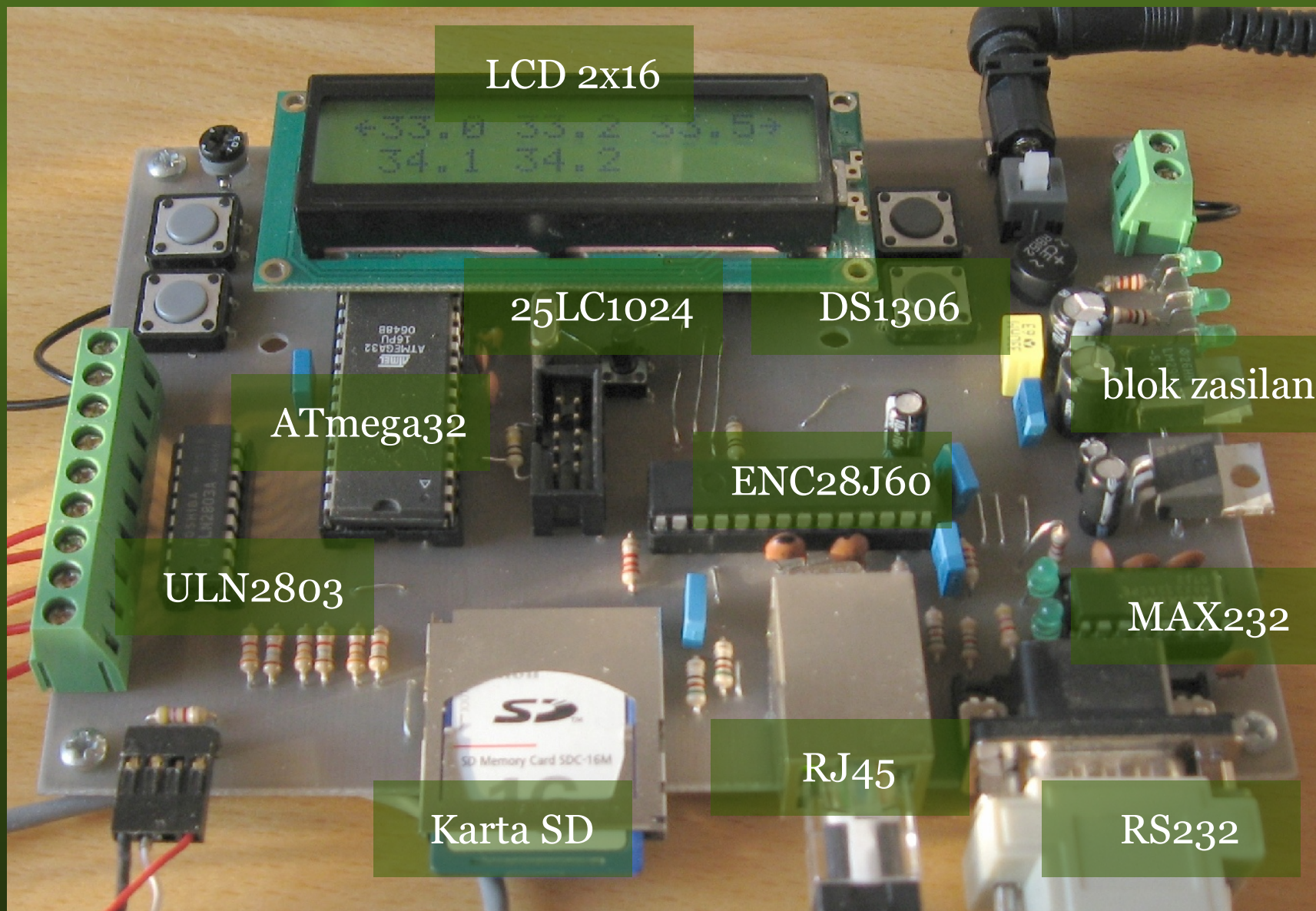
Interfejs ethernetowy: enc28j60



Interfejs ethernetowy: stos TCP/IP

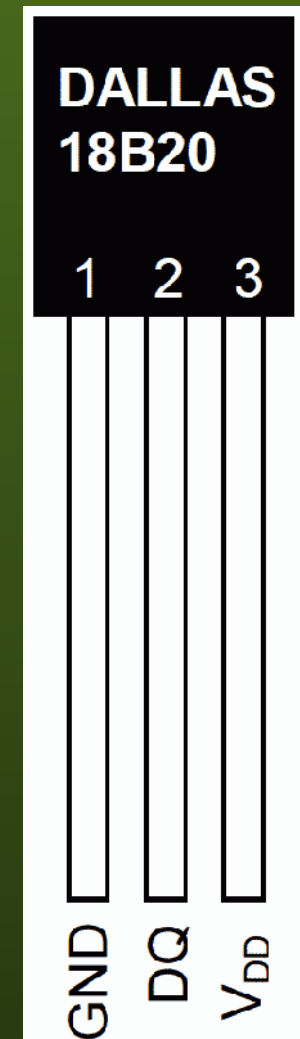
- stos TCP/IP zrealizowany programowo (μ C Atmega32)
- obsługiwane protokoły TCP/UDP:
 - ARP (konwersja adresów IP na MAC)
 - ICMP (**p**ing)
 - DHCP (autokonfiguracja interfejsu)
 - NTP (pobieranie czasu z serwerów NTP)
 - HTTP (dostęp do systemu z poziomu przeglądarki)
- struktura kodu umożliwia łatwe dodawanie obsługi kolejnych protokołów

Elektronika: płyta główna

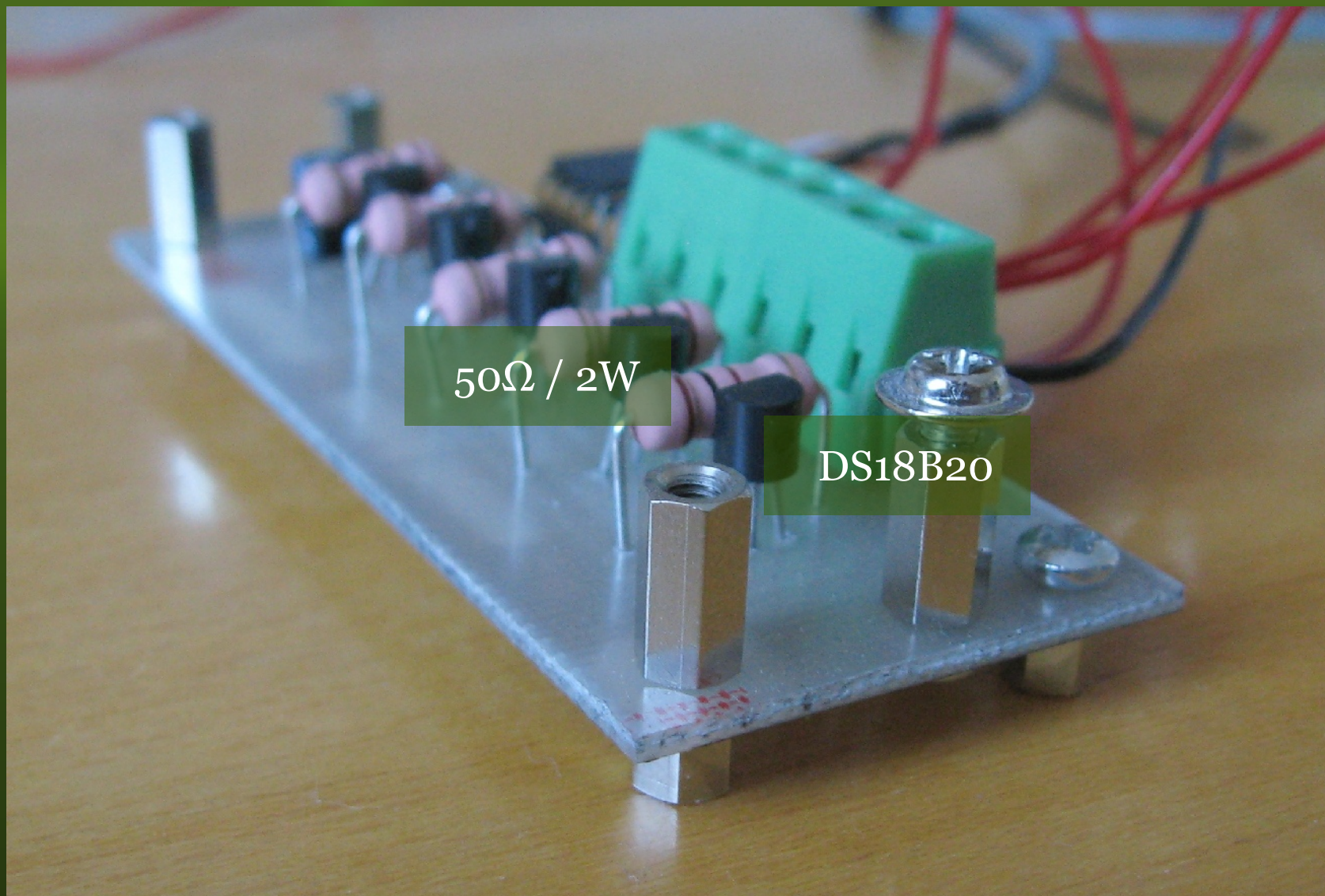


Płytki grzewcza: czujniki

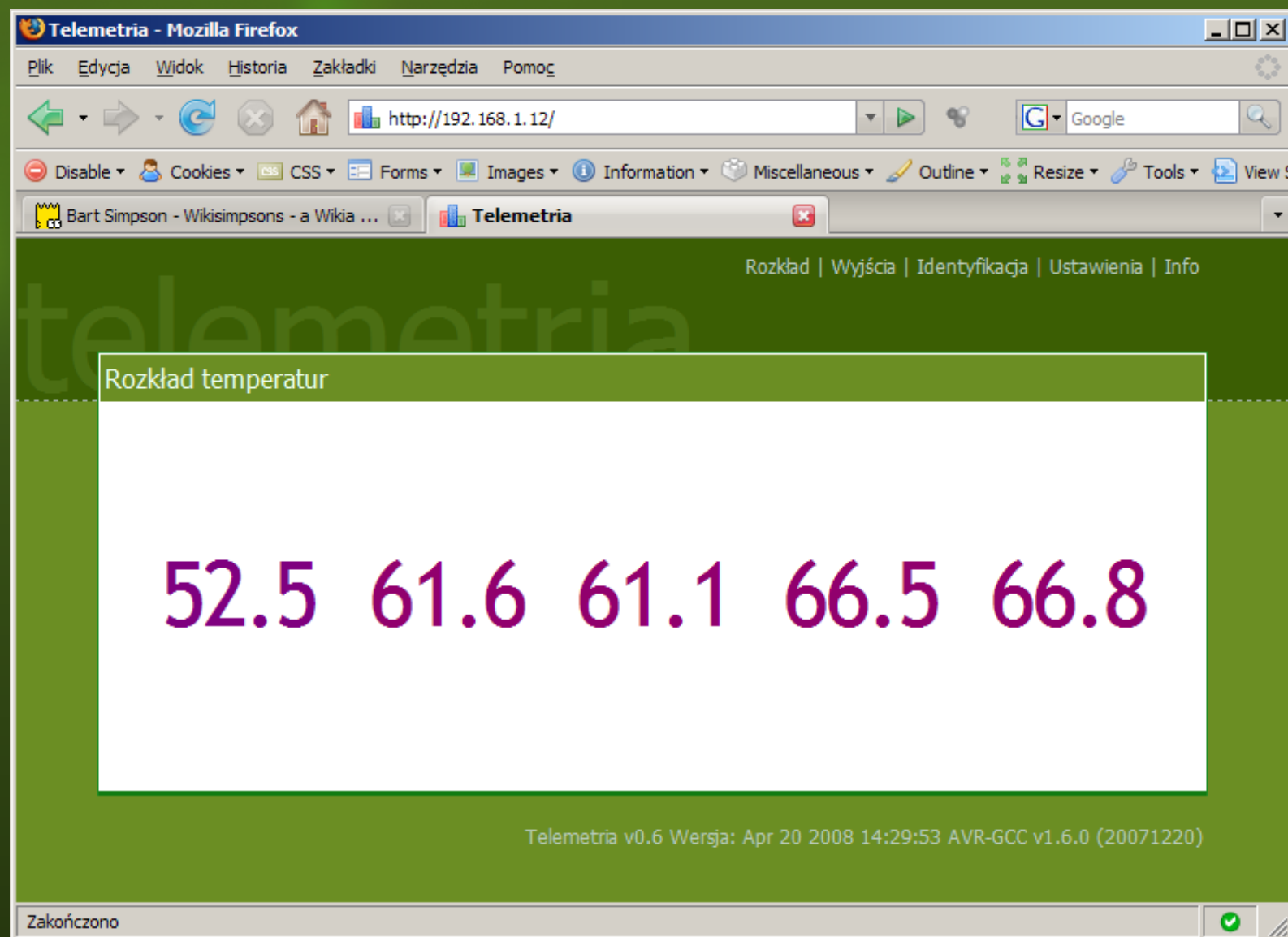
- cyfrowy czujnik temperatury
- zakres -55 do 125°C
- dokładność $\pm 0.5^\circ\text{C}$ (w zakresie od -10 do 85°C)
- interfejs 1-wire
 - transmisja danych po jednym przewodzie
 - sieć typu multi-slave (proste użycie wielu czujników)
- programowalny alarm - temperatura poza ustalonym zakresem $\langle T_L; T_H \rangle$
- czas pomiaru do 750 ms (12-bitowa precyzja)
do 190ms (10-bitowa precyzja)



Płytki grzewcza: realizacja



Interfejs ethernetowy: WWW



Interfejs ethernetowy: WWW

- aktualizowane co sekundę pomiary temperatury
- sterownik PWM (podgląd aktualnych wartości wypełnień w poszczególnych kanałach oraz możliwość ich zmiany)
- akwizycja danych pomiarowych
- procedura identyfikacji parametrów systemu
- informacja o systemie (dane o używanej karcie pamięci, wersja oprogramowania, adres IP i MAC, możliwość zmiany nastaw zegara RTC systemu wg czasu komputera)

Interfejs ethernetowy: WWW

- Wykorzystane nowoczesne technologie internetowe:
 - JavaScript (JS)
 - Cascading Style Sheets (CSS)
 - Asynchronous JavaScript And XML (AJAX)
 - JavaScript Object Notation (JSON)
- Odciążenie mini-serwera WWW
- Aktualizacja danych bez konieczności odświeżania strony
- Odpowiednie nagłówki HTTP odpowiedzialne za buforowanie treści po stronie przeglądarki
- EEPROM (interfejs SPI) o pojemności 128 kB jako pamięć na pliki HTML, CSS i JS wykorzystywane przez serwer – aktualizacja skryptem PHP (pakiety UDP)

Algorytm sterowania gradientem

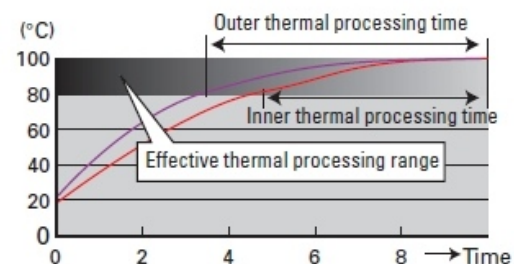
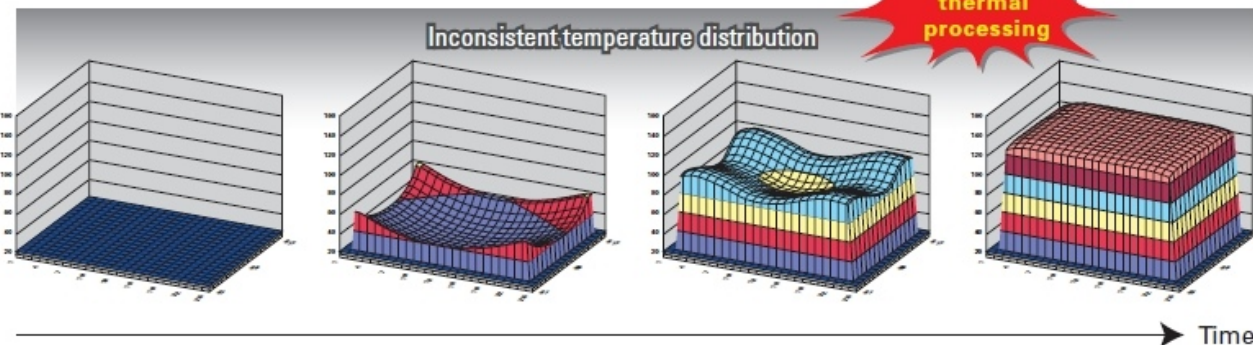
- Jako aplikację dla systemu telemetrycznego wybrano zadanie sterowania wielopunktowym rozkładem temperatur na płytce grzewczej
- Algorytm sterowania gradientem - zadanie: zapewnienie jednorodnego rozkładu temperatury na powierzchni płytki

Matsunaga N., Nanno I., Tanaka M., Kawaji S.
*A New Method of Temperature Control
for Uniform Heating (rok 2003)*

Algorytm sterowania gradientem

Both uniform in-plain control and user-set temperature distribution are possible.

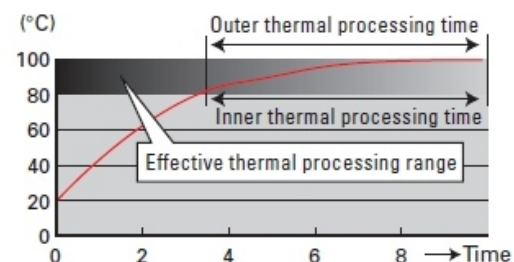
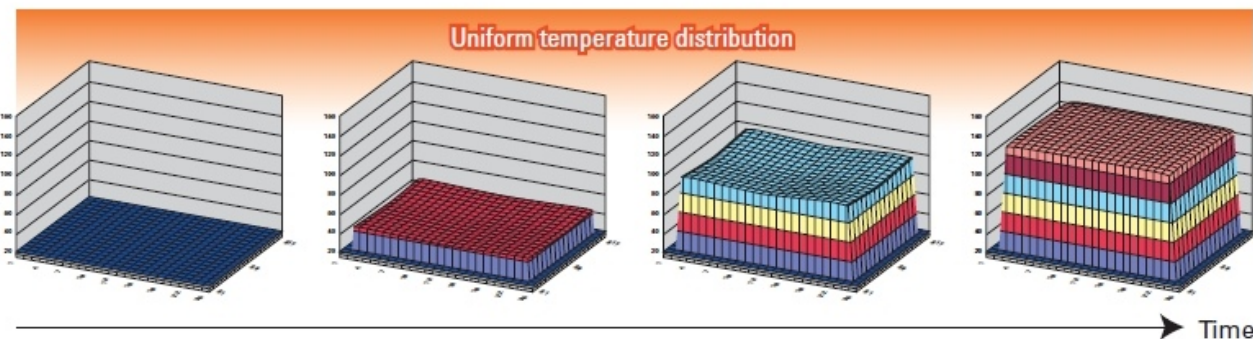
Traditional temperature control



Faulty thermal processing at the outer edges



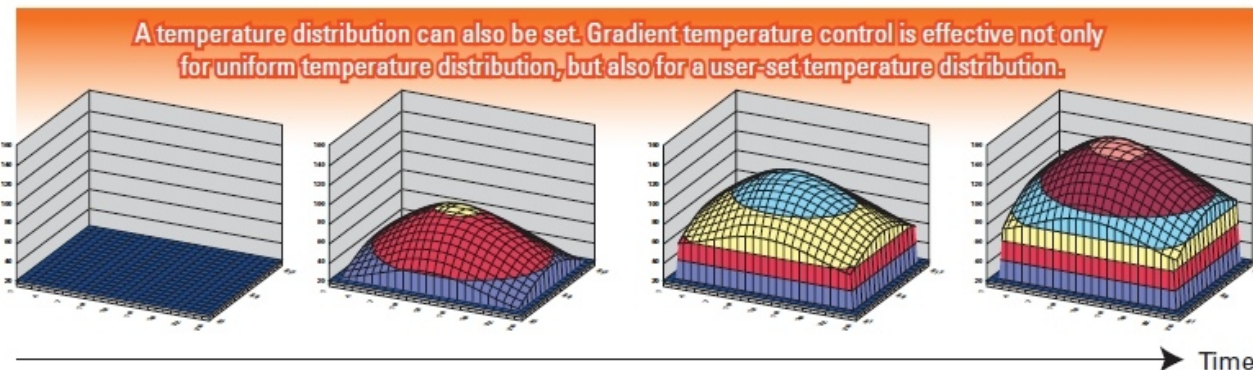
Uniform temperature distribution



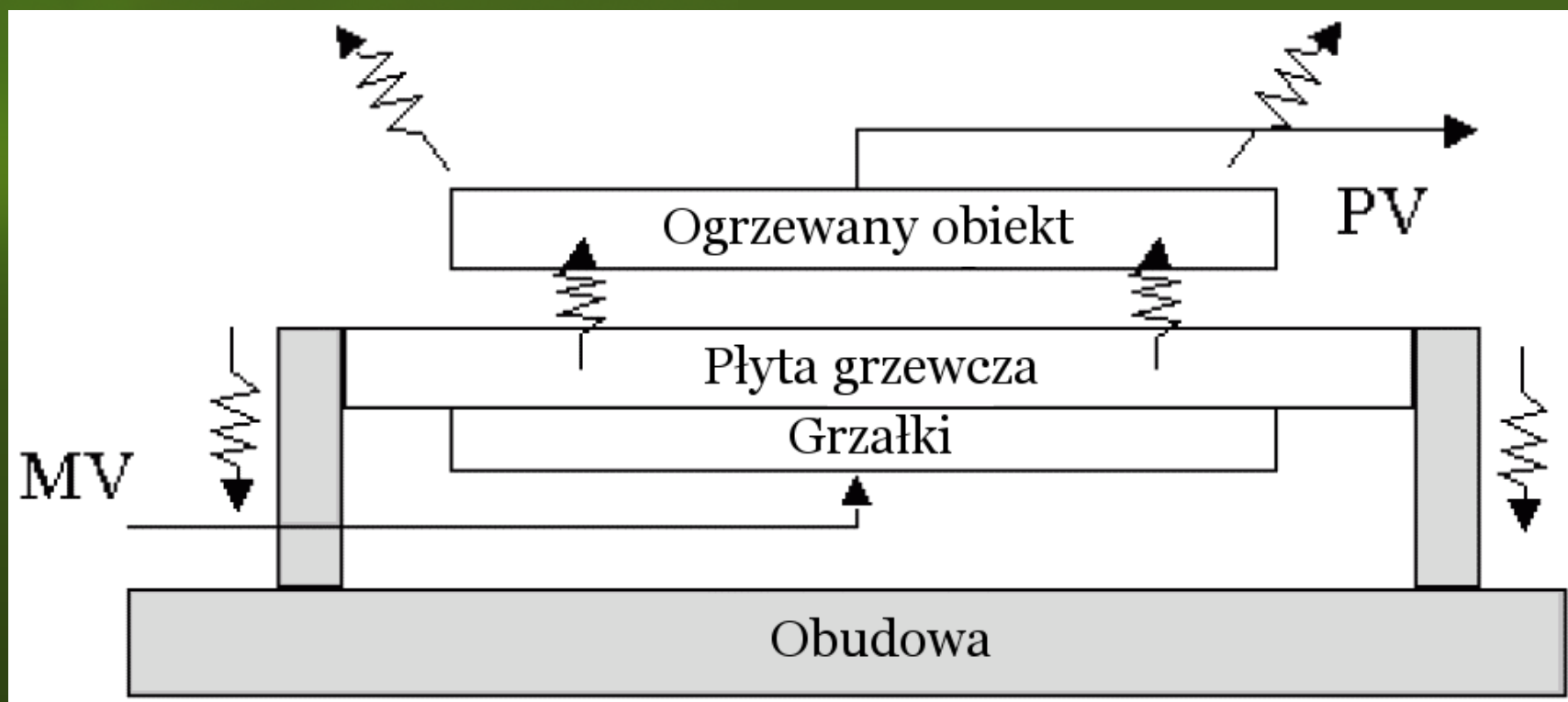
High quality achieved for entire area



Gradient temperature control

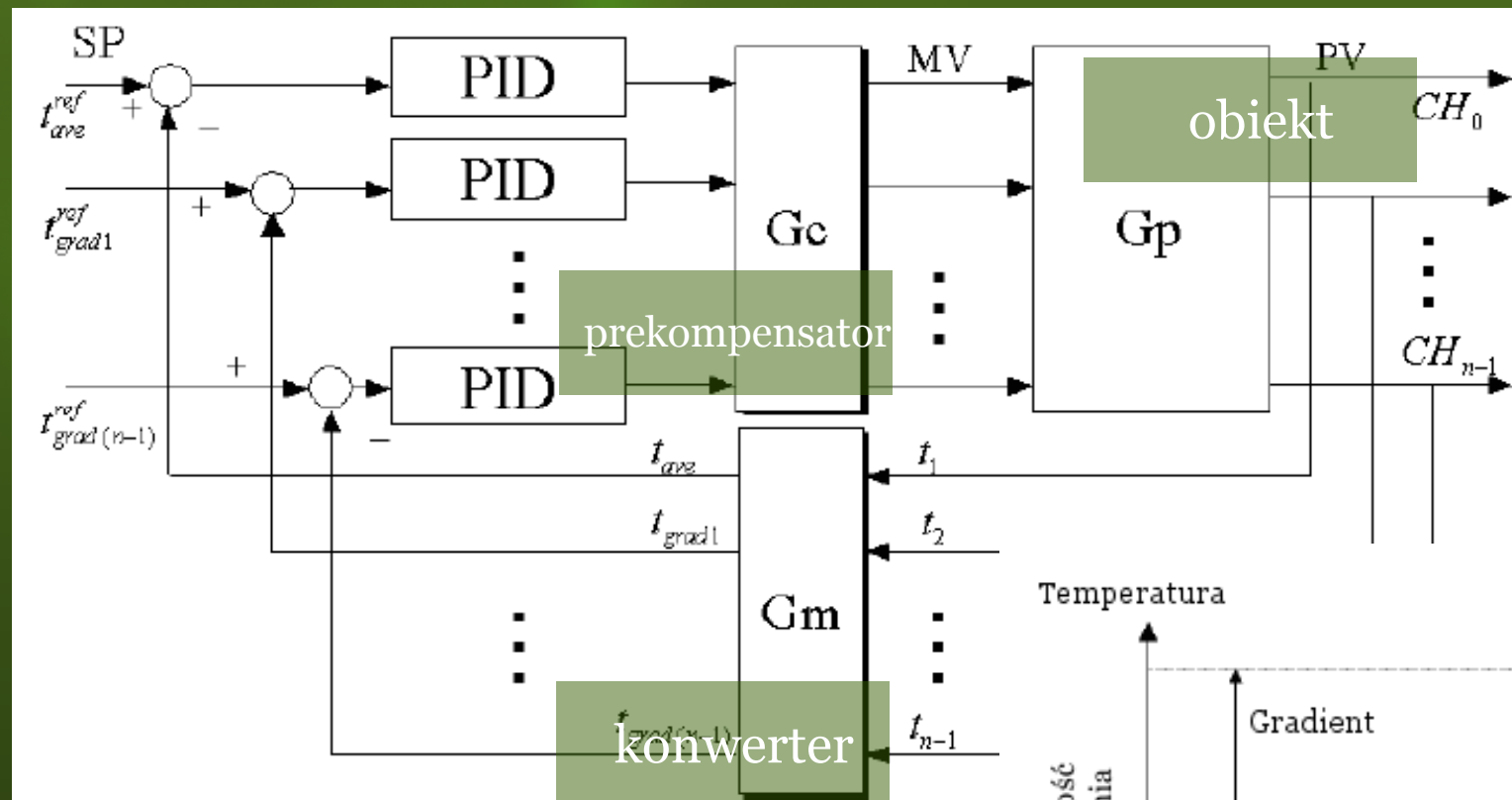


Algorytm sterowania gradientem



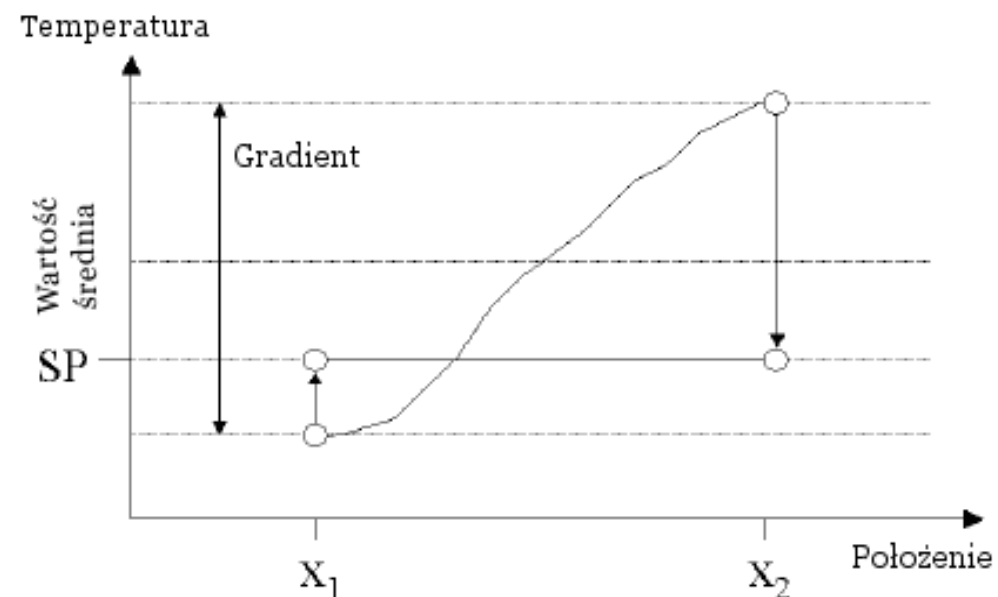
Rozproszenie temperatury na powierzchni płyty grzewczej

Algorytm sterowania gradientem



$$t_{ave} = \sum_{j=0}^{n-1} \frac{t_j}{n}$$

$$t_{grad} = [t_1 - t_0, \dots, t_{n-1} - t_n]$$



Algorytm sterowania gradientem

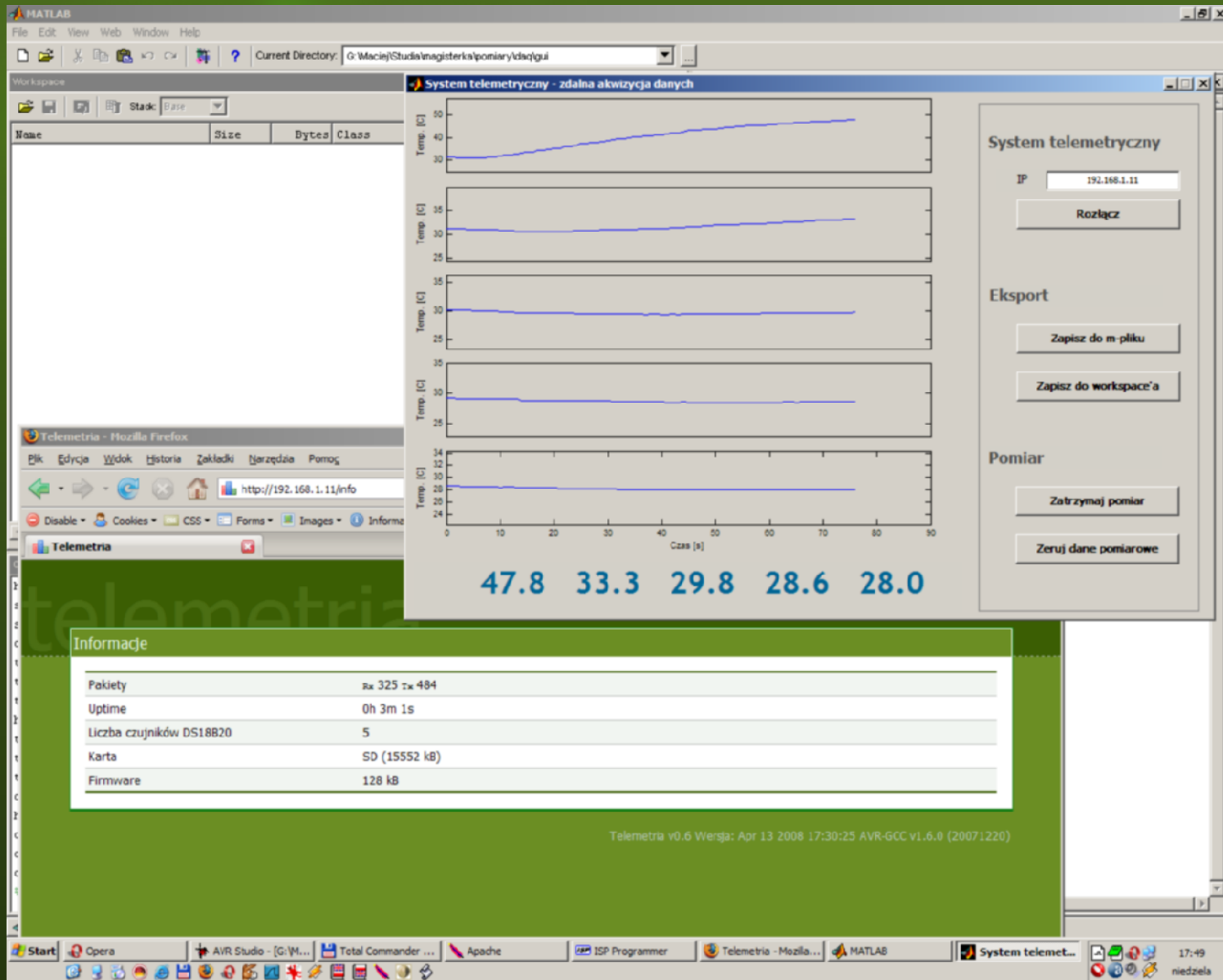
$$G_m = \begin{bmatrix} \frac{1}{n} & \frac{1}{n} & \frac{1}{n} & \cdots & \frac{1}{n} \\ -1 & 1 & 0 & \cdots & 0 \\ 0 & -1 & 1 & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & -1 & 1 \end{bmatrix}$$

$$G_p = \text{diag} (G_{00}, G_{11}, \dots, G_{nn})$$

$$G_c = (G_m \cdot G_p)^{-1}$$

$$|G_m(0) \cdot G_p(0)| \neq 0$$

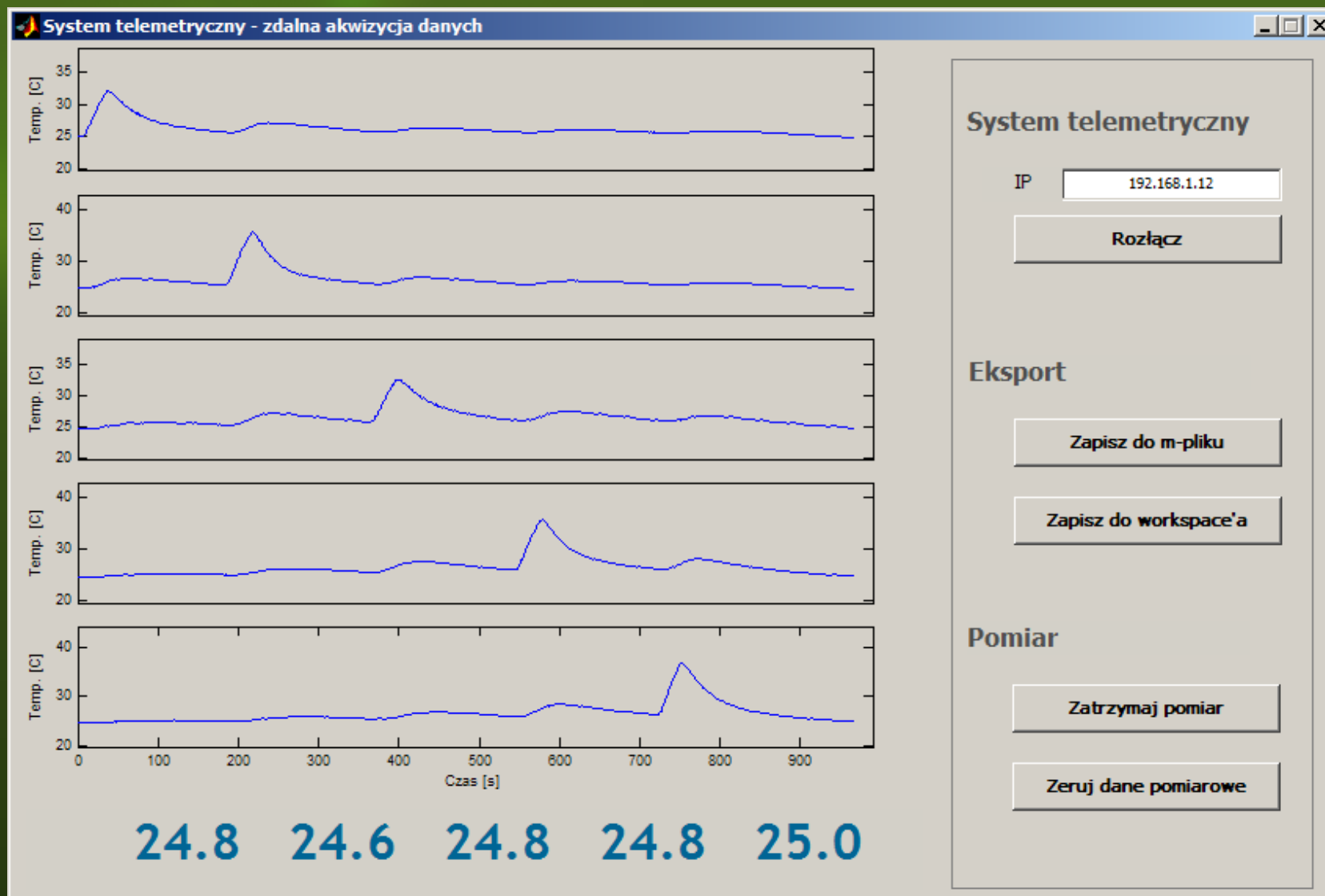
Implementacja w pakiecie Matlab



Implementacja w pakiecie Matlab

- Aplikacja typu DAQ (Data Acquisition) stworzona w graficznym środowisku GUIDE:
 - wizualizacja pomiarów
 - możliwość eksportu danych do plików mat oraz przestrzeni roboczej
 - wykorzystanie darmowego toolboxa pnet
- Zestaw m-skryptów:
 - identyfikacja parametrów macierzy G_C , G_M i G_P
 - wyznaczanie nastaw regulatorów PID (Ziegler-Nichols)
 - wyznaczanie wskaźników jakości regulacji
 - wizualizacja wyników

Identyfikacja modelu płytki grzewczej

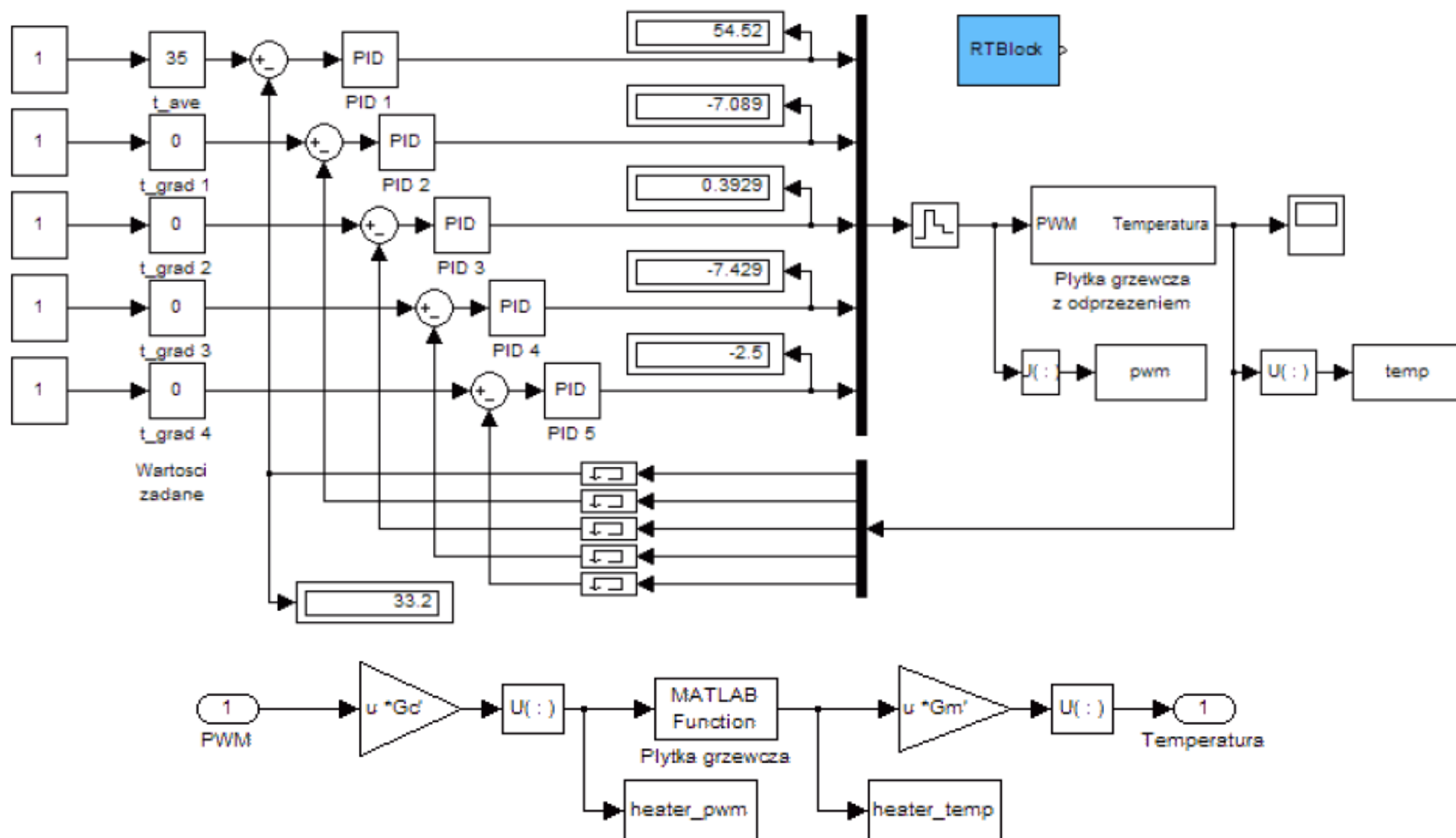


Identyfikacja modelu płytki grzewczej

$$G_p = \begin{bmatrix} 1.2789 & 1.0352 & 1.0039 & 1.0039 & 1 \\ 1.0524 & 1.3891 & 1.0195 & 1.0078 & 1.0039 \\ 1.0121 & 1.0435 & 1.2548 & 1.0269 & 1 \\ 1.004 & 1.02 & 1.0471 & 1.3383 & 1.0383 \\ 1 & 1.008 & 1.0118 & 1.0656 & 1.3992 \end{bmatrix}$$

$$G_c = \begin{bmatrix} 0.2415 & -2.8561 & -2.0547 & -1.2005 & -0.5921 \\ 0.1407 & 0.9486 & -1.4112 & -0.7419 & -0.371 \\ 0.2377 & 0.8694 & 1.763 & -1.4355 & -0.5545 \\ 0.157 & 0.5191 & 0.9187 & 2.0023 & -0.6817 \\ 0.1493 & 0.4768 & 0.7965 & 1.3343 & 2.1823 \end{bmatrix} \quad G_m = \begin{bmatrix} \frac{1}{5} & \frac{1}{5} & \frac{1}{5} & \frac{1}{5} & \frac{1}{5} \\ -1 & 1 & 0 & 0 & 0 \\ 0 & -1 & 1 & 0 & 0 \\ 0 & 0 & -1 & 1 & 0 \\ 0 & 0 & 0 & -1 & 1 \end{bmatrix}$$

Układ zamknięty w pakiecie Simulink



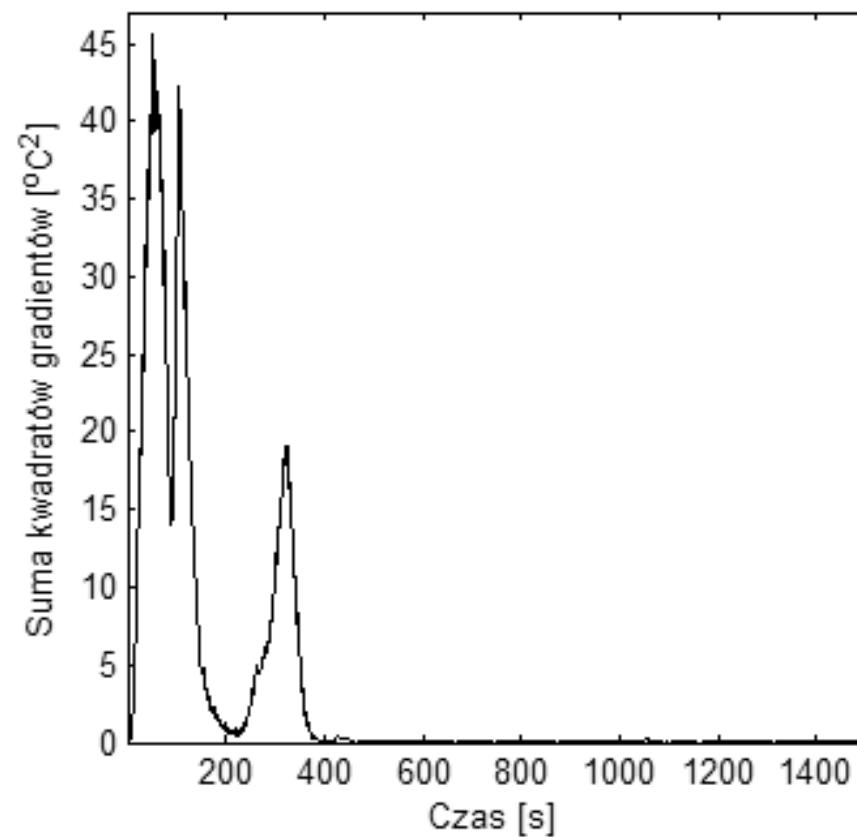
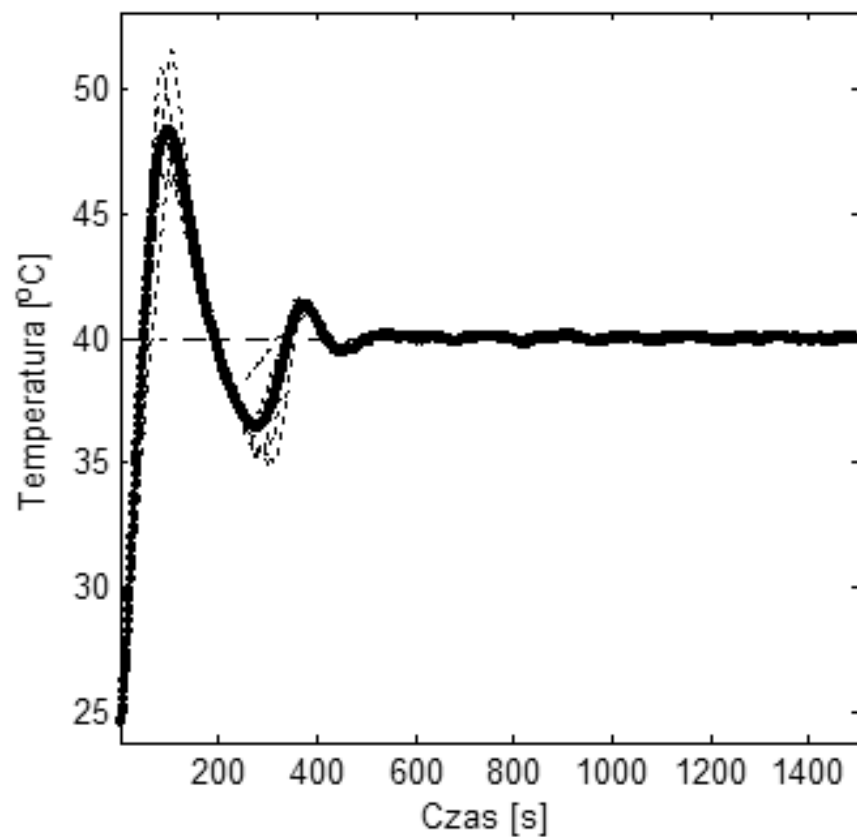
Badania algorytmu sterowania

- Zadania testowe:
 - skok wartości zadanej temperatury średniej 40°C
 - skok wartości zadanej temperatury średniej 60°C
 - skok wartości zadanej temperatury średniej z 60 do 40°C
 - skok wartości zadanej temperatury średniej 60°C przy zadanym gradiencie [1, 1, 1, 1]
- Wykorzystane wskaźniki jakości sterowania:
 - uchyb ustalony temperatury średniej
 - przeregulowanie w torze sterowania temperaturą średnią
 - czas regulacji temperatury średniej
 - kwadrat uchybu temperatury średniej (ISE)
 - suma kwadratów gradientów temperatur:

$$J_{grad} = \int_0^{\infty} \sum_{i=1}^{n-1} \nu_{grad(i)}(t)^2 dt$$

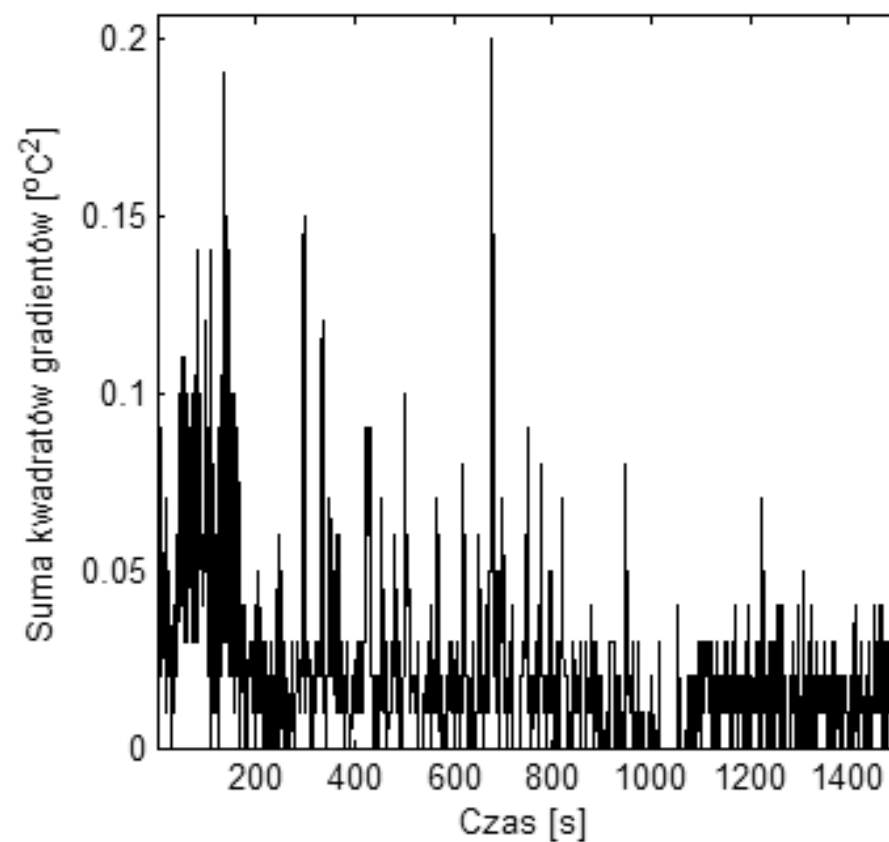
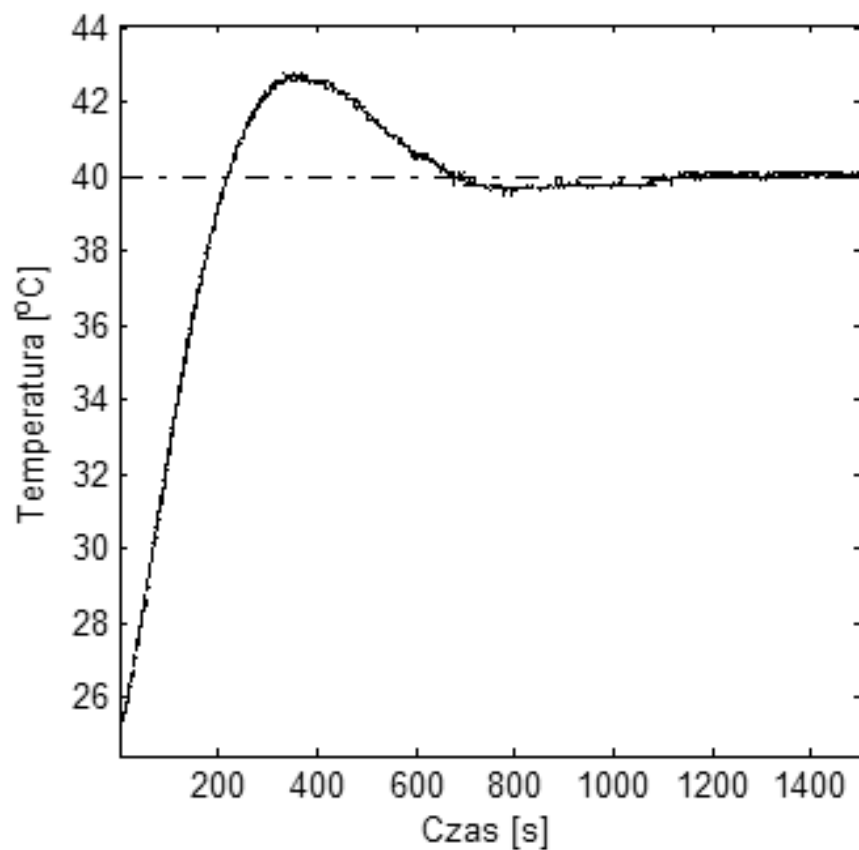
Przykładowe przebiegi

Algorytm „klasyczny”



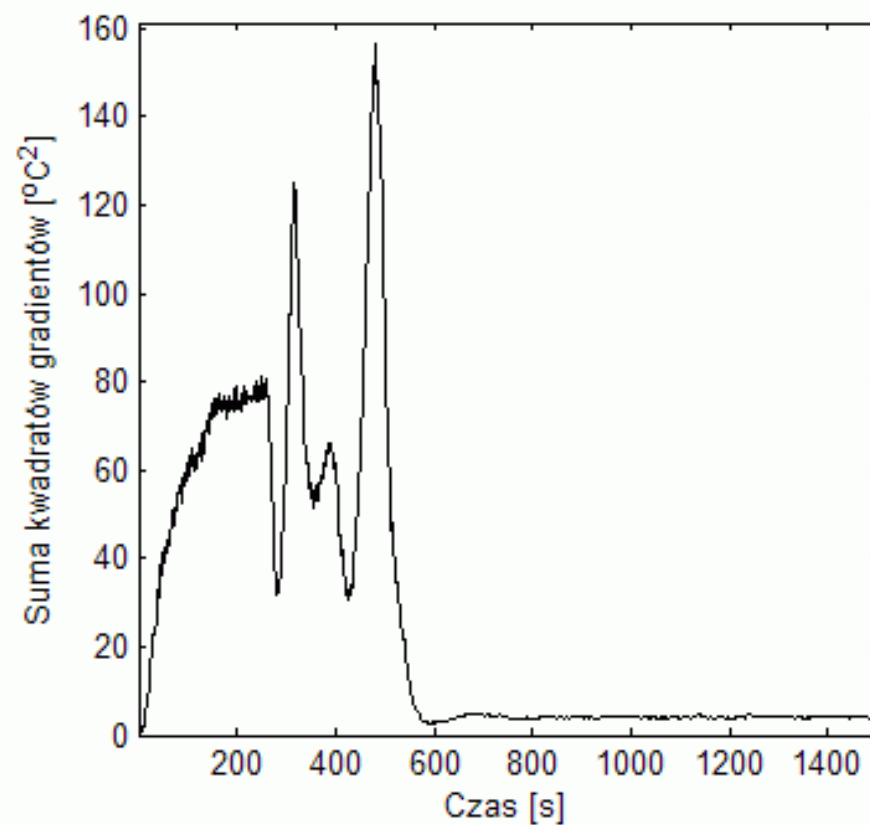
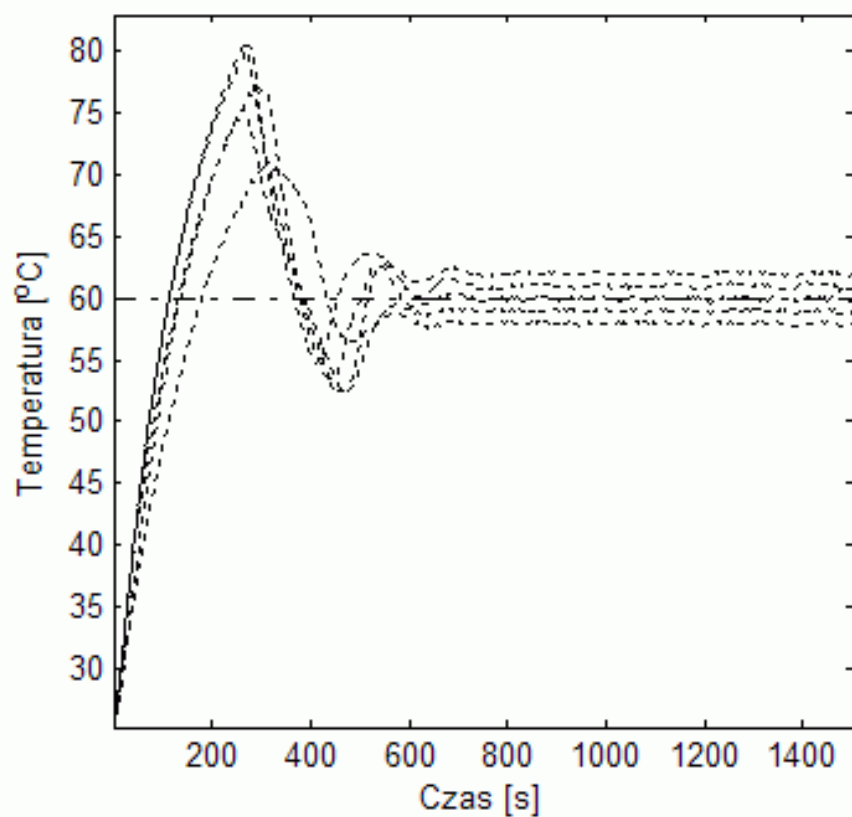
Przykładowe przebiegi

Algorytm sterowania gradientem



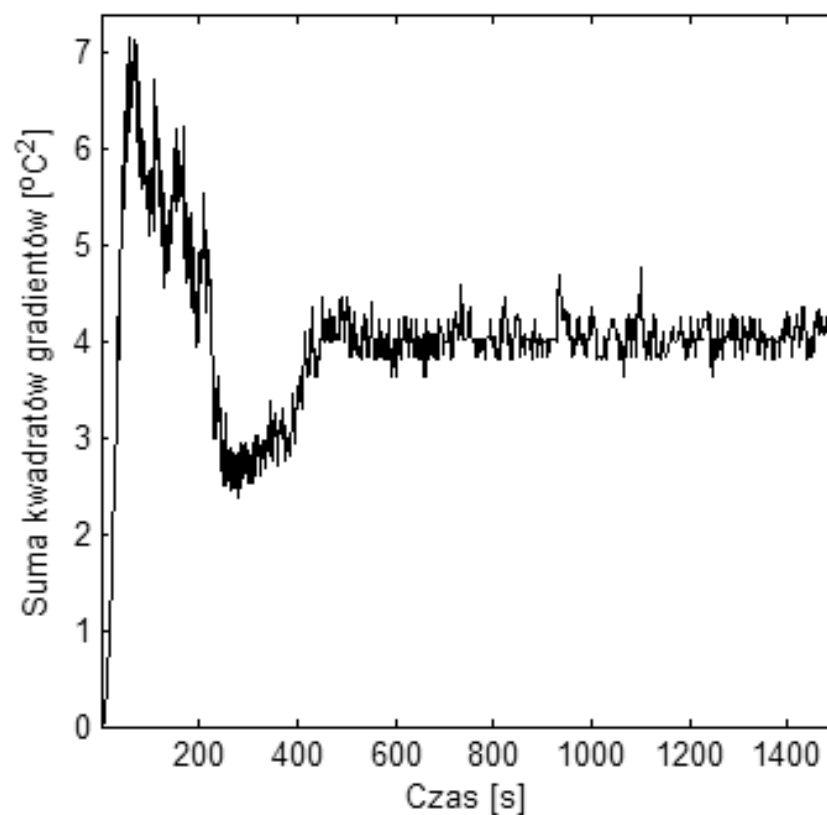
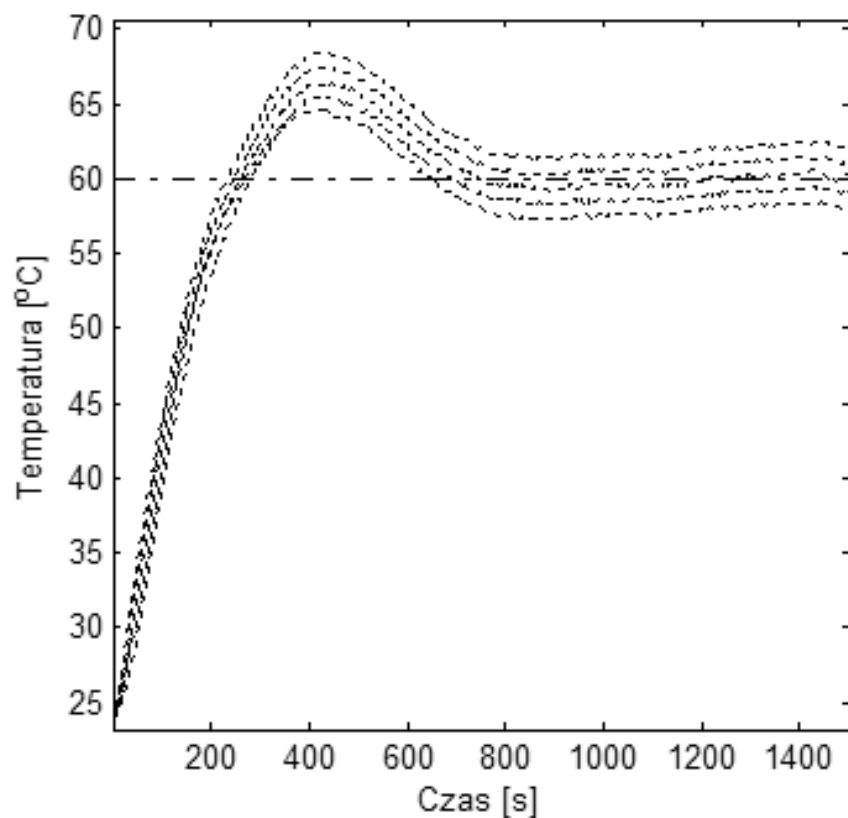
Przykładowe przebiegi

Algorytm „klasyczny”



Przykładowe przebiegi

Algorytm sterowania gradientem



Zestawienie wskaźników jakości

Tablica 5.3: Zestawienie wskaźników jakości regulacji dla algorytmu sterowania gradientem temperatury

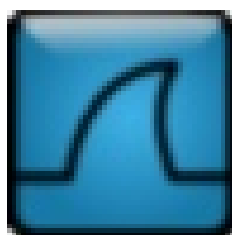
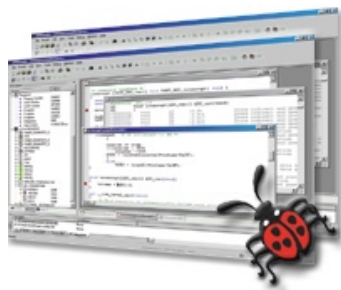
Zadanie	$e_{ave}(\infty)$	ϵ_{ave}	$t_{r,2\%ave}$	J_{ISEave}	J_{grad}
1	0	15%	868 s	17062,08	32,52
2	0	12%	994 s	97917,36	217,38
3	0	39%	1494 s	45279,99	398,55
4	0	11%	903 s	102456,07	307,28

Tablica 5.4: Zestawienie wskaźników jakości regulacji dla „klasycznego” algorytmu sterowania temperaturą

Zadanie	$e_{ave}(\infty)$	ϵ_{ave}	$t_{r,2\%ave}$	J_{ISEave}	J_{grad}
1	0	43%	485 s	10247,39	4695,44
2	0	29%	556 s	66548,38	35266,62
3	0	45%	804 s	30805,02	1012,38
4	0	29%	574 s	75307,58	32070,37

Wykorzystane narzędzia

- AVRstudio + WinAVR (środowisko programistyczne)
- Firefox + Firebug (przeglądarka/narzędzie webdeveloperskie)
- Wireshark (analizator pakietów)
- Eagle (projektowanie schematów ideowych i płytek)
- L^AT_EX + LEd (skład pracy)
- Subversion (zarządzanie kodem)



Dziękuję za uwagę