

Rysunek 1. Schemat stanowiska grzejąco-chłodzącego; zaznaczone elementy wykonawcze: wentylatory W1, W2, W3, W4, grzałki G1, G2, czujniki temperatury T1, T2, T3, T4, T5 (temperatura otoczenia), pomiar prądu P1, pomiar napięcia P2.

### Krótkie wprowadzenie do obsługi stanowiska grzejąco-chłodzącego z poziomu oprogramowania MATLAB

**Ogólny opis:** W ramach zajęć laboratoryjnych zespoły mają za zadanie zaprojektować, zaimplementować oraz dostroić algorytmy regulacji pod kątem wykorzystania ich do pracy ze stanowiskiem grzejąco-chłodzącym. Stanowisko grzejąco-chłodzące jest przykładowym procesem regulacji zaprojektowanym oraz wytworzonym w Instytucie Automatyki i Informatyki Stosowanej Politechniki Warszawskiej. Jest to obiekt o 6 wejściach (sygnałach sterujących):

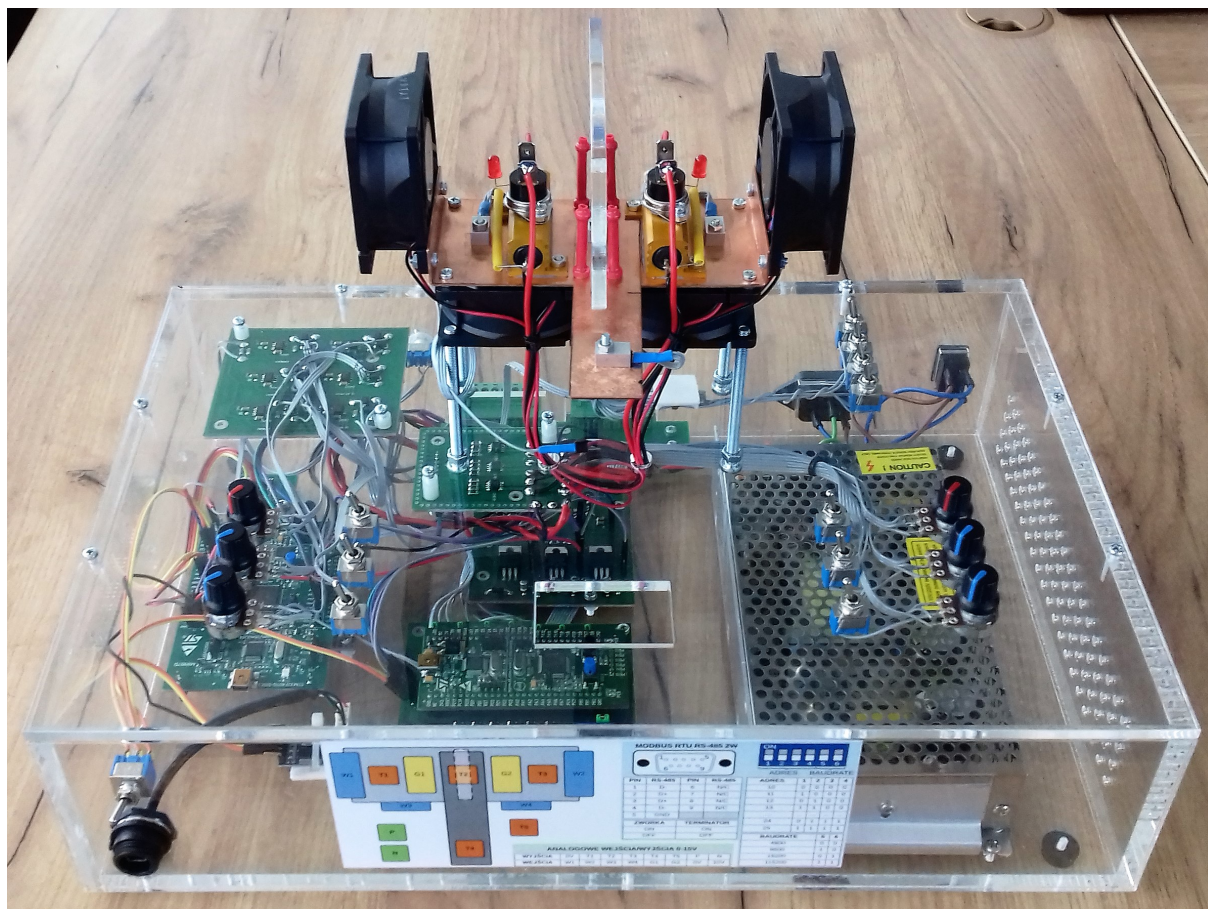
- wentylatory W1, W2, W3, W4,
- grzałki G1, G2.

Obiekt ten posiada następujące wyjścia (sygnały pomiarowe):

- pomiar temperatury stanowiska T1, T2, T3, T4,
- pomiar temperatury otoczenia T5,
- pomiar prądu P1,
- pomiar napięcia P2.

Schemat stanowiska został zaprezentowany na Rys. 1. Na Rys. 2 pokazano fizyczną postać stanowiska. Istnieją różne wersje stanowisk (wprowadzane były kolejne poprawki), co skutkuje tym, że ich wygląd może nie być jednolity. Nie mniej rozmieszczenie elementów jest identyczne na wszystkich stanowiskach.

Szczegóły dotyczące konstrukcji stanowiska oraz oprogramowania można znaleźć w rozszerzonej dokumentacji dostępnej u prowadzącego. Niniejszy dokument jest jednak



Rysunek 2. Zdjęcie stanowiska grzejno-chłodzącego; widok analogiczny do zaprezentowanego na schemacie na Rys. 1.

wystarczający do realizacji wszystkich zadań laboratoryjnych dotyczących implementacji w języku MATLAB.

**Warunki pracy ze stanowiskiem:** Podłączenie stanowiska wykonuje prowadzący zajęcia – studenci nie mogą zmieniać pozycji przełączników na przednim oraz tylnym panelu ani przełączać podłączonych kabli. Nie wolno również w żaden inny sposób modyfikować stanowiska.

Zakazane jest wkładanie palców w obszar roboczy wentylatorów, dotykane rozgrzanych grzałek lub okolicznych elementów gdyż grozi to uszkodzeniami ciała lub poparzeniem. Wszelkie uszkodzenia oraz problemy ze stanowiskiem należy niezwłocznie zgłaszać do prowadzącego zajęcia. Próby własnoręcznej naprawy są surowo zabronione!

Przekroczenie przez grzałki temperatury 90 °C powoduje odłączenie ich od zasilania, co może mieć istotny wpływ na późniejsze pomiary. Ponowne zasilenie grzałek następuje samoczynnie w momencie odpowiedniego spadku temperatury na grzałce. Ponieważ czujniki temperatury nie przylegają bezpośrednio do grzałki, ich pomiary mogą być znacznie niższe niż 90 °C w chwili wystąpienia powyższego zjawiska.

**Oprogramowanie do sterowania stanowiskiem:** Stanowisko zostało przygotowane jako elastyczne narzędzie do testowania algorytmów regulacji. Dlatego też istnieją różne możliwości komunikacji z nim, m.in. przy użyciu protokołu MODBUS RTU (RS485 2W), przy użyciu standardu napięciowego 0-10 V oraz przy użyciu standardu USB oraz oprogramowania MATLAB. To ostatnie podejście będzie głównie wykorzystywane w ramach dalszych zajęć.

W podanym przez prowadzących katalogu znajdują się pliki służące do sterowania stanowiskiem. Znajduje się tam również plik `MinimalWorkingExample.m`, który zawiera minimalny kod pozwalający w pełni korzystać ze stanowiska. Poniżej przedstawiona została jego treść (w poniższym kodzie zmodyfikowane zostały komentarze, aby ograniczyć długość linii kodu i zwiększyć czytelność):

```
1 function MinimalWorkingExample()
2     addpath('E:\SerialCommunication'); % add a path
3     initSerialControl COM6 % initialise com port
4
5     while(1)
6         %% obtaining measurements
7         measurements = readMeasurements(1:7); % read measurements
8
9         %% processing of the measurements
10        disp(measurements); % process measurements
11
12        %% sending new values of control signals
13        sendControls([ 1, 2, 3, 4, 5, 6],
14                     [ 0, 0, 0, 0, 0, 0]);
15
16        %% synchronising with the control process
17        waitForNewIteration(); % wait for new iteration
18    end
19 end
```

Rozpoczęcie korzystania z powyższego kodu wymaga znajomości dwóch parametrów. Pierwszym z nich jest wspomniana ścieżka do plików ze skryptami służącymi do obsługi stanowiska. Jest to także ścieżka do katalogu zawierającego powyższy skrypt. Tę właśnie ścieżkę (do katalogu, a nie do pliku!) należy wpisać w linii 2. jako argument funkcji `addpath`. Drugą istotną informacją jest numer portu COM, do którego podłączone jest stanowisko. Numer tego portu należy znaleźć w Menedżerze Urządzeń systemu Windows. Jest ona wpisana w nawiasie obok nazwy konwertera USB-UART (dokładna nazwa urządzenia może różnić się między wersjami stanowiska). Po szczegóły warto zgłosić się do prowadzącego zajęcia. W powyższym przykładzie stanowisko podłączone jest do portu COM6 – wartość tę należy zmienić w razie potrzeby.

**Funkcja do odczytu pomiarów:** Istotne z punktu widzenia realizacji zadania funkcje są kolejno wywoływane w powyższym skrypcie. Pierwszą z nich jest funkcja `readMeasurements`.

Jako argumenty przyjmuje ona numery identyfikacyjne sygnałów, których pomiary mają zostać odczytane. Jako argument można podać zarówno pojedynczą wartość jak i wektor wartości. W powyższym przykładzie podany został wektor zawierający wartości od 1 o 7 włącznie. Oznacza to, że odczytane zostaną pomiary sygnałów o indeksach od 1 do 7. Sygnały wyjściowe obiektu (wielkości mierzone) mają następujące numery identyfikacyjne:

1. T1 – temperatura wyrażona w °C,
2. T2 – temperatura wyrażona w °C,
3. T3 – temperatura wyrażona w °C,
4. T4 – temperatura wyrażona w °C,
5. T5 – temperatura otoczenia wyrażona w °C,
6. P1 – wartość prądu wykorzystywana przez stanowisko wyrażona w A,
7. P2 – wartość napięcia użyta do zasilania stanowiska wyrażona w V.

Funkcja ta zwraca odczytane pomiary w kolejności zgodnej z kolejnością sygnałów przekazanej w argumencie wywołania. Przykładowo wywołanie `readMeasurements([5,6,1])` zwróci wynik w postaci wektora pomiarów, gdzie kolejne elementy będą odpowiadały: pomiarowi temperatury otoczenia, pomiarowi wykorzystywanego przez stanowisko prądu oraz temperaturze zmierzonej przez czujnik T1.

**Funkcja do zmiany wartości sterowania:** Kolejną funkcją służącą do kontroli stanowiska jest funkcja `sendControls`, która przyjmuje dwa argumenty i nie zwraca żadnych wartości. Funkcja ta służy do zmiany wartości sygnałów wejściowych stanowiska (sygnałów sterujących). Jej konstrukcja pozwala na jednoczesną zmianę wielu wartości przy użyciu pojedynczego wywołania tej funkcji. Pierwszy jej argument zawiera wektor numerów identyfikacyjnych elementów wykonawczych, których wartości sygnałów użytkownik chce zmienić. Elementy wykonawcze posiadają następujące numery:

1. W1 – wentylator sterowany sygnałem 0-100 (co odpowiada mocy 0%-100% elementu),
2. W2 – wentylator sterowany sygnałem 0-100 (co odpowiada mocy 0%-100% elementu),
3. W3 – wentylator sterowany sygnałem 0-100 (co odpowiada mocy 0%-100% elementu),
4. W4 – wentylator sterowany sygnałem 0-100 (co odpowiada mocy 0%-100% elementu),
5. G1 – grzałka sterowana sygnałem 0-100 (co odpowiada mocy 0%-100% elementu),
6. G2 – grzałka sterowana sygnałem 0-100 (co odpowiada mocy 0%-100% elementu).

Drugim argumentem jest wektor nowych wartości sterowań. Wartości te podane są w kolejności odpowiadającej kolejności numerów identyfikacyjnych elementów przekazanych w pierwszym argumencie. Elementy nie wymienione w pierwszym argumencie pozostają bez zmian! Oznacza to, że wywołanie tej funkcji w powyższym przykładzie skutkować będzie wyłączeniem wszystkich elementów. Aby włączyć wentylator W1 na 20%, wentylator

W3 na 100% a grzałkę G1 na 50% pozostawiając pozostałe elementy wykonawcze w tym samym stanie co były, należy wywołać `sendControls([1,3,5],[20,100,50])`.

**Funkcja do oczekiwania na aktualizację stanu stanowiska:** Ostatnią funkcją wartą uwagi jest `waitForNewIteration`. Jest to funkcja nie przyjmująca żadnych argumentów oraz nie zwracająca żadnych wartości. Jej zadaniem jest odczekanie czasu potrzebnego stanowisku na pozyskanie oraz przygotowanie nowych pomiarów a także na aplikację wartości sterowań do kolejnych elementów wykonawczych. Oznacza to, że wywołanie `sendControls` nie powoduje natychmiastowej zmiany stanu stanowiska – zmiany te następują regularnie co 1 s. Aby zsynchronizować liczniki wykorzystane w ramach stanowiska ze skryptem napisanym w MATLAB-ie została wprowadzona niniejsza funkcja, która odczekuje czas pozostały do upłynięcia pozostałej części sekundy.