

System sterowania modelem elektrowni wiatrowej

Tomasz ŻENTARA¹, Jan CHUDZIKIEWICZ²

¹Instytut Teleinformatyki i Automatyki WAT,
ul. Kaliskiego 2, 00-908 Warszawa
tomasz.zentara@wp.pl

²Instytut Teleinformatyki i Automatyki WAT,
ul. Kaliskiego 2, 00-908 Warszawa
jan.chudzikiewicz@wat.edu.pl

STRESZCZENIE: W artykule przedstawiono opis projektu oraz implementację systemu sterowania dla modelu elektrowni wiatrowej. Zdefiniowano wymagania funkcjonalne i нефункционалне nakładane na system. Zaprezentowano diagramy interakcji wybranych przypadków użycia w notacji UML. Omówiono wybrane rozwiązania implementacyjne wykorzystane w projektowanym systemie. Całość ma stanowić stanowisko laboratoryjne wykorzystywane na zajęciach laboratoryjnych prowadzonych w Instytucie Teleinformatyki i Automatyki Wydziału Cybernetyki WAT.

SŁOWA KLUCZOWE: turbiny wiatrowe, Mała Elektrownia Wiatrowa (MEW), platforma Arduino, sterowniki PLC

1. Wprowadzenie

Zagadnienia związane z elektrowniami wiatrowymi są obecnie bardzo aktualne, ponieważ większość energii produkowanej w Polsce pochodzi z elektrowni opalanych złożami kopalnianymi, które z czasem się wyczerpią. Dlatego poszukiwane są zastępcze, a przede wszystkim odnawialne źródła energii. Dodatkowo Polska zobowiązała się wobec Unii Europejskiej, przyjmując w 2010 r. „Krajowy Plan Działania w zakresie energii ze źródeł odnawialnych”, że do roku 2020 osiągnięty zostanie 15% próg udziału odnawialnych źródeł energii w końcowym zużyciu energii [1], [2].

Mając powyższe na uwadze za cel realizacji, opisanego w artykule, „Systemu sterowania modelem elektrowni wiatrowej” było wykonanie

stanowiska laboratoryjnego, które może zostać użyte na zajęciach prowadzonych w Instytucie Automatyki i Teleinformatyki Wydziału Cybernetyki WAT. Zbudowanie takiego stanowiska daje możliwość zapoznania studentów z podstawami sterowania, budową modułów sterownia w tym oprogramowania takich modułów. Jako cel dalszy, postawiony przed jednym z Autorów, było porównanie dwóch dostępnych na rynku rozwiązań układowych do sterowania automatyką. Jedno z rozwiązań opierało się na ogólnie dostępnych (tanich), ale nie spełniają rygorystycznych norm dotyczących jakości, modułach Arduino, drugie opierało się na sterownikach reprogramowalnych PLC, które spełniają normy eksploatacyjne, lecz są drogie w zakupie [3], [4].

2. Projekt systemu sterowania Małą Elektrownią Wiatrową

Każdorazowe zaprojektowanie jakiegokolwiek systemu, w tym systemu sterowania wiąże się z przejściem projektanta przez kilka faz projektowych. W tym wypadku projekt systemu sterowania małą elektrownią wiatrową wiązał się z wykonaniem modelu turbiny wiatrowej, zaprojektowaniem i zbudowaniem elektronicznej części systemów sterowania, jak również napisaniem oprogramowania sterującego dla wykonanych systemów. W niniejszym rozdziale przedstawiony zostanie pełen przebieg realizacji procesu projektowania oraz budowy systemu sterowania dla modelu MEW.

2.1. Wymagania funkcjonalne nakładane na system pod kątem zastosowań w dydaktyce

Pierwszą czynnością, która została wykonana jeszcze przed przystąpieniem do projektowania i realizacji modelu oraz systemów sterowania, była analiza zadań pod kątem zastosowań w dydaktyce. W trakcie analizy wyszczególnione zostały główne wymagania funkcjonalne, które powinny spełniać elementy znajdujące się w projekcie tj. makieta elektrowni oraz systemy sterowania. Główne wymagania, które wzięto pod uwagę wyszczególnione zostały w Tabeli 1.

Tabela 1. Tabela wymagań funkcjonalnych stawianych na projekt MEW

ID Wymagania	Opis
REQ001	Makieta elektrowni powinna zajmować mało miejsca na stanowisku laboratoryjnym.
REQ002	Makieta elektrowni musi zapewnić odpowiednią ochronę życia i zdrowia użytkownika.

REQ003	Makieta powinna być zasilana napięciem stałym, wartości nieprzekraczającej 24 V.
REQ004	Makieta powinna gwarantować na wyjściu napięcie stałe o minimalnej wartości 12 V
REQ005	System sterowania powinien umożliwiać pomiar podstawowych parametrów cyrkulatora i turbiny wiatrowej.
REQ006	System sterowania musi umożliwiać regulację napięcia na cyrkulatorze.
REQ007	System sterowania powinien być chroniony przed niepowołaną ingerencją osób trzecich.
REQ008	System sterowania powinien zapewniać warunki BHP i ochrony przeciwporażeniowej.
REQ009	System sterowania powinien być modułowy, aby zapewnić skalowalność i możliwość dalszej rozbudowy.
REQ010	Użytkownicy powinni posiadać możliwość wgrywania oprogramowania na sterownik.
REQ011	System sterowania powinien komunikować się z aplikacją monitorującą jego pracę.
REQ012	Aplikacja powinna posiadać przejrzysty interfejs użytkownika.
REQ013	Użytkownik powinien posiadać możliwość sterowania parametrami MEW z poziomu aplikacji komputerowej.

2.2. Model elektrowni wiatrowej

Niestety wykonanie pełnowymiarowego modelu elektrowni wiatrowej nie jest możliwe z dwóch głównych powodów: po pierwsze z przyczyn finansowych, a po drugie z przyjętych założeń, iż model spełniać ma zadania stanowiska laboratoryjnego używanego przez studentów. Finalnie zbudowany został tunel, do którego dołączone zostały dwa bezszczotkowe wentylatory osiowe. Pierwszy z nich służył, jako cyrkulator, czyli urządzenie do sterowanego wymuszania przepływu powietrza (określanego w tym przypadku mianem wiatru), drugi zaś był odbiornikiem wiatru, tj. turbiną wiatrową, tudzież generatorem.

2.2.1. Projekt 3D modelu MEW w AutoCad

Przed przystąpieniem do fizycznej realizacji modelu elektrowni wiatrowej

w programie AutoCad 2013 wykonany został rysunek przedstawiający trójwymiarowy widok modelu (Rysunek 1). Dzięki takiemu rozwiązaniu możliwa była konsultacja proponowanego pomysłu, co znacząco uprościło późniejszy proces produkcyjny gotowej makiety.

Projekt v.1 – Tunel z pleksiglasu

W pierwszej wersji zaprojektowany został tunel aerodynamiczny, który swoimi rozmiarami odwzorowywał obudowy urządzeń automatyki znajdujące się w laboratorium Instytutu Teleinformatyki i Automatyki WAT. Rozwiązanie pozwalało na modyfikacje elementów modelu i integrację z wykorzystywanymi na laboratoriach modułami. Wentylatory, dla zapewnienia odpowiedniego bezpieczeństwa, miały być schowane do wnętrza obudowy i zabezpieczone odpowiednimi osłonami. Na górnej pokrywie modelu umieszczono wąż, którego regulowana wysokość odpowiadałaby wymuszonemu upływowi powietrza, przez co studenci mierzyliby bezwładność układu sterowania.

Rysunek 1 przedstawia opisywany wcześniej model 3D - makiety elektrowni wiatrowej. Na rysunku widać, że wewnątrz obudowy, przy wentylatorach, znajdują się dodatkowe ścianki z pleksiglasu. Takie rozwiązanie zapewnia większą szczelność w środku obudowy, jak również stanowi dodatkowy punkt podparcia dla wentylatorów osiowych, które ważą około 1 kg każdy. Wspomniane ścianki działowe nie są przymocowane na stałe do obudowy, dzięki czemu w późniejszym procesie eksploatacji istniałaby możliwość wymiany uszkodzonych wirników na nowe, które posiadałyby standardowy rozstaw otworów mocujących, lecz grubszy korpus.

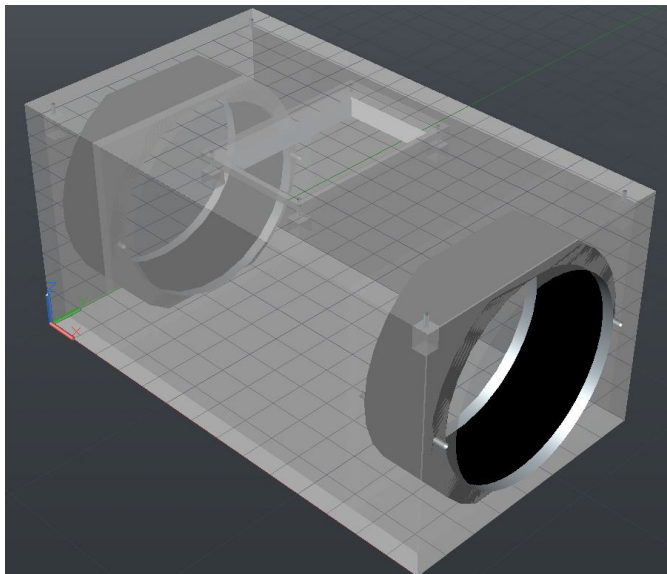
Niestety projekt obudowy okazał się drogi w wykonaniu, więc jego dalsza realizacja została zwieszona, a całą dokumentację zachowano, co daje możliwość wznowienia projektu w przypadku wygospodarowania funduszy na jego wykonanie.

Projekt v.2 – Tunel z węża technicznego

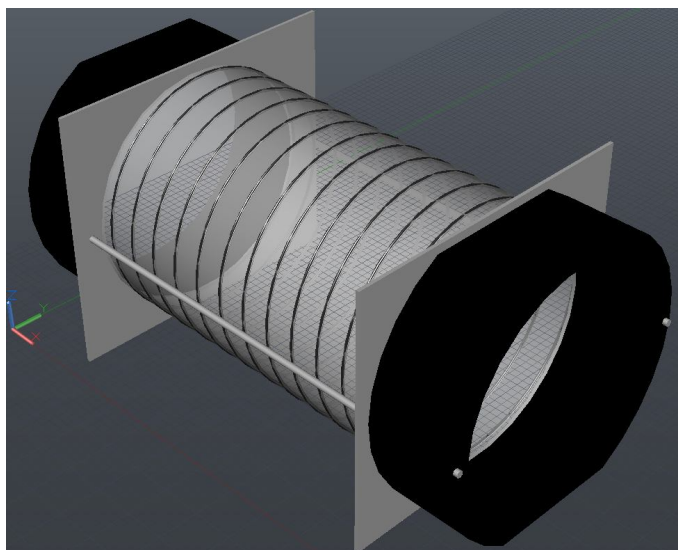
W drugiej odsłonie tunel aerodynamiczny został nieco zmodyfikowany. Usunięto pleksiglasowe, zewnętrzne elementy obudowy, zaś podstawki z pleksi znajdujące się między wentylatorami zastąpiono innym materiałem, a pomiędzy wentylatorami rozciągnięto miękki rękaw, którego głównym zadaniem jest możliwie jak największe ukierunkowanie strugi wdmuchiwanego powietrza do wentylatora turbiny wiatrowej. Całość konstrukcji wzmocniono metalowymi prętami w celu zapewnienia sztywności makiety, wycentrowania znajdujących się przeciwległe wirników, jak również wyprostowania elastycznego rękawa.

Rysunek 2 przedstawia model 3D drugiej wersji makiety elektrowni wiatrowej. Na rysunku widać, że wentylatory osiowe znajdują się na zewnątrz, a między nimi rozciągnięty jest sprężysty rękaw uszczelniający. Należy zwrócić uwagę, że tak zaprojektowany model zapewnia mniejszą ochronę i narażony jest

bardziej na niepowołane manipulacje ze strony osób trzecich, dlatego podobnie jak w wersji poprzedniej na końcach wirników umieszczono metalowe grille, zabezpieczające przed np. włożeniem do środka większych przedmiotów.



Rysunek 1. Pierwsza wersja projektu MEW wykonana w AutoCad 2013

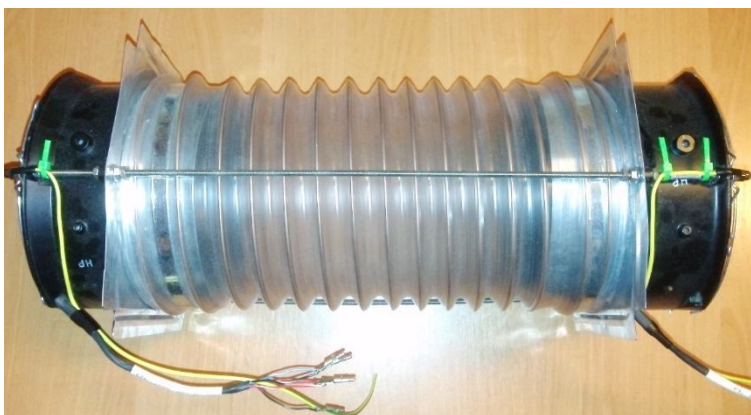


Rysunek 2. Druga wersja projektu MEW wykonana w AutoCad 2013

2.2.2. Realizacja modelu elektrowni wiatrowej

Do budowy makiety użyte zostały dwa bezszczotkowe wentylatory osiowe o wymiarach 150x172x50 mm (WxHxD) i napięciu zasilania równym 24 V. Zapewniają one przepływ strumienia powietrza rzędu 400 m³ na godzinę przy obrotach wirnika w okolicy 4000 obr/min i maksymalnym zużyciu prądu wynoszącym ok. 2 A. Dodatkowo dzięki wbudowaniu czujnika Halla umożliwiają one pomiar prędkości obrotowej wirnika, co w budowanym modelu wymagało zastosowania dodatkowego rozwiązania. W przypadku wentylatora turbiny wiatrowej obracający się z pełną prędkością wirnik wytwarzał na wyjściu napięcie stałe wynoszące zaledwie 5 V, które nie pozwalało dokładnie symulować działania turbiny wiatrowej. Przyczyną takiego stanu rzeczy był moduł falownika, którego zadaniem jestysterowanie nieruchomych cewek tak, aby wprowadzić w ruch obrotowy magnes. Problem udało się rozwiązać poprzez bezpośrednie wpięcie się w uzwojenie cewek. Ponieważ dodatkowo użyte wentylatory okazały się dwufazowe, połączono cewki szeregowo co pozwoliło na zwiększenia napięcia wyjściowego. Po tak wykonanych modyfikacjach wentylator turbiny wiatrowej generował na wyjściu napięcie przemiennego o wartości ok. 21 V.

Wzorując się na drugiej wersji projektu makiety MEW najlepszą konstrukcją nośną dla wentylatorów okazały się kratki kominkowe o wymiarach 15x15 cm i średnicy otworu 150 mm. Całość osadzono na czterech metalowych gwintowanych szpilkach m4 o długości 410 mm każda, które regulują odstęp między wiatrakami. Pomiędzy kratkami rozciągnięto wąż techniczny o średnicy równej średnicy otworu kratki i zaciśnięto go przy pomocy regulowanej metalowej opaski. Efekt końcowy pracy widoczny jest na Rysunek 3.



Rysunek 3. Zbudowana makieta elektrowni wiatrowej

2.3. System sterowania modelem elektrowni wiatrowej

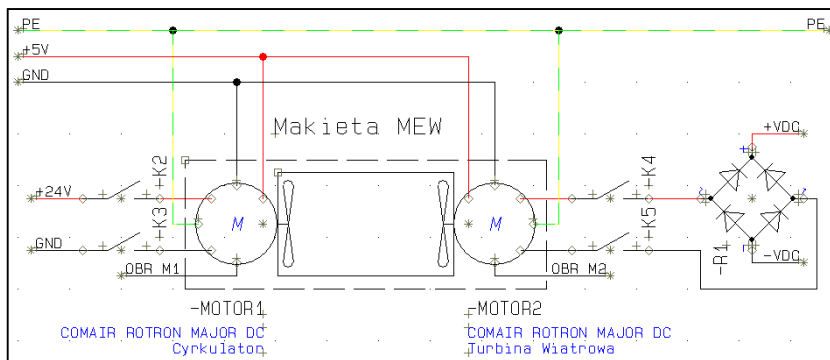
W niniejszym podrozdziale przedstawiona zostanie koncepcja systemów sterowania MWE [5], [6]. W projekcie modelu występuje jeden układ elektryczny, do którego podłączone zostały dwa niezależne układy sterowania: zestaw uruchomieniowy Arduino oraz system sterowników reprogramowalnych PLC. Wykonanie dwóch układów sterowania, które dublują swoje funkcje, wynika z możliwości użycia modelu na różnych zajęciach prowadzonych w Instytucie. Dodatkowo platforma Arduino realizuje dwa zadania: monitorowania urządzeń PLC oraz mostu łączącego elektronikę sterującą MEW z aplikacją działającą na komputerze PC.

2.3.1. Projekt systemu sterowania

Analizując konstrukcję makiety MEW, wyszczególnić możemy dwa niezależne podsystemy sterowania:

- cyrkulatora,
- turbiny wiatrowej.

W układzie zastępczym, którego schemat przedstawiono na Rysunek 4, wyszczególnić możemy elementy dla: cyrkulatora i turbiny wiatrowej. Moduły te stanowią podstawę do komponowania elementów składowych systemów sterowania, a w przypadku sterownika PLC pozwolą określić ilość modułów wejść/wyjść potrzebnych do realizacji zadania.



Rysunek 4. Schemat zastępczy układu elektrycznego MEW

2.3.2. Projekt zasilania z sieci energetycznej

Aby użyć do zasilania układów napięcia z jednofazowej sieci energetycznej (230V/50Hz), konieczne było zaprojektowanie obwodów podziału

napięcia zasilania na odbiorniki, które dodatkowo zabezpieczałyby użytkownika przed porażeniem. Koncepcja zakładała zbudowanie dwóch paneli elektrycznych, z których pierwszy zawierał bezpieczniki i wykonywał zadania rozdzielni elektrycznej, a drugi mieścił zespół trzech zasilaczy impulsowych.

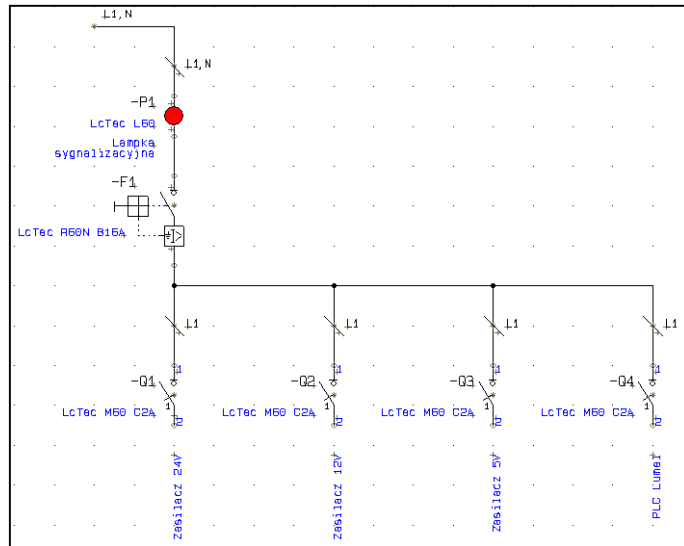
W pierwszej kolejności wykonany został projekt rozdzielni elektrycznej. Do jego realizacji wykorzystano oprogramowanie PCSchematic Automation 16.0 w wersji demonstracyjnej, w którym narysowany został uproszczony (jednokreskowy) schemat rozdzielni elektrycznej (Rysunek 5). W trakcie budowy użyte zostały cztery wyłączniki nadprądowe o wartości progowej 2A, które zabezpieczały trzy obwody zasilaczy impulsowych i obwód urządzeń PLC. Dzięki takiemu rozwiązaniu odłączenie jednego obwodu nie spowoduje rozłączenia pozostałych, a użytkownik jest w stanie oddzielnie włączać/wyłączać każdy z zasilaczy.

Dodatkowo na Rysunek 5 widać, że powyżej bezpieczników znajduje się wyłącznik różnicowoprądowy o wartości 16A, który spełnia rolę kolejnego poziomu zabezpieczeń nadprądowych, a także chroni użytkownika przed porażeniem elektrycznym, automatycznie odłączając obwód, gdy pomiędzy fazą, a przewodem neutralnym powstanie uchyb większy niż 30 mA (tzw. zwarcie). Ostatnim elementem obwodu rozdzielni jest czerwona lamka sygnalizująca podłączenie urządzenia do sieci pod napięciem.

Bezpieczniki zamontowane zostały w zakupionym panelu dystrybucji napięć ARADIN1 dostarczonym przez firmę Pulsar, który montuje się w standardowej szafie typu RACK 19" i pomieścić może 24 pojedyncze moduły wyłączników. Panel umieszczono wewnątrz szafy RACK 19" o wysokości 10 U.

Przy pomocy oprogramowania Automation wykonany został przykładowy widok rozdzielni (panel dystrybucji napięć zamieszczono na Rysunek 6), który sugeruje rozmieszczenie elementów na szynie DIN (TH35), jak również sposób poprowadzenia, podłączenia, a także średnicę użytych przewodów itp.

Niestety baza aparatury w demonstracyjnej wersji oprogramowania nie obejmuje wykorzystanych do realizacji wyłączników i lampki firmy LcTec, dlatego użyte zostały przykładowe symbole i kształty znajdujące się w bazie, które najwierniej odzwierciedlają kształty posiadanej aparatury.



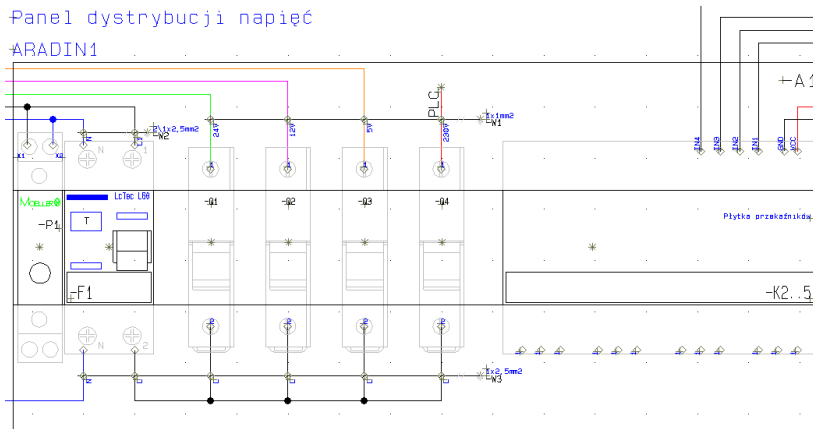
Rysunek 5. Jednokreskowy schemat rozdzielni elektrycznej dla SS MEW

Dbając o bezpieczeństwo użytkowników, w projekcie użyte zostały trzy impulsowe zasilacze przemysłowe firmy Mean Well o różnych napięciach wyjściowych, montowane na szynę TH35, które mają wbudowanych szereg zabezpieczeń w tym: zwarciovowe, przeciążeniowe i nadnapięciowe. Dodatkowo każdy z nich ma wbudowaną funkcję miękkiego startu niwelującą prąd udarowy oraz wyjście na diodę sygnalizacyjną, którą można wyprowadzić na panel czołowy szafy sterowniczej.

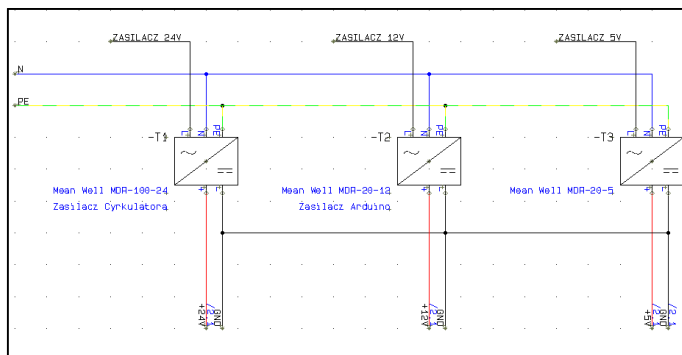
Przed przystąpieniem do montażu elementów wykonany został schemat zastępczy układów, który przedstawia sposób podłączenia zasilaczy do rozdzielni elektrycznej. Należy zaznaczyć, że widoczny na Rysunek 7 przewód oznaczony, jako neutralny N podłączony jest do wyjścia wyłącznika różnicowoprądowego.

Wyjścia mas zasilaczy zostały połączone w celu wyrównania punktu odniesienia dla wszystkich napięć generowanych przez zasilacze.

Podobnie jak w przypadku rozdzielni elektrycznej, tak i tutaj wykonany został widok rozmieszczenia elementów na panelu zasilaczy gdzie, jako podstawę użyta została blacha montażowa Pulsar RADIN o szerokości 19” z dołączoną szyną TH35, na którą doczepione zostały elementy systemu. Blacha, z racji oszczędności miejsca, wstawiona została w tylną część dwudzielnej szafy RACK.



Rysunek 6. Widok panelu dystrybucji napięć SS MEW opartego o schemat jednokreskowy

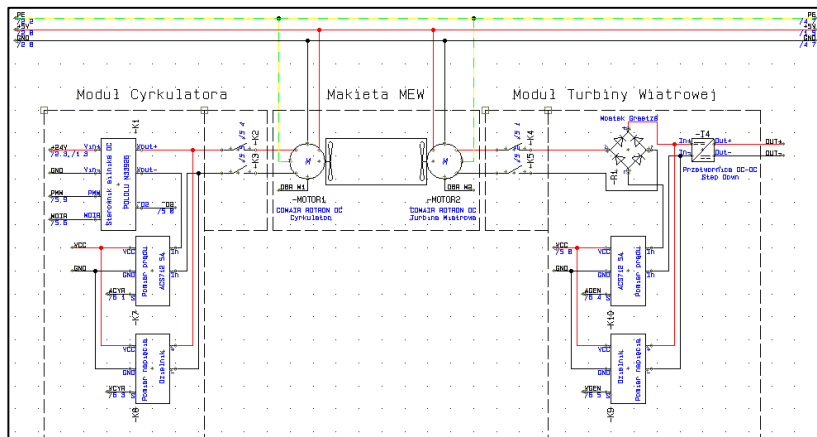


Rysunek 7. Schemat zasilania dla SS MEW

2.3.3. System sterowania oparty o Arduino

Jako element bazy do rozbudowy systemu sterowania opartego o mikrokontroler Arduino wykorzystany został model zastępczy makiety MEW przedstawiony na Rysunek 4. Do makiety dołożone zostały dodatkowe elementy sterujące i pomiarowe, których podłączenie widać na Rysunek 8.

Ponieważ napięcie generowane na wyjściu cyrkulatora zmienia się w zakresie 0 – 25 VDC, a napięcie wejściowe, akceptowalne przez wejścia analogowe układu Arduino, nie może przekraczać 5 VDC, nie było możliwe bezpośrednie podłączenie się do mikrokontrolera. W celu obniżenia generowanego napięcia do wymaganej przez wejścia analogowe Arduino wartości, zastosowany został gotowy moduł dzielnika napięcia.



Rysunek 8. Schemat systemu sterowania MEW

Pomiar natężenia przepływającego prądu nie jest możliwy bezpośrednio dla układów Arduino, dlatego wykorzystany został gotowy moduł pomiarowy, który umożliwia proporcjonalne przekształcenie przepływającego prądu, na napięcie wyjściowe o wartości w przedziale 0 – 5 V. Moduł wykorzystuje układ elektroniczny ACS712, który działa na zasadzie wykorzystania efektu Hall'a i galwanicznie separuje obwód pomiarowy od obwodu Arduino, dzięki czemu możliwe jest jego bezpośrednie podłączenie do wejścia analogowego płytki Arduino.

Sterowanie silnikiem zasilanym napięciem 24 VDC, który przy osiągnięciu maksymalnych obrotów pobiera prąd rzędu 2 A, nie może być realizowane bezpośrednio z wyjść modułu Arduino pomimo, że moduł Arduino udostępnia 8 bitowe wyjście PWM (ang. *Pulse Width Modulation*), za pomocą którego można sterować pracą silników prądu stałego. Przyczyną takiego stanu rzeczy jest niska obciążalność prądowa tych wyjść, która wynosi kilkadziesiąt miliamper przy napięciu nieprzekraczającym wartości 5 VDC. W tym przypadku koniecznym było zastosowanie zewnętrznego sterownika silników DC. Jednym z dostępnych na rynku tego typu rozwiązań jest, zastosowany w omawianym modelu, sterownik Pololu MC33926. Udostępnia on szeroki zakres napięcia wyjściowego (5 – 28 V) i może być obciążany prądem do 3 A, co pokrywa wymagania stawiane przez wykorzystane wentylatory. Sygnałem wejściowym dla sterownika Pololu jest sygnał zadawany z wyjścia PWM modułu Arduino, dzięki czemu istnieje możliwość modyfikacji prędkości obrotowej silnika z poziomu aplikacji współpracującej z modułem Arduino.

Napięcie zasilające silniki wentylatorów podawane jest poprzez styki zwierne przekaźników, których cewki sterowane są za pomocą cyfrowych wyjść Arduino. Przekazniki pełnią rolę zabezpieczającą, rozłączając obwody zasilania w przypadku awarii lub jakiegokolwiek innej usterki technicznej.

Aby umożliwić szybki demontaż (modyfikację) elementów sterujących zostały one pogrupowane w moduły i umieszczone w osobnych obudowach. Do tego celu wykorzystane zostały obudowy z tworzywa sztucznego, które można zamontować na szynie TH35. Wierzchnia pokrywa obudowy zastąpiona została 3 mm płytką z laminatu bez miedzi, do której na słupkach dystansowych przykręcono elementy elektroniczne. Tak powstałe moduły umieszczono wewnątrz obudowy ARADIN3 firmy Pulsar, którą później zamontowano w szafie RACK 19”.

2.3.4. System sterowania dla platformy PLC

Projekt systemu sterowania dla platformy sterowników PLC został oparty, podobnie jak dla Arduino, o schemat zastępczy układu MEW.

W pierwszej kolejności należało zapoznać się z ofertą sprzętową firmy LUMEL, której urządzenia zostały zagwarantowane przez uczelnię do realizacji pracy. Po analizie potrzeb i możliwości wybrano moduły, które pokrywały pełne spektrum działania na potrzeby realizacji pracy [2].

Kolejnym krokiem było narysowanie schematu systemu sterowania, który podobny jest do opisywanego wcześniej schematu systemu Arduino, jednakże sygnały sterujące pochodzą z innych urządzeń.

Przyjęte założenia projektowe wymagały realizacji pomiaru wartości napięcia stałego w zakresie od 0 do 24 V i pomiaru natężenia prądu maksymalnie do 3 A. Pomiar tych wartości realizowany jest przez moduły wejść analogowych, dla których to wejść (zgodnie z instrukcją obsługi) maksymalna wartość napięcia wynosi 10 V, a prądu nie może przekraczać 20 mA. Problem związany z tymi ograniczeniami rozwiązano w oparciu o elementy zastosowane dla modułów Arduino. Zastosowanie rozwiązania dla Arduino pozwoliło na obniżenie wartości napięcia do poziomu 5V, natomiast moduł pomiaru prądu przekształca mierzoną wartość do poziomu zalecanego dla modułu wejść analogowych SM2 firmy LUMEL.

Pozostała infrastruktura systemu sterowania platformy Arduino wykorzystana została do zmniejszenia ilości poprowadzonych przewodów oraz uproszczeniu konstrukcji. Rozwiązało to również problem sterowania silnikiem DC, gdyż wybrane moduły firmy LUMEL nie dają możliwości sterowania napięciem na cyrkulatorze i nie mają wyjścia PWM pozwalającego na podłączenie się do sterownika silników DC firmy Pololu. W tym przypadku do sterowania użyty został port szeregowy RS-232, który posłużył, jako most pomiędzy sterownikiem SMC, a zestawem uruchomieniowym Arduino MEGA 2560. Komunikaty sterujące ze sterownika PLC, poprzez moduł Arduino, którego zadaniem jest ich dekodowanie, przesyłane są do silnika DC.

Urządzenia firmy LUMEL umieszczone zostały na szynie TH35

znajdującej się na płycie RADIN umieszczonej w głębi obudowy RACK 19", tuż obok zasilaczy impulsowych.

3. Projekt oprogramowania systemu sterowania MEW

W niniejszym rozdziale przedstawiono projekt oprogramowania, przeznaczonego do systemu sterowania, które omówiono była w poprzednim rozdziale.

Do realizacji procesu sterowania utworzono trzy odrębne programy:

- dla platformy Arduino MEGA 2560, przy pomocy darmowego oprogramowania Arduino i języka zbliżonego składnią do C [4];
- dla sterownika PLC, przy pomocy oprogramowania CPDev dostarczanego przez producenta i języka ST [3];
- dla komputera PC, przy pomocy oprogramowania Visual Studio 2012 i języka C# platformy .Net [7].

Pierwsze dwa wyżej wymienione programy pozwalają na sterowanie urządzeniami wbudowanymi, które dublują swoje funkcje względem układu sterowania. Programy osadzone w pamięci wewnętrznej jednostek sterujących, poprzez zaprogramowanie mikroprocesorów. Aplikacja służąca do monitorowania parametrów MEW i sterowania makieta elektrowni uruchamiana jest z poziomu komputera PC.

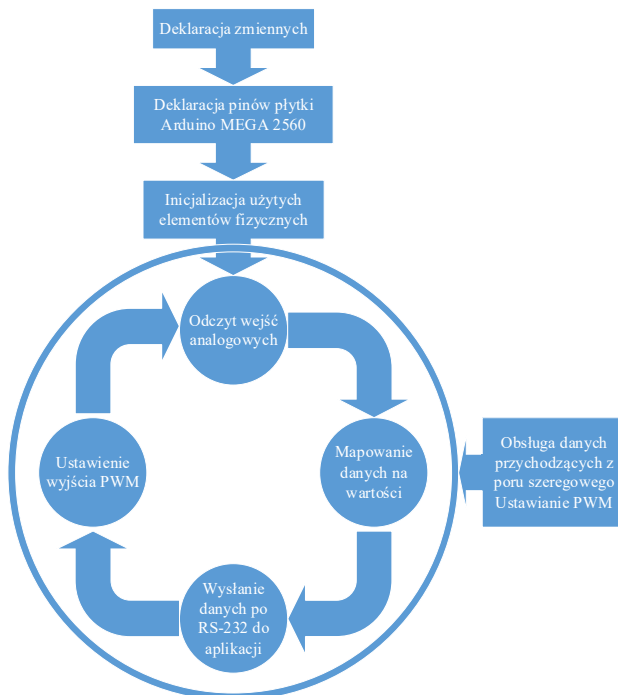
3.1. Projekt oprogramowania dla platformy uruchomieniowej Arduino

W trakcie projektowania oprogramowania należało pamiętać, iż mikrokontrolery układów wbudowanych po włączeniu zasilania i przejściu inicjalizacji działają cyklicznie, tj. kod programu umieszczony w pętli głównej wykonywany jest nieustannie w taktach określonych przez zegar kwarcowy umieszczony na płycie PCB.

W przypadku oprogramowania MEW (od strony Arduino) zaproponowany został cykl przetwarzania mikroprocesora przedstawiony na Rysunek 9.

Na Rysunek 9 widać, iż pierwszą czynnością programisty było zadeklarowanie zmiennych globalnych, które wykorzystywane będą w całości programu do przechowywania wartości odczytanych z czujników zewnętrznych, jak również danych przekazanych przez port szeregowy. Zmiennym nadano zerową wartość początkową, aby w trakcie uruchamiania zestawu nie doszło do zaburzeń pracy systemu, co spowodować może np. wymuszenie na wentylatorze

osiągnięcia maksymalnej prędkości obrotowej i wynika z losowych wartości znajdujących się w rejestrach.



Rysunek 9. Cykl pracy programu dla sterownika Arduino MEGA 2560

Kolejnym niecyklicznym etapem działania programu, było zadeklarowanie zmiennych przedstawiających numery pinów użytych do sterowania systemem. Deklaracja pinów określona może być na dwa sposoby: poprzez nadanie zmiennej wartości liczbowej odpowiadającej numerowi wyjścia/wejścia, bądź poprzez podanie nazwy wykorzystywanego pinu (zgodnie z nomenklaturą nazewniczą płytki Arduino).

Trzecim i ostatnim niecyklicznym etapem jest inicjalizacja. Dla systemu sterowania Arduino odnosi się do: zainicjalizowania komunikacji na porcie szeregowym RS-232, wprowadzenia cyfrowych wejść/wyjść w odpowiedni tryb działania (tj. wyjście) oraz nadania stałych wartości (poziom wysoki, czyli logiczna jedynka) dla wartości odpowiadających za wysterowanie kierunku obrotu silnika oraz jego włączenie.

Po deklaracji zmiennych i inicjalizacji portów program przechodzi do pętli głównej, w której to kolejno wykonywane są etapy zgodnie z Rysunek 9, czyli odczyt wejść analogowych, zestawienie i wysłanie wiadomości na port szeregowy do komputera PC, ustawienie wyjścia PWM i okresu oczekiwania.

Odczyt wartości połączony został z mapowaniem danych na użyte w projekcie wartości zmiennych, odpowiadające realnym poziomom napięć i natężeń prądów.

Poza pętlą główną programu umieszczony został kod odpowiadający za obsługę zdarzeń przychodzących na port szeregowy. Kod programu wykonywany jest tylko w momencie pojawienia się danych do obsłużenia. Jeśli pojawi się pakiet danych na linii szeregowej jest on odczytywany w całości. Następnie dane są porównywane z wartościami zamieszczonymi w funkcjach warunkowych. Jeżeli warunek zostanie spełniony wykonywana jest odpowiednia część programu. Na końcu bloku warunków znajduje się instrukcja, odpowiadająca za wysterowanie prędkości obrotowej wirnika cyrkulatora.

3.2. Projekt oprogramowania dla sterownika reprogramowalnego PLC

Przed przystąpieniem do pisania oprogramowania na sterownik PLC należało odpowiednio skonfigurować system, ponieważ sterownik SMC firmy LUMEL pracuje z urządzeniami wejść i wyjść w topologii *Master* ↔ *Slave*, zaś komunikacja realizowana jest po magistrali RS-485 i bazuje na protokole Modbus. Użycie tego protokołu wymusza, przed przystąpieniem do pisania kodu, zdefiniowanie urządzeń podrzędnych. Każde z nich powinno mieć unikalny identyfikator (adres), a prędkość transmisji powinna być identyczna dla wszystkich jednostek zastosowanych w systemie. Adresowanie należało wykonać przy użyciu oprogramowania LPCon, poprzez podłączenie się dedykowanym przewodem do płyty czołowej urządzenia podrzędnego.

Następnie w oprogramowaniu CPDev firmy LUMEL utworzony został nowy projekt, do którego dodany został plik z kodem programu, zadeklarowany później w dzienniku zadań, jako cykl powtarzany, co 100 ms. Dodatkowo dodane zostały zaadresowane zmienne globalne, które w przypadku braku bloków komunikacyjnych odpowiadają za komunikację pomiędzy podłączonymi elementami podrzędnymi serii SM, a sterownikiem PLC SMC. Należy pamiętać, aby podczas programowania sterownika SMC określić zadania komunikacyjne działające cyklicznie, które wywołane z określonymi funkcjami (kody funkcji protokołu Modbus) przekazywać będą dane do i z modułów podrzędnych. Rozwiązuje to problem rozgłaszania po sieci podrzędnej rozkazów o modyfikacji rejestrów.

W tym momencie można przystąpić do pisania kodu oprogramowania, który podobnie jak Arduino, możemy rozdzielić na segmenty. Pierwszy z nich to deklaracja zmiennych globalnych i lokalnych. W przypadku zmiennych globalnych, podczas kompilacji przydzielany jest im automatycznie adres w pamięci podręcznej urządzenia. Dodatkowo zadeklarowane zostały lokalne

zmienne rzeczywiste, odpowiedzialne za przechowywanie zamapowanych wartości oraz zmienne *Boolean* wykorzystywane do obsługi przekaźników.

Do sterowania modułem sterownika silników DC Pololu MC33926 zastosowano globalną zmienną całkowitą o nazwie PWM, która przyjmuje wartości od 0 do 255. Ustawienie wartości odbywa się poprzez wysłanie odpowiedniego komunikatu z aplikacji komputerowej, który następnie w Arduino dekodowany jest do formatu rozkazu magistrali Modbus i przekazywany jest do sterownika SMC. Dodatkowo dodano globalną zmienną rzeczywistą PWM_SET, do której przypisywana jest zamapowana wartość PWM, odpowiadająca napięciu zasilania.

Poprawne wyświetlanie wyników w aplikacji komputerowej, w przypadku, gdy realizowana jest operacja sterowania sterownikiem PLC, wymaga cyklicznego wysyłania przez Arduino do modułu sterownika SMC rozkaz o odczytaniu wartości rejestru. Rozkaz ten odpowiada wysterowaniu zmiennej PWM. Następnie odpowiedź sterownika jest dekodowana w mikrokontrolerze, po czym odczytywana jest wartość rejestru PWM i przesyłana do aplikacji.

W podobny sposób realizowane jest sterowanie wyjściami przekaźnikowymi modułu SM4 jak również odczytywanie pozostałych wartości zmiennych. Arduino ma pulę przypisanych na sztywno tablic z wartościami heksadecymalnymi odpowiadającymi konstrukcji właściwego rozkazu sterownika SMC.

3.3. Komputerowa aplikacja sterująca

Na potrzeby realizacji projektu, w celu wizualizacji wykonywanych przez elementy Arduino pomiarów, napisana została w języku C# platformy .Net aplikacja sterująca.

3.3.1. Projekt aplikacji

Proces projektowania aplikacji sterującej pracą zbudowanych modułów elektronicznych rozpoczęto od określenia wymagań, które przedstawia Tabela 2. Następnie z analizy wymagań wyszczególniono przypadki użycia, które zostały przedstawione w Tabeli 3. Na Rysunek 10 przedstawiono wyszczególnione przypadki użycia wraz z relacjami występującymi pomiędzy nimi.

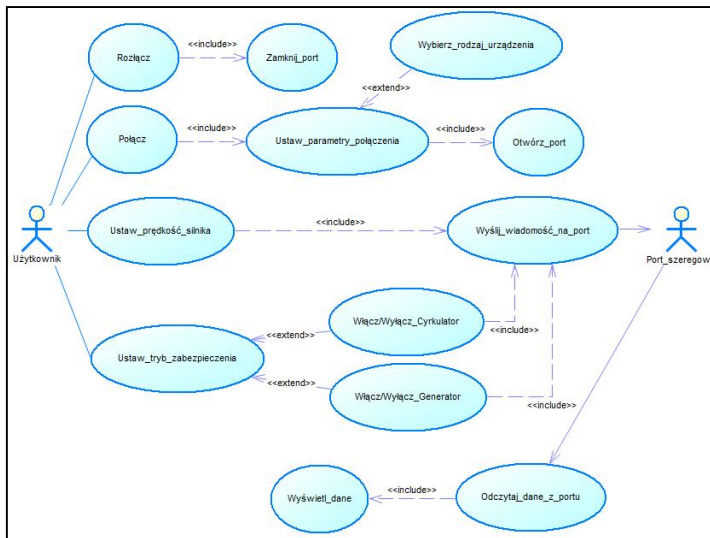
Tabela 2. Wymagania funkcjonalne dla aplikacji

ID Wymagania	Opis
REQ001	Użytkownik ma możliwość połączenia się z systemami sterowania.
REQ002	Użytkownik ma możliwość rozłączenia się z systemami sterowania.
REQ003	Użytkownik ma możliwość wyborów parametrów zestawienia połączenia portu szeregowego.
REQ004	Użytkownik ma możliwość wyboru systemu, z którym chce się połączyć.
REQ005	Użytkownik ma możliwość odczytu z portu szeregowego danych zwrotnych.
REQ006	Użytkownik ma możliwość wyboru trybu pracy zabezpieczeń przekaźnikowych.
REQ007	Użytkownik ma możliwość włączania i wyłączania przekaźników cyrkulatora i generatora.

Tabela 3. Przypadki użycia dla aplikacji sterującej systemami MEW

Typ	ID	Tytuł	Opis
UC	001	Rozłącz	Rozłącza połączenie z Arduino i PLC.
UC	002	Połącz	Zestawia połączenie z określonymi parametrami.
UC	003	Zamknij port	Zamyka port szeregowy.
UC	004	Otwórz port	Otwiera port szeregowy.
UC	005	Ustaw parametry połączenia	Ustawia parametry zestawienia połączenia dla portu szeregowego RS-232.
UC	006	Wybierz rodzaj połączenia	Wybiera urządzenie, którym będziemy zarządzać z poziomu aplikacji, tj. Arduino, czy PLC.
UC	007	Ustaw prędkość silnika	Ustawiaysterowanie silnika cyrkulatora.
UC	008	Wyślij wiadomość na port	Wysyła wiadomość na otwarty port szeregowy, który podłączony jest do Arduino.
UC	009	Odczytaj dane z portu	Odczytuje dane z portu szeregowego, które wystawione zostały przez układ Arduino.
UC	010	Wyświetl dane	Wyświetla otrzymane od Arduino dane w polach

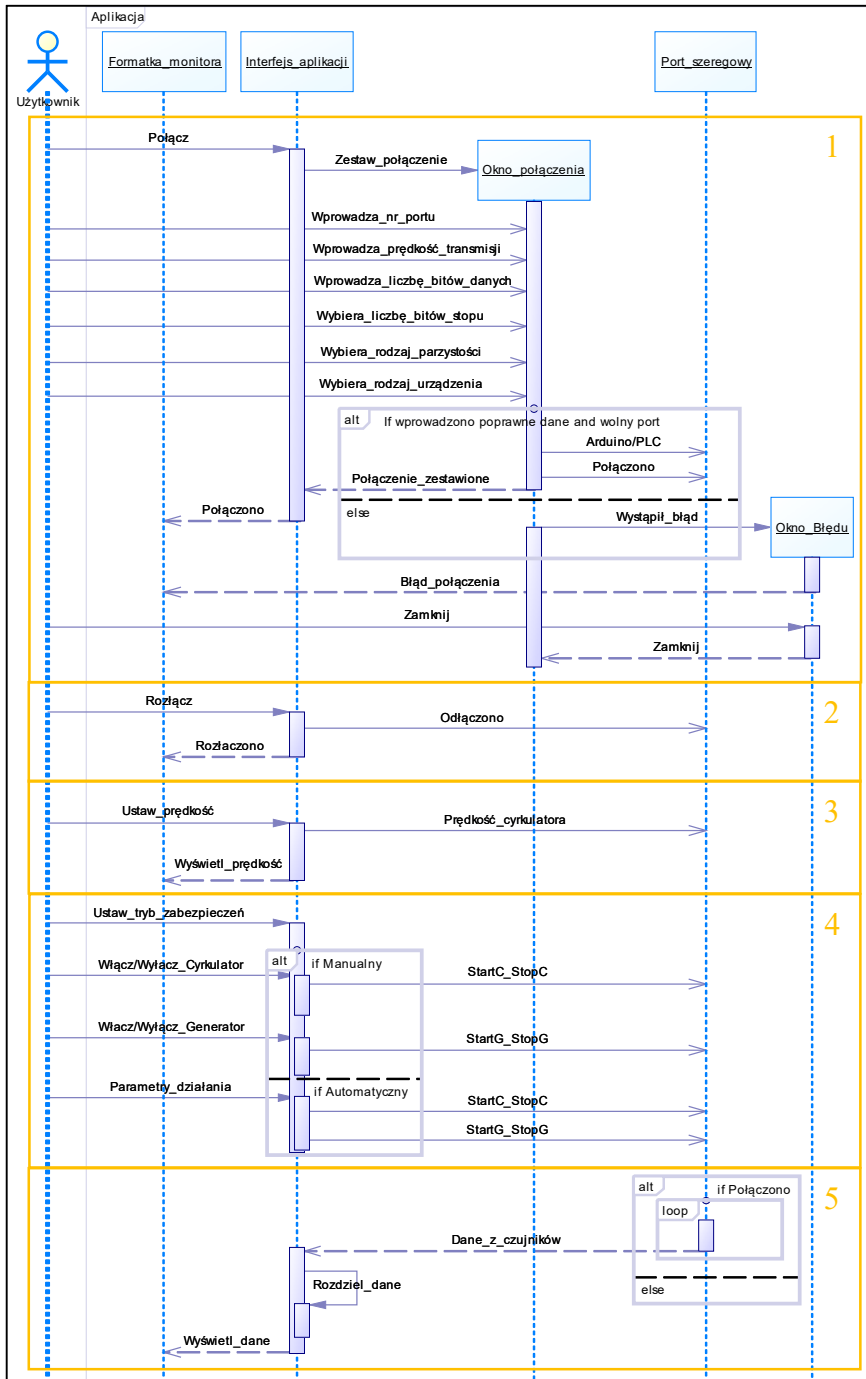
			tekstowych.
UC	011	Ustaw tryb zabezpieczenia	Ustawia tryb zabezpieczenia, w jakim działać ma płytka przekaźników Arduino.
UC	012	Włącz / Wyłącz Cyrkulator	Włącza i wyłącza przekaźniki odpowiedzialne za cyrkulator.
UC	013	Włącz / Wyłącz Generator	Włącza i wyłącza przekaźniki odpowiedzialne za generator (turbinę wiatrową).



Rysunek 10. Diagram przypadków użycia dla aplikacji

Użytkownik, z poziomu aplikacji, ma możliwość ustawienia parametrów połączenia oraz dokonania wyboru urządzenia, z jakim będzie się łączyć aplikacja (Arduino, bądź PLC). Po przejściu fazy zestawienia połączenia użytkownik może ręcznie ustawiać prędkość obrotową wirnika, która przesyłana jest do modułu wykonawczego w postaci wiadomości. Odczyt danych odbywa się poprzez odebranie wiadomości wysłanej przez Arduino. Na Rysunek 11 przedstawiono diagram sekwencji opisujący działanie aplikacji. Kolorowymi prostokątami zaznaczono realizację poszczególnych funkcji, czyli kolejno:

- zestawianie połączenia (pkt 1),
- rozłączanie połączenia (pkt 2),
- ustawienie prędkości obrotowej wirnika (pkt 3),
- ustawienie trybu pracy zabezpieczeń (pkt 4),
- odczyt danych z portu szeregowego (pkt 5).



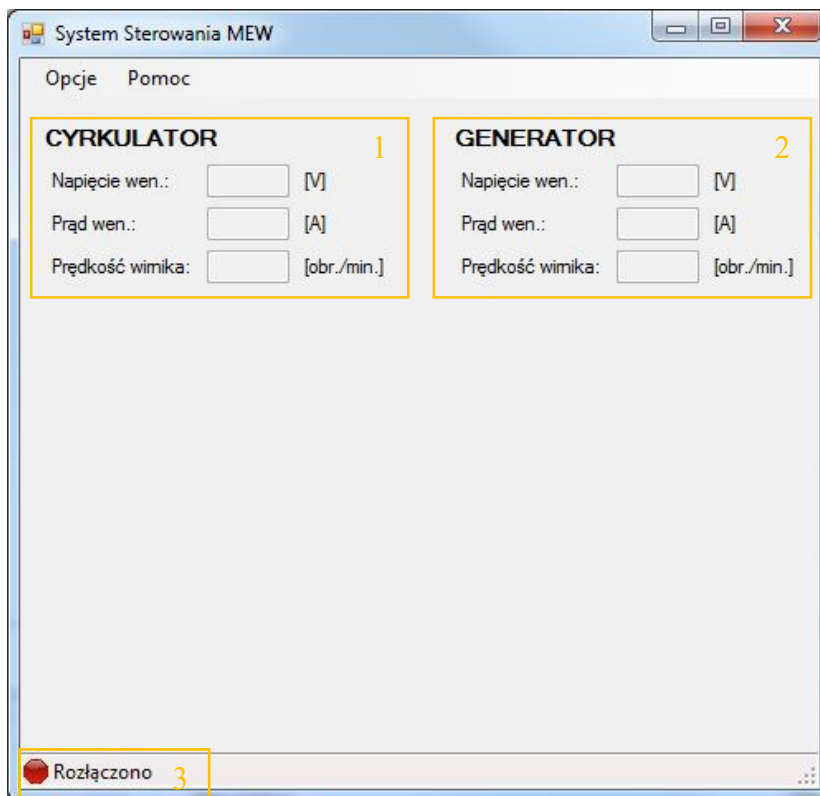
Rysunek 11. Diagram sekwencji dla aplikacji

3.3.2. Implementacja podstawowych funkcji

Okno interfejsu aplikacji przedstawione zostało na Rysunek 12. Podzielono je na dwie główne sekcje: cyrkulator oraz generator.

Sekcja cyrkulatora, wyszczególniona jest na Rysunek 12, jako punkt 1, składa się z trzech nieaktywnych pól tekstowych, w których wyświetlane są wartości odczytane przez Arduino. Kolejno są to: napięcie wentylatora, prąd pobierany przez wentylator oraz prędkość obrotowa wirnika. Sekcję generatora wyszczególniono na Rysunek 12, jako punkt 2, ma on identyczne dane pomiarowe jak cyrkulator

Odczytywanie wartości parametrów jest możliwe dopiero po podłączeniu się do zestawu uruchomieniowego Arduino. Poprawne podłączenie sygnalizowane jest świeceniem kontrolki oznaczonej na Rysunek 12, jako punkt 3.



Rysunek 12. Okno aplikacji po uruchomieniu

W celu podłączenia do elementu wykonawczego należy rozwinąć z menu zakładkę *Opcje*, a następnie z zakładki *Połączenie* wybrać opcję *Połącz*.

Na ekranie wyświetlone zostanie nowe okno zawierające parametry połączenia oraz rodzaj urządzenia, z którego odczytywane będą wartości czujników. Po wprowadzeniu wszystkich parametrów połączenia i wyborze systemu należy wcisnąć przycisk *Połącz*.

Sygnalizacja nawiązania połączenia realizowana jest przez odsłonięcie na głównym oknie aplikacji pola przeznaczonego do sterowania modułem Pololu i przekaźnikami, jak również zmianę koloru koła oraz zmianę napisu na „Połączono” w miejscu wyszczególnionym na Rysunek 12 punkt 3. Jest to zabieg celowy, pozwalający na poinformowanie użytkownika o zainicjalizowaniu połączenia.

W przypadku wpisania niepoprawnych danych, bądź niedostępności portu, wyświetlony zostanie w nowym oknie komunikat informujący użytkownika o wystąpieniu błędu.

Użytkownik może sterować napięciem na cyrkulatorze poprzez przesuwanie suwaka, który wyszczególniono na Rysunek 13 punkt 2. Natomiast w polu tekstowym (oznaczonym na Rysunek 13 punkt 1), znajduje się wyliczona wartość napięcia podawanego na cyrkulator.

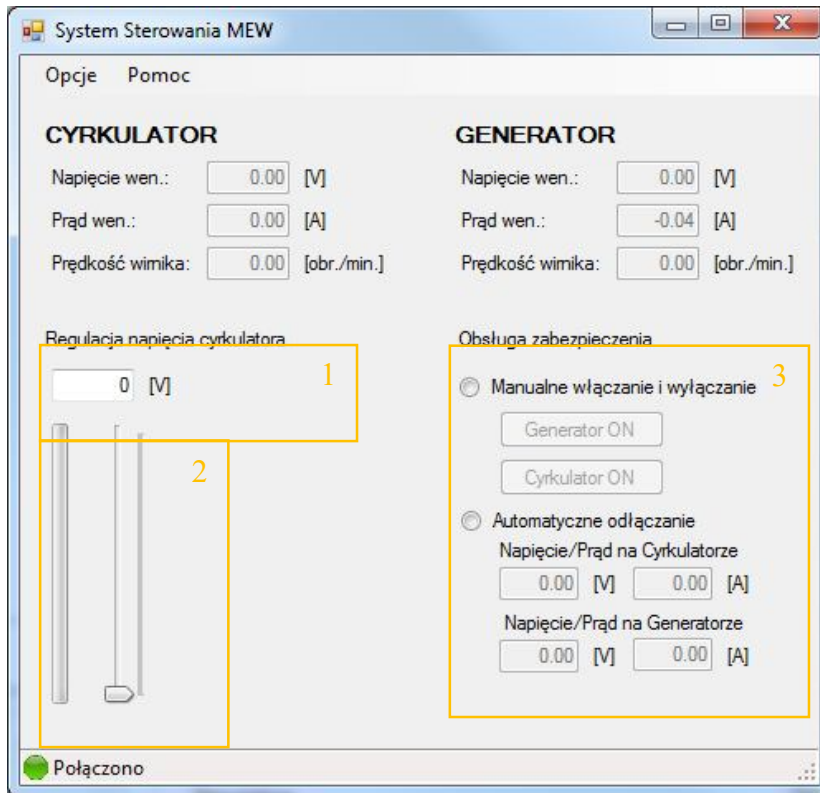
Wartości wysterowania modułu (tj. PWM z zakresu 0..255) przeskalowano, aby pokrywały zakres napięcia zasilania wentylatora, czyli 0..24 V.

Na Rysunek 13 punkt 3 przedstawione zostało rozwiązanie przeznaczone do obsługi włączania/wyłączania przekaźników, czyli obsługi zabezpieczeń, które pracować może w jednym z dwóch trybów: manualnym oraz automatycznym.

W przypadku wybrania trybu manualnego, do dyspozycji użytkownika są dwa przyciski, które odpowiednio włączają/wyłączają napięcie na cyrkulatorze, bądź odłączają/dołączają obciążenie w postaci mostka Graetza z przetwornicą step down.

Wybór trybu automatycznego wymaga określenia górnego progu, którego przekroczenie spowoduje zadziałanie zabezpieczenia i odcięcie źródła zasilania.

Odłączenie od urządzenia może zostać zrealizowane na jeden z dwóch sposobów: poprzez zamknięcie aplikacji, albo wybrania z menu opcji *Połączenia-Rozłącz*. W drugim przypadku okno główne aplikacji wyglądać będzie jak przed połączeniem, co zostało zobrazowane na Rysunek 12.



Rysunek 13. Widok głównego okna aplikacji po połączeniu z zestawem Arduino

4. Testowanie poprawności działania elementów systemu

Niniejszy rozdział zawiera wyniki testów praktycznych przeprowadzonych dla obydwu zaprojektowanych systemów sterowania. Dodatkowo testom poddana została aplikacja, której zadaniem jest monitorowanie pracy systemów oraz sterowanie opisywanymi wcześniej systemami.

4.1. Testy systemu wbudowanego Arduino

Jako pierwszy testom poddany został model elektrowni wraz z podłączonym systemem sterowania opartym o moduł Arduino. Do celów testowania wymienionego systemu napisane zostały przypadki testowe, które zawiera Tabela 4.

Tabela 4. Przypadki testowe dla systemu Arduino

ID	Tytuł	Środowisko	Kroki do wykonania	Oczekiwany rezultat
1	Nawiązanie połączenie z modułem Arduino.	Aplikacja oraz zestaw uruchomieniowy Arduino MEGA 2560.	<ol style="list-style-type: none"> 1. Podłącz Arduino do komputera PC. 2. Włącz aplikację do sterowania systemem MEW. 3. Wybierz z menu opcję połącz. 4. Wprowadź dane do zestawienia połączenia. 5. Wybierz Arduino do połączenia. 6. Naciśnij przycisk „Połącz”. 	<p><i>Poprawny:</i> Nawiązana komunikacja [Połączono].</p> <p><i>Niepoprawny:</i> Okienko błędu połączenia.</p>
2	Rozłączenie komunikacji z zestawem Arduino.	Aplikacja oraz zestaw uruchomieniowy Arduino MEGA 2560.	<ol style="list-style-type: none"> 1. Wybierz z menu opcję rozłączenia. 2. Naciśnij przycisk rozłącz. 	<p><i>Poprawny:</i> Rozłączenie komunikacji [Rozłączono].</p> <p><i>Niepoprawny:</i> Błąd</p>
3	Wysterowanie prędkości silnika.	Aplikacja, zestaw uruchomieniowy Arduino MEGA 2560 oraz sterownik silników DC Pololu MC33926.	<ol style="list-style-type: none"> 1. Nawiąż połączenie z Arduino. 2. Przemieść suwak regulacji napięcia. 	<p><i>Poprawne:</i> Zmiana prędkości obrotowej wirnika i napięcia na cyrkulatorze</p> <p><i>Niepoprawne:</i> Brak odpowiedzi ze strony silnika.</p>
4	Włączanie i wyłączanie przekaźników zabezpieczających.	Aplikacja, zestaw uruchomieniowy Arduino MEGA 2560 oraz płytki przekaźników.	<ol style="list-style-type: none"> 1. Nawiąż połączenie z Arduino. 2. Naciśnij przycisk 	<p><i>Poprawne:</i> Włączenie i wyłączenie przekaźników cyrkulatora lub</p>

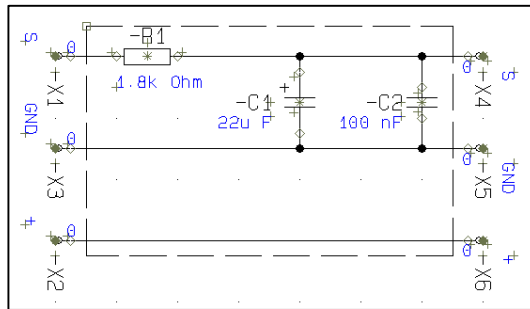
5	Odczyt wartości danych odczytywanych z czujników.	Aplikacja, zestaw uruchomieniowy Arduino MEGA 2560 oraz systemy podrzędne.	włącz/wyłącz cyrkulator lub generator. Nawiąż połączenie z Arduino.	generatora. <i>Niepoprawne:</i> Brak odpowiedzi ze strony płytki przekaźników. <i>Poprawne:</i> Poprawne dane odczytane z czujników powinny zostać wyświetlone w polach tekstowych. <i>Niepoprawne:</i> Brak wyników.
---	---	--	--	---

Testy podłączania i rozłączania połączenia, określone pierwszym i drugim przypadkiem testowym, nie wskazały błędów, a uzyskane wyniki były zgodne z oczekiwanymi rezultatami.

W trakcie przeprowadzania trzeciego testu zauważono, iż moduł sterownika silników DC (Pololu) przestał odpowiadać na sygnały sterujące ze strony aplikacji. Prawdopodobnie mogło to być spowodowane działaniem wbudowanych w układ zabezpieczeń nadprądowych, które aktywowały się podczas gwałtownego (przekraczającego zalecane 3 A) skoku chwilowego poboru prądu. Odłączenie zasilania układu (restart) przywróciło sprawność sterownika Pololu. W celu uniknięcia opisywanego problemu nie należy gwałtownie załączać cyrkulatora na pełną moc lub zastosować wentylator o mniejszej mocy.

Czwarty test również nie wygenerował błędu a uzyskany rezultat był zgodny z oczekiwanym.

Realizacja piątego testu spowodowała wyświetlenie w polach tekstowych błędnych wartości napięć. Pomimo tego, że napięcie podawane na wentylator cyrkulatora wynosiło ok. 12 V, w polu tekstowym wyświetlana była na przemian maksymalna wartość napięcia i wartość zerowa. Rozwiązaniem tego problemu okazało się dołączenie przed wejście analogowe układu całkującego w postaci filtra dolnoprzepustowego o niskiej częstotliwości odcięcia, który przekształcał napięcie impulsowe na wartość przybliżoną do liniowej. Rysunek 14 przedstawia zaproponowany układ.



Rysunek 14. Schemat układu filtra LPF

Układ filtra został w pierwszej kolejności przetestowany na symulacji komputerowej wykonanej za pomocą programu Proteus ISIS, gdzie wartość kondensatora dobrana została tak, aby uzyskać jak najbardziej wygładzoną linię napięcia i w miarę możliwości jak najszybszą odpowiedź układu na zmiany napięcia zasilania. Filtr podłączono przed wejście analogowe Arduino, zaraz po dzielniku napięcia i powtórzono piąty przypadek testowy, który zakończył się poprawnym wyświetleniem danych odczytanych z czujników.

W trakcie realizacji testów dochodziło kilkakrotnie do zresetowania zestawu uruchomieniowego Arduino. Spowodowane to było nieprawidłowym włączeniem napięcia dla poszczególnych modułów - najpierw włączona została część odpowiedzialna za dostarczenie zasilania do Arduino, więc moduł sterowania silników Pololu pobierał napięcie z Arduino, co powodowało przygaśnięcie diody zasilania w Arduino i przerwanie zakończenie pracy. Dlatego system zasilania skonstruowany został w ten sposób, aby włączać go od strony lewej do prawej, zaś wyłączać na odwrót. Dzięki temu napięcie na moduły podawane jest w odpowiedniej kolejności, która nie spowoduje zakłócenia ich pracy.

4.2. Testy systemu ze sterownikiem PLC firmy LUMEL

W drugiej kolejności testom poddany został model elektrowni wraz z podłączonym systemem sterowania opartym o sterownik reprogramowalny PLC SMC firmy LUMEL. Do celów testowania wymienionego systemu, identycznie jak w przypadku Arduino, napisane zostały podobne przypadki testowe, które wypisano w Tabeli 5.

Najwięcej błędów wykazał test nr 1, pomimo, że komunikacja z Arduino została nawiązana poprawnie, jednakże przekazywanie wiadomości do sterownika SMC nie było wykonywane. Problem nawiązywania komunikacji z PLC został jednak rozwiązany, więc test powtórzono i zakończył się on pełnym sukcesem.

Tabela 5. Przypadki testowe dla systemu ze sterownikiem SMC

ID	Tytuł	Środowisko	Kroki do wykonania	Oczekiwany rezultat
1	Nawiązanie połączenie ze sterownikiem SMC.	Aplikacja, zestaw uruchomieniowy Arduino MEGA 2560, sterownik SMC.	<ol style="list-style-type: none"> 1. Podłącz Arduino do komputera PC. 2. Podłącz SMC do wyjścia Serial1 układu Arduino. 3. Włącz aplikację do sterowania systemem MEW. 4. Wybierz z menu opcję połącz. 5. Wprowadź dane do zestawienia połączenia. 6. Wybierz w opcji sterownik PLC do połączenia. 7. Naciśnij przycisk „Połącz”. 	<p><i>Poprawny:</i> Nawiązana komunikacja [Połączono] oraz miganie diody na obudowie sterownika SMC.</p> <p><i>Niepoprawny:</i> Okienko błędu połączenia.</p>
2	Rozłączenie komunikacji z sterownikiem SMC.	Aplikacja, zestaw uruchomieniowy Arduino MEGA 2560, sterownik SMC.	<ol style="list-style-type: none"> 1. Wybierz z menu opcję rozłączenia. 2. Naciśnij rozłącz. 	<p><i>Poprawny:</i> Rozłączenie komunikacji [Rozłączono] oraz zaprzestanie migania diody na obudowie sterownika SMC.</p> <p><i>Niepoprawny:</i> Błąd</p>
3	Wysterowanie prędkości silnika.	Aplikacja, zestaw uruchomieniowy Arduino MEGA 2560, sterownik silników DC Pololu MC33926, sterownik SMC.	<ol style="list-style-type: none"> 1. Nawiąż połączenie z PLC. 2. Przemieść suwak regulacji napięcia. 	<p><i>Poprawne:</i> Zmiana prędkości obrotowej wirnika i napięcia na cyrkulatorze</p> <p><i>Niepoprawne:</i> Brak odpowiedzi ze strony silnika.</p>

4	Włączanie i wyłączanie przekaźników zabezpieczających.	Aplikacja, zestaw uruchomieniowy Arduino MEGA 2560, sterownik silników DC Pololu MC33926, sterownik SMC, moduł wyjść binarnych SM4.	1. Nawiąż połączenie z PLC. 2. Naciśnij przycisk włącz/wyłącz cyrkulator lub generator.	<i>Poprawne:</i> Włączenie wyłączenie przekaźników cyrkulatora lub generatora. <i>Niepoprawne:</i> Brak odpowiedzi ze strony płytki przekaźników.
5	Odczyt wartości danych odczytywanych z czujników.	Aplikacja, zestaw uruchomieniowy Arduino MEGA 2560, sterownik silników DC Pololu MC33926, sterownik SMC, moduł wejść analogowych SM2.	Nawiąż połączenie z PLC.	<i>Poprawne:</i> Poprawne dane odczytane z czujników powinny zostać wyświetlone w polach tekstowych. <i>Niepoprawne:</i> Brak wyników.

Test opisany drugim przypadkiem testowym, związanym z rozłączaniem, przeszedł poprawnie. Komunikację z systemem PLC i mostem Arduino można załączać i rozłączać, jednakże w przypadku rozłączenia się z PLC, połączenia z Arduino, po czym ponownego połączenia z PLC występuje brak komunikacji spowodowany nieznanym zjawiskiem. Wyjęcie i włożenie wtyczki RS-232 rozwiązuje problem nawiązania komunikacji.

Realizacja testu trzeciego i czwartego nie wykazała żadnych błędów. Możliwe było załączanie i rozłączanie przekaźników na płycie systemu Arduino, a także sterowanie modułem Pololu, tj. zmiana prędkości obrotowej wentylatora. Aczkolwiek odpowiedź systemu sterowania na zmianę parametru prędkości, wykonaną w aplikacji nie była tak płynna i szybka jak w przypadku systemu opartego na Arduino. Związane jest to z dodatkowym opóźnieniem pętli głównej programu, które wprowadzono w celu nawiązania poprawnej komunikacji z SMC.

5. Podsumowanie

W artykule zaprezentowano realizację etapu projektowania, budowy oraz testowania „Systemu sterowania modelem elektrowni wiatrowej”.

Projekt modelu został wykonany przy pomocy wizualizacji 3D zrealizowanej za pomocą programu AutoCAD 2013, co ułatwiło wypracowanie koncepcji mechanicznej modelu. Niestety pierwsza wersja opracowanego rozwiązania zakładała użycie kosztownej i skomplikowanej obudowy z pleksiglasu, przez co projekt nie doszedł do skutku. Drugą wersję makiety również zamodelowano w AutoCAD 2013, a jej realizacja zakończyła się sukcesem.

System sterowania, dla wykonanej makiety, został zrealizowany na dwóch platformach dublujących swoją funkcjonalność, tj. układzie wbudowanym Arduino oraz sterowniku reprogramowalnym SMC firmy LUMEL. Do sterowania silnikiem DC wykorzystano gotowy układ firmy Pololu, który jak się okazało w trakcie pomiarów oscyloskopem, ma wyjście impulsowe. Zastosowanie tego rodzaju wyjścia może wpływać negatywnie na falownik umieszczony wewnątrz obudowy wentylatora osiowego, który przekształca napięcie stałe na impulsy dla cewek uzwojenia. Dlatego należy rozważyć możliwość użycia, zamiast sterownika Pololu układu LM317 w połączeniu z przedwzmacniaczem operacyjnym, który może być sterowany z wyjścia PWM systemów wbudowanych. Układ LM317 gwarantuje liniową charakterystykę działania wyjścia napięciowego.

Dodatkowo w trakcie budowy modelu wystąpił problem z realizacją pomiaru napięcia podawanego na silnik cyrkulatora. Problem wynikał z impulsowego trybu pracy wyjścia sterownika Pololu MC 33926. Problem rozwiązano dodając filtr dolnoprzepustowy na wejście układu pomiarowego, tj. po dzielniku napięcia przed wejściem analogowym. Zastosowanie filtra spowodowało niewielki spadek napięcia, który skorygowano w oprogramowaniu systemów sterowania wykonując korektę w mapowaniu wartości mierzonych parametrów napięcia.

W celu realizacji monitorowania parametrów zastosowanych czujników i sterowania układami, napisana została aplikacja. W trakcie uruchamiania aplikacji najwięcej problemów sprawiło nawiązanie komunikacji z modułem Arduino oraz odpowiednie dobranie czasów opóźnienia pętli głównej mikrokontrolera. Dobór czasów opóźnienia jest istotny z punktu widzenia zapewnienia odpowiednio szybkiej reakcji na zmianę wartości parametru wysterowania silnika DC wykonaną przez użytkownika.

Projekt „Systemu sterowania modelem elektrowni wiatrowej” może być w przyszłości rozbudowany o kolejne elementy, jak np. moduł do komunikacji bezprzewodowej zgodny ze standardem XBee. W przypadku PLC dołączane mogą być kolejne moduły zwiększające ilość wejść i wyjść, bądź dodatkowo dublujące funkcjonalność np. w postaci mierników panelowych, które komunikować się mogą z modułem SMC po magistrali szeregowej.

Literatura

- [1] NALEPA K., MIĄSKOWSKI W., PIETKIEWICZ P., PIECHOCKI J., BOGACZ P., *Poradnik małej Energetyki Wiatrowej*, Olsztyn, 2011.
- [2] PELCZAR P., *Elektrownie wiatrowe w systemie energetycznym. Pomiary, zjawiska, ocena.*, Wrocław, 2008.
- [3] Dokumentacja techniczna urządzeń firmy LUMEL.
- [4] ANDERSON R., CERVO D., *Arduino dla zaawansowanych*, HELION, Gliwice, 2014.
- [5] POJMAŃSKI G., *Opinia dotycząca zagrożeń związanych z eksploatacją i awariami turbin wiatrowych*, Uniwersytet Warszawski Wydział Fizyki, Warszawa.
- [6] UTKIN M., *Turbiny wiatrowe*, Młody Technik, Luty 2005, str. 26-29.
- [7] MAYO J., *C# 3.0 dla .NET 3.5. Księga Eksperta*, HELION, Gliwice, 2010.

The control system of a wind power plant model

ABSTRACT: The paper describes a design and implementation of the control system for the model wind power plant. The functional and nonfunctional requirements imposed on the designed system are presented. Moreover interaction diagrams for selected use cases in UML notation are presented. The paper also discusses some of implementing solutions used in the proposed system. The whole is to constitute the laboratory stand used for laboratory classes conducted in the Institute of Teleinformatics and Automation Department of Cybernetics MUT.

KEYWORDS: wind turbines, Small Wind Power Plants, Arduino, PLCs controllers

Praca wpłynęła do redakcji: 10.12.2015 r.