

# **AKADEMIA HUMANISTYCZNO – EKONOMICZNA W ŁODZI**

**Kierunek Informatyka**

**Praca inżynierska**

**Przemysław Marcin Zapart**  
Nr albumu: 144071

## **Projekt sterownika pompy wodnej.**

Przyjmuję pracę jako inżynierską:

podpis promotora .....

data .....



Odstępy, rozmiary, rodzaje czcionek  
muszą być zachowane, zgodnie z tym  
wzorem.

**Promotor:**  
**dr inż. Wanda Gryglewicz -**  
**Kacerka**

**ŁÓDŹ 2023**

## 1. Wstęp



*Rysunek 1. Pompa wodna w instalacji tryskaczowej.*

Systemy przeciwpożarowe we współczesnym świecie odgrywają nieocenioną rolę w celu ochrony życia ludzkiego oraz mienia. Są to złożone systemy mające na celu wykrywanie zagrożenia, alarmowanie, przekazywanie informacji do odpowiednich służb o tym zagrożeniu, uruchomieniu odpowiednich środków zapobiegawczych, a także wspomagające ewakuację z miejsca zagrożenia. Każdy z tych czynników odgrywa niebagatelną rolę w tym systemie. System taki potrafi być bardzo złożony i odpowiednio kosztowny w zależności od ochranianego obiektu. Często względy ubezpieczeniowe wymagają tych bardzo kosztownych rozwiązań. Systemy te są określane przez prawo w danym kraju, dlatego powinny spełniać określone normy. W całym tym systemie jedną z głównych ról odgrywają instalacje przeciwpożarowe, które w zależności od przeznaczenia mogą być bardzo zróżnicowane. Instalacje te można podzielić na suche oraz mokre. Jedną z najpowszechniej spotykaną instalacją jest instalacja tryskaczowa, która pozwala osiągnąć optymalny poziom bezpieczeństwa.

## **2. Zakres pracy**

Celem projektu jest opracowanie kontrolera do przeciwpożarowej pompy wodnej w instalacji tryskaczowej działającego w oparciu o platformę Andino. Kontroler ma być wykorzystywany do testowania oraz rozwijania nowych funkcjonalności urządzenia. Projekt zakłada modułową budowę podstawowych komponentów tj. zasilanie, karta SD, multiplexer oraz procesor. Projekt zakłada również napisanie oprogramowania służącego jako interfejs dla użytkownika systemu oraz układu testowego pozwalającego na testowanie projektu. Układ testowy będzie również oparty o platformę Andino.

## **3. Charakterystyka wybranych narzędzi do wykonania pracy**

### **3.1. Visual Studio**

Jest to zintegrowane środowisko programistyczne (IDE) firmy Microsoft. Służy do tworzenia programów komputerowych, w tym stron internetowych, aplikacji internetowych, usług internetowych i aplikacji mobilnych. Visual Studio korzysta z platform programistycznych firmy Microsoft, takich jak Windows API, Windows Forms, Windows Presentation Foundation, Windows Store i Microsoft Silverlight. Może generować zarówno kod natywny, jak i kod zarządzany. Visual Studio obsługuje 36 różnych języków programowania i pozwala edytorowi kodu i debuggerowi obsługiwać (w różnym stopniu) prawie każdy język programowania, pod warunkiem, że istnieje usługa specyficzna dla języka. Wbudowane języki to C/C++, C++/CLI, Visual Basic .NET, C#, F#, JavaScript, TypeScript, XML, XSLT, HTML i CSS. Obsługa innych języków, takich jak między innymi Python, Ruby, Node.js i M, jest dostępna za pośrednictwem wtyczek.

### **3.2. Easy EDA**

Jest internetowy pakiet narzędzi EDA, który umożliwia inżynierom projektowanie, symulowanie, udostępnianie (publicznie lub prywatnie) oraz omawianie schematów, symulacji i płytek drukowanych. Inne funkcje obejmują tworzenie zestawień materiałów, plików Gerber oraz plików typu *pick and place*, a także dokumentów wyjściowych w formatach PDF, PNG i SVG. Program umożliwia tworzenie i edycję schematów, symulację SPICE mieszanych obwodów analogowych i cyfrowych, tworzenie i edycję układów obwodów drukowanych oraz opcjonalnie produkcję płytek drukowanych.

### 3.3. Diagrams.net

To darmowe i wieloplatformowe oprogramowanie do rysowania wykresów typu *open source*, opracowane w HTML5 i JavaScript. Jego interfejs może być używany do tworzenia diagramów, takich jak schematy blokowe, szkielety, diagramy UML, schematy organizacyjne i diagramy sieciowe. Diagrams.net jest dostępny online jako aplikacja internetowa obsługująca wiele przeglądarek oraz jako aplikacja offline na komputery stacjonarne dla systemów Linux, macOS i Windows. Jego aplikacja offline jest zbudowana przy użyciu frameworka Electron. Aplikacja internetowa nie wymaga logowania ani rejestracji online i może otwierać się i zapisywać na lokalnym dysku twardym. Obsługiwane formaty przechowywania i eksportu do pobrania to PNG, JPEG, SVG i PDF. Integruje się również z usługami przechowywania w chmurze, takimi jak Dropbox, OneDrive, Google Drive, GitHub i GitLab.com. Jest również dostępny jako wtyczka do osadzania aplikacji internetowej na platformach takich jak NextCloud, MediaWiki, Notion, Atlassian Confluence i Jira. Jest to alternatywa dla takich programów jak: Lucidchart, Microsoft Visio i SmartDraw.

### 3.4. Platforma Arduino

Jest to platforma programistyczna dla systemów wbudowanych oparta na prostym projekcie Open Hardware przeznaczonym dla mikrokontrolerów montowanych w pojedynczym obwodzie drukowanym, z wbudowaną obsługą układów wejścia/wyjścia oraz standaryzowanym językiem programowania. Język programowania Arduino jest oparty na środowisku Wiring i zasadniczo na języku C/C++ (kilka prostych przekształceń kodu wykonywane jest przed przejściem do avr-gcc). Celem projektu Arduino jest przygotowanie narzędzi – ogólnodostępnych, tanich, niewymagających dużych nakładów finansowych, elastycznych i łatwych w użyciu przez hobbystów. Częściowo Arduino stanowi również alternatywę dla osób, które nie mają dostępu do bardziej zaawansowanych kontrolerów, wymagających bardziej skomplikowanych narzędzi. Arduino może być wykorzystany do tworzenia samodzielnych interaktywnych obiektów lub może być podłączony do komputera-hosta. Typowa płyta Arduino zawiera kontroler, cyfrowe i analogowe linie wejścia/wyjścia oraz interfejs UART lub USB dla połączeń z komputerem-hostem. Komputer jest wykorzystywany do programowania kontrolera oraz do interakcji w czasie działania z Arduino. Możliwe jest programowanie interfejsu w kilkunastu językach programowania, m.in. w Javie, ActionScript, C/C++, C#, Perl, VBScript. Programowanie odbywa się najczęściej za pośrednictwem Arduino IDE bazującym na projekcie Processing. Wewnątrz Arduino IDE kod programu jest pośrednio kompilowany przez avr-gcc, a następnie wgrywany do podłączonej płyty Arduino. IDE działa

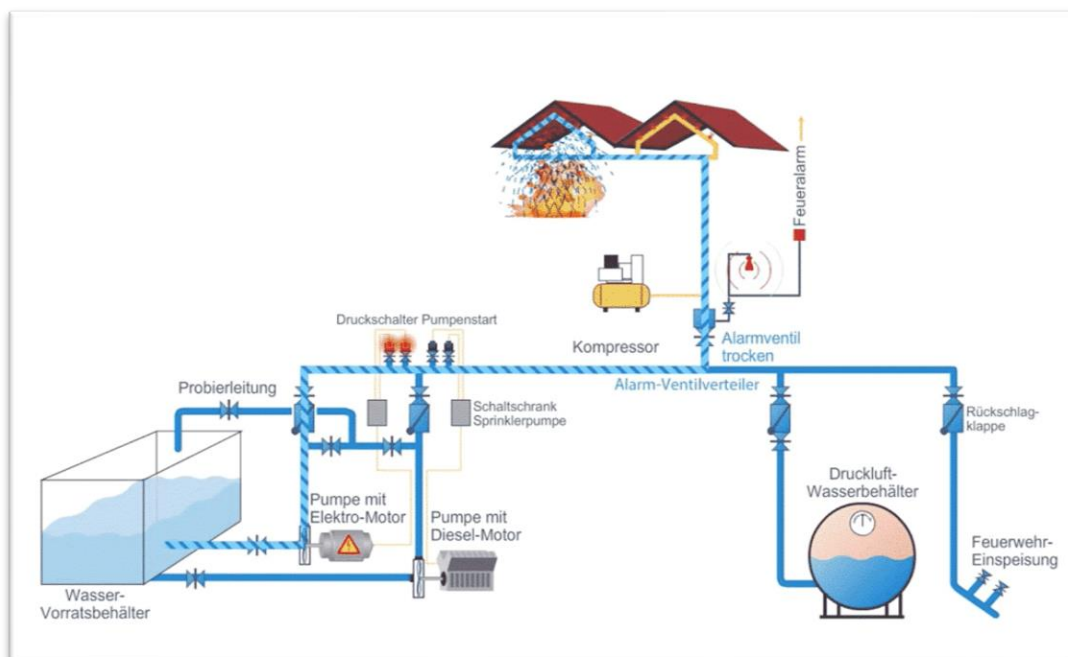
wtedy jako emulator terminala szeregowego, pozwalając na interakcję z pracującym Arduino. Obecnie układy Arduino bazują na mikrokontrolerach Atmel AVR. Nie jest to formalnym wymogiem i może być rozszerzony, o ile sprzęt i narzędzia będą wspierały język Arduino i zostaną zaakceptowane przez Projekt Arduino. Na rynku istnieje wiele klonów inspirowanych oryginalnym Arduino, bądź kompatybilne z jego rozszerzeniami, takimi jak karty rozszerzeń, jednak rozprowadzanych pod innymi nazwami. Obecnie sprzedawane są wstępnie zmontowane układy Arduino, jednak cały czas jest dostępny schemat sprzętowy dla tych, którzy chcą zbudować Arduino samodzielnie.

### **3.5. Visual micro**

Jest to rozszerzenie (wtyczka/dodatek) dla Microsoft Visual Studio 2017, 2019 i 2022 oraz dla Atmel Studio 7, które umożliwia opracowywanie, kompilowanie i przesyłanie dowolnego projektu Arduino na dowolną płytę Arduino, jednocześnie korzystając z potężne funkcje Visual Studio i Atmel Studio. Visual Micro współpracuje i jest kompatybilny ze środowiskiem programistycznym Arduino, korzystając z tych samych bibliotek, kodu źródłowego i narzędzi programistycznych. Różnica polega na interfejsie użytkownika Visual Micro, który zapewnia zaawansowane i profesjonalne środowisko programistyczne oraz pozwala na bardziej zaawansowane programowanie niż istniejący Arduino IDE.

## 4. Projekt systemu

### 4.1. Zasada działania



Rysunek 2. Typowa instalacja tryskaczowa.

Wodna instalacja tryskaczowa w najbardziej podstawowym modelu jest oparta na pompie wodnej napędzanej silnikiem spalinowym lub elektrycznym, skrzynki rozdzielczej sterującej pracą silnika, zbiornikiem wody oraz instalacją rurową doprowadzającą wodę do miejsc docelowych. Każdy koniec systemu zakończony jest tryskaczem, który ma postać szklanej ampułki wypełnionej cieczą. Ciecz ta pod wpływem temperatury zwiększa swoją objętość do momentu rozerwania szklanej ampułki. Woda w instalacji zostaje uwolniona a ciśnienie w instalacji rurowej spada, wskutek czego zostaje uruchomiony silnik pompy wodnej. Pompa zapewnia odpowiednią ilość wody do gaszenia lub ograniczenia pożaru. Samo gaszenie pożaru następuje w sekcji zagrożonej pożarem, w związku z czym szkody materiałowe mogą zostać znacznie ograniczone. Silnik może zostać uruchomiony również ręcznie w zależności od wytycznych projektowych.



Rysunek 3. Tryskacz ze szklaną ampułką.

## 4.2. Skrzynka rozdzielcza



Rysunek 4. Skrzynka kontrolna WBFP-D4 w instalacji tryskaczowej z wykorzystaniem silnika spalinowego.

Skrzynka rozdzielcza ma za zadanie sterowaniem silnikiem. Wszelkie czujniki, za pomocą dedykowanego okablowania elektrycznego, są podłączone do skrzynki, a także zasilanie, ładowarki do baterii oraz cały interfejs użytkownika służący do wydawania poleceń i wyświetlaniu stanu instalacji. Kontroler może znajdować się w jednym z trzech stanów:

- OUT - nie można uruchomić instalacji, ale zbierane są podstawowe informacje z czujników.
- HAND - możliwy ręczny rozruch urządzenia za pomocą przycisku *start*.
- AUTO – urządzenie znajduje się w trybie auto. Nie jest możliwy ręczny rozruch urządzenia.

## 4.3. Podstawowe założenia projektu

Założeniem projektu jest zaprojektowanie i oprogramowanie kontrolera do skrzynki rozdzielczej sterującą pracą silnika spalinowego instalacji tryskaczowej oraz panelu sterowania tego kontrolera symulującego działanie skrzynki rozdzielczej. Projekt ma być eksperymentalną platformą umożliwiającą wdrażanie nowych rozwiązań lub do celów edukacyjnych. Każde takie urządzenie w realnym świecie, musi spełniać różne określone normy w różnych krajach, dlatego projekt nie będzie podlegał pod żadne, z góry określone normy, aby był bardziej elastyczny. Także strona sprzętowa będzie dostosowana do pracy z napięciem +5V. Projekt powinien cechować się dużą elastycznością oraz łatwą możliwością dalszego rozwoju. Cała logika systemu znajduje się po stronie mikrokontrolera. Wszystkie ustawienia takie jak nazwy, ustawienia czasu, numeru seryjnego czy ilości zębów na kole zamachowym znajdują się w pamięci *eeeprom* mikrokontrolera. Z poziomu panelu sterowania można zmienić niektóre

ustawienia. Powinna istnieć również możliwość przywrócenia ustawień fabrycznych. Program dla mikrokontrolera zostanie napisany w języku C++ natomiast oprogramowanie panelu sterowania w języku C#. System składa się z 16 wejść oraz 16 wyjść cyfrowych oraz 16 wejść analogowych służących do pomiaru napięcia na poszczególnych sensorach analogowych. Połączenie z panelem sterowania odbywa się za pomocą szeregowej transmisji danych RS 232. Jest to standard przesyłania danych pomiędzy urządzeniami elektronicznymi.

#### **4.4. Założenia projektu**

##### **4.4.1. Podstawowe założenia kontrolera**

###### **4.4.1.1. Założenia sprzętowe**

Kontroler ma mieć charakter modułową. Napięcie zasilana kontrolera 5 – 12 VCD. Regulator napięcia pozwala uzyskać napięcie 5 volt oraz 3,3 volt. Wbudowana dioda *Led* ma informować o poprawnym podłączonym napięciu. Komunikacja z panelem sterowania musi odbywać się za pomocą transmisji szeregowej poprzez układ scalony *max232*. Kontroler musi posiadać wejścia cyfrowe do sprawdzania stanów cyfrowych czujników oraz wejścia analogowe do pomiaru napięcia. Układ RTC DS1307 zapewni obsługę czasu. Dodatkowo na kontrolerze powinno znajdować się miejsce do obsługi modułu karty SD, aby zapisywać wszelkie zdarzenia.

###### **4.4.1.2. Założenia programowe**

Program odbiera dane z panelu sterowania a następnie je właściwie interpretuje. Kontroler może otrzymywać trzy rodzaje danych: dane sterujące pracą kontrolera takie jak start, stop czy reset, dane zmieniające ustawienia kontrolera takie jak zmiana nazwy czujnika oraz dane żądające informacji na temat kontrolera takie jak numer seryjny czy wersja oprogramowania. Wszelkie nazwy są przechowywane w pamięci eeprom i zostają wczytane w chwili uruchomienia procesora. Istnieje możliwość zmiany tych ustawień za pomocą panelu sterowania. Wszystkie alarmy zostaną zapisane na karcie SD w pliku *log.csv*. Program musi także obsługiwać funkcje czasowe przy pomocy układu DS1307. **Założenia funkcyjne:**

- Start automatyczny – sekwencja startowa polega na sześciokrotnej próbie rozruchu silnika do uzyskania przynajmniej 600 obr/min. Każda próba trwa 15 sekund. Pomiędzy każdą próbą następuje 15 sekundowa przerwa standardowo każde urządzenie posiada 2 baterie rozruchowe. Podczas rozruchu baterie są używane



naprzemiennie, pod warunkiem, że dana bateria jest odpowiednio naładowana. Jeżeli obie baterie są rozładowane start jest niemożliwy a system zgłasza błąd.

- Start ręczny – w każdej próbie startowej jest wykorzystywana inna bateria pod warunkiem, że jest ona naładowana. Jeżeli obie są rozładowane start jest niemożliwy a system zgłasza błąd.
- Reset – po wciśnięciu przycisku reset wszelkie błędy zgłaszane przez czujniki cyfrowe zostają skasowane a następnie po 5 sekundach zostanie sprawdzony ich stan ponownie. Raz włączony alarm nie może się samoistnie wyłączyć. Należy go zresetować.
- Start automatyczny jest możliwy tylko w przypadku ustawienia przełącznika HOA (HAND, OUT, AUTO) w pozycji **A** oraz przy spadku ciśnienia w instalacji.
- Start ręczny jest tylko możliwy w przypadku, kiedy przełącznik HOA jest w pozycji **H**, stan ciśnienia w instalacji nie ma wpływu.
- Stop polega na odcięciu paliwa za pomocą elektromagnesu. W przypadku tryb automatycznego ciśnienie w instalacji musi być przywrócone do odpowiedniego poziomu inaczej nie można zatrzymać maszyny. Po spadku obrotów silnika do 0 obrotów na minutę, magnes pozostaje włączony jeszcze przez 5 sekund, aby nie nastąpił restart maszyny.
- W czasie gotowości do pracy, system zbiera i wyświetla informacje z sensorów analogowych oraz cyfrowych takich jak: ciśnienie wody, niska temperatura silnika, poziom paliwa, przeciek zbiornika paliwa, zawór paliwa zamknięty.
- W czasie pracy silnika zbierane są dodatkowe informacje takie jak: ciśnienie oleju, wysoka temperatura silnika, przepływ wody przez instalację chłodzącą silnika.
- Włączenie czujnika *Niska temperatura* powoduje uruchomienie ogrzewania silnika jednak nie powoduje włączenia alarmu. Po osiągnięciu właściwej temperatury po 5 sekundach ogrzewanie zostaje wyłączone. Ma to zapewnić, że silnik jest zawsze gotowy do optymalnej pracy.
- Wszelkie ustawienia dotyczące urządzenia zapisane są w pamięci eeprom.

- Każde zdarzenie zapisane jest na kartę SD.
- Możliwość edycji ustawień.
- Odczyt czasu.
- Obliczanie obrotów silnika.

#### **4.4.2. Podstawowe założenia panelu sterowania**

Podstawowym zadaniem panelu sterowania jest komunikacja pomiędzy użytkownikiem a kontrolerem. Panel sterowania musi mieć zaimplementowaną możliwość komunikacji szeregową w celu wymiany informacji z kontrolerem. Panel powinien mieć zdefiniowane podstawowe przyciski, bo sterowaniem kontrolera takie jak: *start*, *stop*, *reset*, *hand*, *out*, *auto*. Służą one do wydawania podstawowych poleceń do sterowania kontrolerem. Istotnym elementem powinno być możliwość wyświetlania stanu kontrolera oraz podłączonych do niego czujników analogowych oraz cyfrowych. Użytkownik powinien mieć także możliwość własnej edycji wybranych ustawień czujników bądź samego kontrolera. Przed zmianą ustawień należy uzyskać prawa na modyfikację ustawień poprzez zalogowanie się do systemu. Część zdefiniowanych ustawień podstawowych nie jest możliwa do edycji.

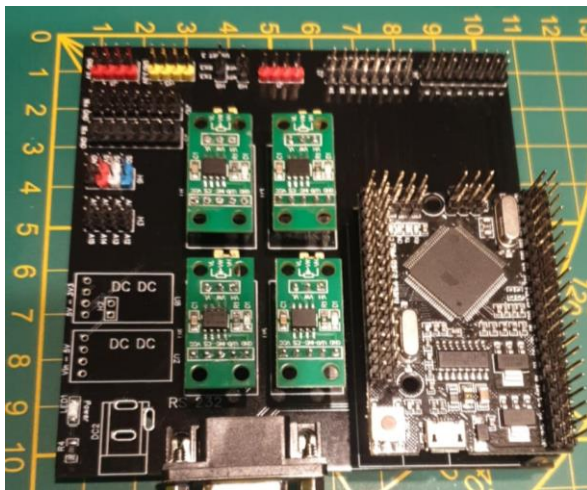
#### **Założenia funkcyjne:**

- Panel posiada trzy przyciski do sterowania kontrolerem: START, STOP, RESET oraz trzy przyciski ustawiające tryb pracy kontrolera: HAND OUT, AUTO (przełącznik HOA) . Każde ustawienie przycisków HOA ma wpływ na przyciski sterowania. W trybie HAND możliwa jest ręczna obsługa maszyny. W trybie AUTO nie jest możliwe ręczne wystartowanie maszyny. Zawsze można zresetować alarm czujników cyfrowych a poza trybem OUT jest możliwe zatrzymanie maszyny.
- Wyświetlanie aktualnego stanu czujników cyfrowych, analogowych oraz stanu logicznego kontrolera.
- Wyświetlany czas otrzymany z kontrolera
- Wyświetlana aktualna prędkość obrotów silnika
- Możliwość edycji nazwy wybranych czujników cyfrowych

- Możliwość edycji ustawień wybranych czujników analogowych
- Możliwość edycji nazwy wybranych przełączników
- Możliwość zmiany numeru seryjnego kontrolera
- Wyświetlanie wszystkich ustawień wszystkich czujników
- Zmiana praw dostępu poprzez logowanie

## 4.5. Testowanie

### 4.5.1. Płytki testowa



Rysunek 5. Płytki testowa

Do testów projektowych została również zaprojektowana i wdrożona platforma oparta o płytkę Arduino Mega PRO, wraz z dedykowanym do niej panelem sterowania napisanym w języku C#. Płytki posiada 16 cyfrowych pinów wejścia oraz 16 cyfrowych pinów wyjścia. Na pokładzie znajdują się również cztery cyfrowe potencjometry X9C103 służące do symulowania czujników analogowych. Potencjometry pracują z napięciem 5V, ale istnieje możliwość podłączenia dla każdego potencjometru innego, osobnego napięcia. Wyprowadzone zostało również 8 analogowych wyjść Arduino, dwa wyjścia dla magistrali I2C oraz dwa wyjścia dla transmisji szeregowej. Usart 2 podłączony jest do standardowego wyjścia SUB-D 9 – pinowego poprzez układ scalony max232. Wyprowadzone zostały dwa piny mikrokontrolera obsługujących przerwania. Na płytce znajduje się również po 5 połączeń 5V, GND oraz 3.3V, GND oraz 32 piny podłączone do masy. Napięcie zasilania 5V – 12V poprzez dwa konwertery napięć dostarczają napięcie 5V oraz 3.3V. Płytki jest podłączona z komputerem za pomocą

kabla USB poprzez port Arduino. Program wysyła co 200ms dane na temat ustawień wszystkich pinów cyfrowych oraz odbiera dane z panelu sterowania dotyczących ustawień pinów cyfrowych, cyfrowych potencjometrów oraz prędkości obrotowej silnika. Prędkość silnika symulowana jest za pomocą naprzemiennego ustawiania w stan wysoki i niski pinu *INT* 1.

#### 4.5.2. Dedykowane oprogramowanie płytki testowej

The screenshot shows a software application window titled "Form1". It contains several sections for monitoring and controlling a system:

- Serial settings:** Includes dropdowns for "Select COM Port" (set to COM4) and "Select Baud Rate" (set to 19200). There are "Disconnect", "Connect", and "Refresh" buttons. The "Status" indicator shows "Connected" in green.
- Digital states:**
  - Digital out:** A list of 16 digital output pins with their corresponding labels:
 

0	System pressure	8	Charger 1 fail
1	Oil pressure	9	Charger 2 fail
2	Engint temp low	10	Starter 1 fail
3	Engine temp high	11	Starter 2 fail
4	Flow water	12	User 1
5	Fuel level low	13	User 2
6	Fueal valve closed	14	User 3
7	Fuel tank leak	15	user1
  - Digital in:** A list of 16 digital input pins with their corresponding labels:
 

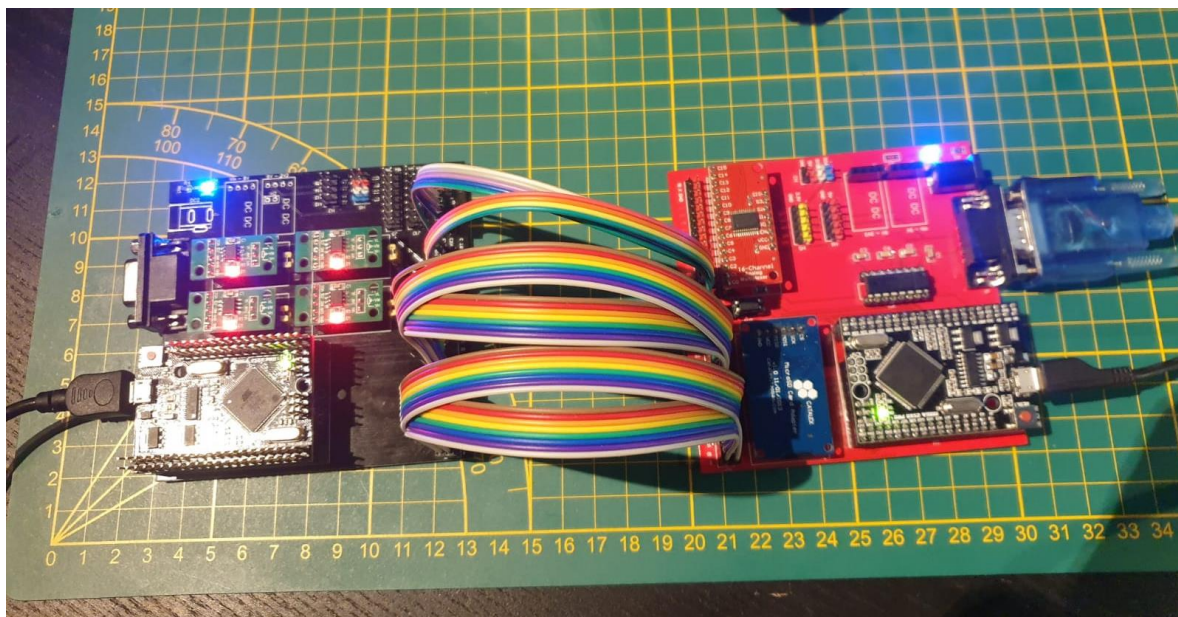
0	Starter 1	8	Alam 2
1	Satrtter 2	9	Alam 2
2	Stop	10	Alam 2
3	Error	11	Alam 2
4	Heating	12	Alam 2
5	Alarm 0	13	Alam 2
6	Alarm 1	14	Alam 2
7	Alarm 2	15	Alam 2
- Rpm:** Radio buttons for selecting the engine speed: 0, 540 (selected), 1360, and 2640.
- Analog value:** A table for monitoring analog inputs:
 

Battery 1	2.24	V	-	<div style="width: 50%;"></div>	+
Battery 2	0.0	1.25	-	<div style="width: 25%;"></div>	+
Analog value	0.0	V	-	<div style="width: 0%;"></div>	+
Analog value	0.0	V	-	<div style="width: 0%;"></div>	+

Rysunek 6. Program testowy

Pane sterowania płytki testowej składa się z następujących funkcjonalności:

- *Serial settings* – jest to okienko umożliwiające połączenie z płytką testową poprzez wybrany port COM. Szczegółowe wyjaśnienie tych ustawień znajduje się w dalszej części pracy.
- *Digital state* – pole odpowiedzialne za wszelkie piny cyfrowe mikrokontrolera.
- *Digital out* znajdują się aktywne przyciski sterujące pinami wyjściowymi mikrokontrolera symulujące wszystkie czujniki cyfrowe. Po wciśnięciu danego przycisku kolor kontrolera zostaje ustawiony w stan wysoki. Jeżeli mikrokontroler poprawnie obsłuży zadanie i zostanie wysłane potwierdzenie kolor zmieni się na czerwony. Analogicznie dzieje się w przypadku ustawieniu w stan niski tylko kolor zmienia się na szary.
- *Digital in* – nieaktywne przyciski symulujące wszelkie akcje kontrolera takie jak: ogrzewanie silnika, stop, start, alarm. Stan przycisków zmienia się samoistnie pod wpływem danych z mikrokontrolera. Kolor czerwony oznacza, że pin cyfrowy jest w stanie niskim czyli jest wykonywana jakaś akcja na kontrolerze.
- *RPM* – w tej sekcji istnieje możliwość symulowania obrotów silnika. Do dyspozycji są wartości: 0 – silnik nie pracuje, 520 – silnik pracuje ale nie osiągnął wystarczających obrotów, - 1360 oraz 2640 – silnik osiągnął odpowiednie obroty.
- *Analog value* – pole służące do sterowania potencjometrami cyfrowymi. Przyciski plus i minus służą do ustawiania w górę bądź dół nastawienia potencjometru. Każda zmiana powoduje zmianę napięcia o 0,25V czyli 1/20 napięcia wejściowego.

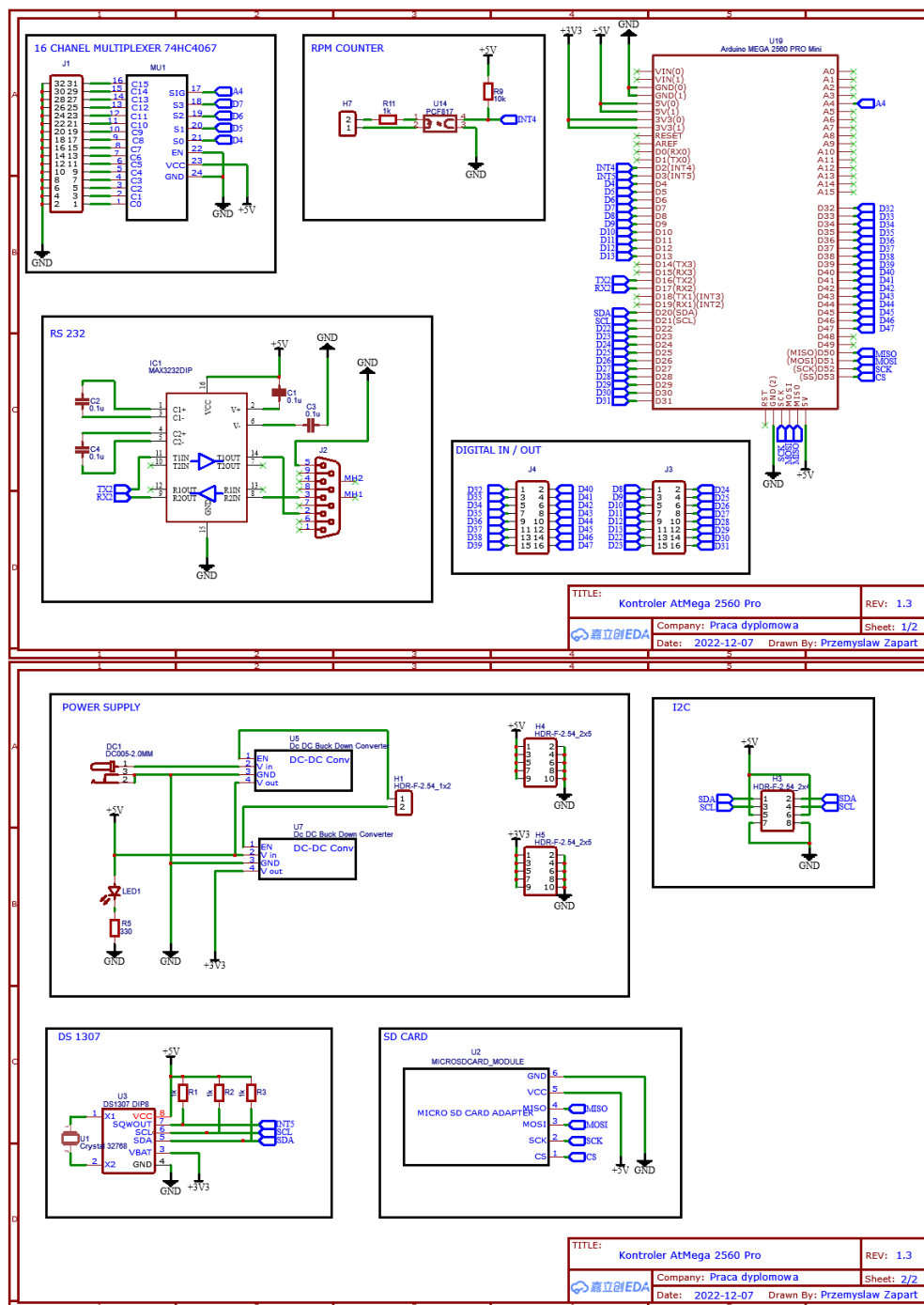


Rysunek 7. Podłączenie całego układu testowego.

## 5. Instalacja projektu – wdrożenie

### 5.1. Kontroler - projekt płytki.

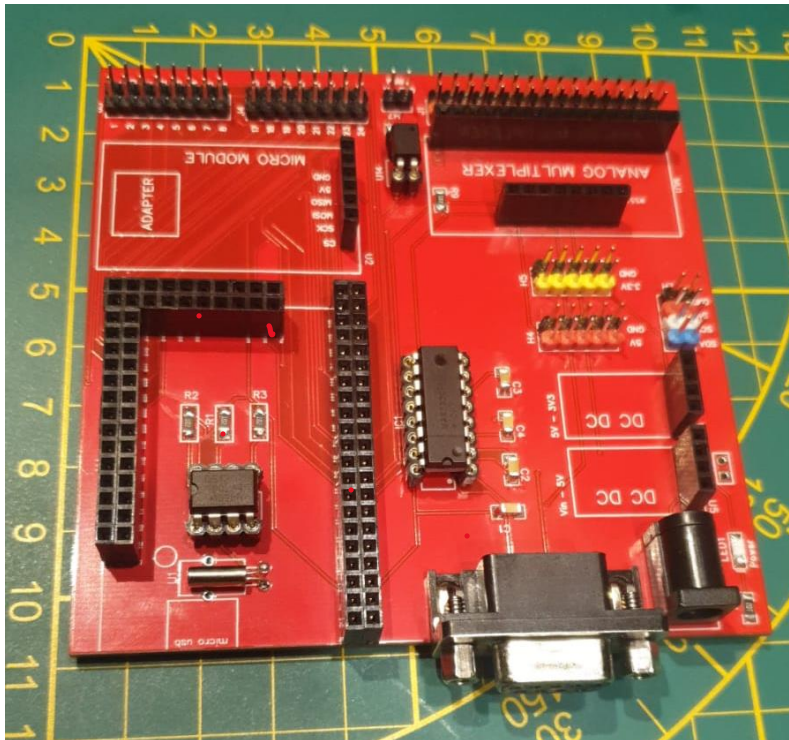
#### 5.1.1. Schemat płytki



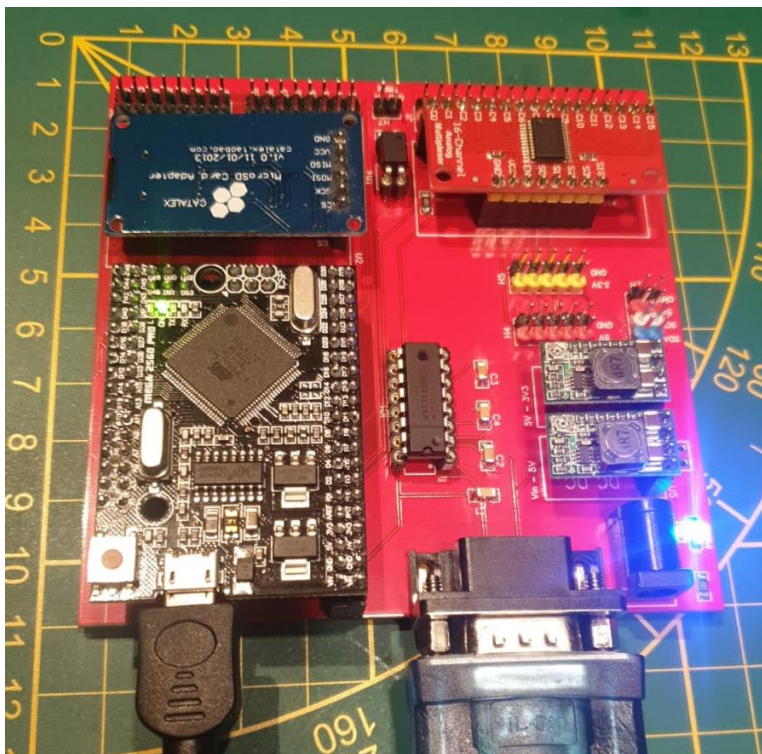
Rysunek 8. Schemat ideowy kontrolera



### 5.1.2. Rozmieszczenie elementów na płytce



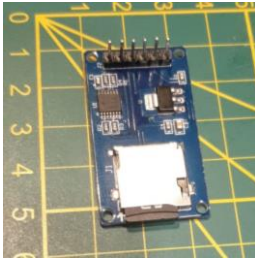
Rysunek 9. Widok gotowej płytki



Rysunek 10. Gotowa płytk z zainstalowanymi komponentami.



### 5.1.3. Moduł karty SD



Rysunek 11. Moduł karty SD

Gotowy moduł karty SD. Moduł działa z magistralą SPI przy zasilaniu od 4.5 V do 5.5V. Pin CS (chip select) podłączony do pinu Arduino 53.

### 5.1.4. Multiplexer



Rysunek 12. Multiplexer CD74HC4067

16 kanałowy analogowo / cyfrowy multiplexer / demultiplekser CD74HC4067. Działa w zakresie 3V do 5V. Cztery cyfrowe piny (S0, S1, S2, S3) służą do sterowania adresem chipu podłączone do pinów cyfrowych Arduino D4, D5, D6, D7 oraz pin danych SIG podłączony do pinu analogowego A4.

### 5.1.5. Transmisja szeregową RS – 232



Rysunek 13. Podłączenie RS 232

Komunikacja transmisji szeregowej do połączenia z panelem sterowania. Standardowe wyjście 9-pinowe **SUB – D** z żeńską końcówką. Realizowane za pomocą układu max 232.

Podłączone do portu **USART 2** mikrokontrolera. **USART** (Uniwersalny asynchroniczny nadajnik-odbiornik, od ang. *universal asynchronous receiver-transmitter*) jest to układ scalony służący do asynchronicznego przekazywania i odbierania informacji poprzez port szeregowy. Zawiera on konwerter równoległo-szeregowy (ang. *parallel-to-serial*), do konwersji danych przesyłanych z komputera, i szeregowo-równoległy (ang. *serial-to-parallel*), do konwersji danych przychodzących do komputera. **USART** posiada także bufor do tymczasowego gromadzenia danych w przypadku szybkiej transmisji. Transmisję informacji złączem szeregowym inicjuje tzw. bit startowy (w każdym przypadku jest to logiczne zero). W dalszej kolejności przesłana zostaje informacja w postaci 7, 8 lub 9 kolejnych bitów (w zależności od ustalonej konfiguracji urządzenia). Za zakończenie transmisji odpowiedzialny jest tzw. bit stopu (logiczna jedynka). Całość tworzy tzw. ramkę **USART**, która zawiera w sobie kompletną przesyłaną informację.

#### 5.1.6. Pomiar prędkości obrotów silnika



Rysunek 14. Widok koła zamachowego z wierćem zębatym

Pomiar prędkości obrotowej następuje za pomocą zliczania zębów na dwumasowym kole zamachowym za pomocą czujnika magnetycznego. Czujnik wykrywa zbliżający się ząb a następnie pole magnetyczne przemieniane jest w impuls elektryczny który można odczytać za pomocą mikrokontroler. Każdy silnik ma określoną liczbę zębów na kole. Program zlicza wszystkie impulsy występujące w ciągu sekundy a następnie oblicza prawidłową wartość korzystając ze wzoru:

$$\text{RPM} = \text{Ilość impulsów} / \text{ilość zębów} * 60$$



Rysunek 15. Moduł RPM kontrolera

Układ odpowiedzialny za realizowanie tego zadania znajduje się na płycie kontrolera. Składa się z optoizolatora PCF 817 oraz rezystora 10 k $\Omega$  podciągniętego do układu zasilania po stronie tranzystora oraz 1k $\Omega$  po stronie diody.

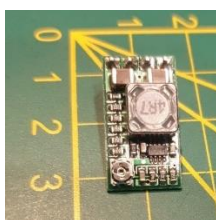
#### 5.1.7. Moduł RTC DS1307



Rysunek 16. Moduł RTC kontrolera

Zegar czasu rzeczywistego (ang. Real-Time Clock RTC) – element systemów komputerowych służący do odliczania czasu niezależnie od stanu maszyny (pracy, zablokowania, wyłączenia). Na płycie został zainstalowany moduł RTC DS1307 podłączony za pomocą magistrali TWI. Jest to szeregową, dwukierunkową magistralę służącą do przesyłania danych w urządzeniach elektronicznych. Została opracowana przez przedsiębiorstwo Philips na początku lat 80. Znana również pod akronimem IIC, którego angielskie rozwinięcie *Inter-Integrated Circuit* oznacza „pośrednik pomiędzy układami scalonymi”.

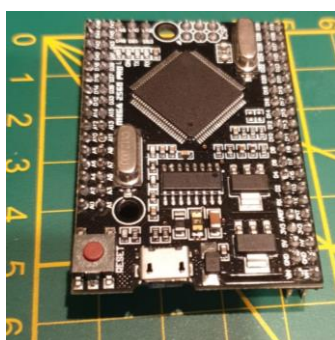
#### 5.1.8. Przetwornica DC – DC Step Down 3A



Rysunek 17. DC - DC

Przetwornica napięcia, nazywana także konwerterem mocy lub przekształtnikiem napięcia – urządzenie elektryczne lub elektromechaniczne pozwalające na zasilanie odbiorników energii elektrycznej z układów zasilających, których parametry prądowo-napięciowe nie pozwalają na bezpośrednie połączenie z odbiornikiem. Zadaniem konwertera mocy jest zmiana wartości natężenia i napięcia w sposób odpowiadający wymaganiom zasilanego odbiornika, z możliwie najmniejszymi stratami mocy. Niskie straty mocy są równoznaczne z wysoką sprawnością konwertera. Na płytce znajdują się dwie przetwornice prądu stałego zapewniające napięcie 5V oraz 3.3V.

### 5.1.9. Arduino mega pro



Rysunek 18. Widok Arduino mega PRO

**Atmega2560** Wysokowydajny, energooszczędny 8-bitowy mikrokontroler Microchip AVR® RISC łączy w sobie 256 KB pamięci flash ISP, 8 KB SRAM, 4 KB EEPROM, 86 linii I/O ogólnego przeznaczenia, 32 rejestry robocze ogólnego przeznaczenia, licznik, sześć elastycznych timerów / liczników z trybami porównania, PWM, cztery kanały USART, 16-kanałowy 10-bitowy przetwornik A/D oraz interfejs JTAG do debugowania na chipie. Urządzenie działa w zakresie 4,5-5,5 woltów.

## 5.2. Sensory

Projekt przewiduje ciągłe monitorowanie podłączonych sensorów i odpowiednie reagowanie na ich stan. Najbardziej podstawowym podziałem sensorów jest podział na sensory analogowe oraz cyfrowe. Czujnik cyfrowy generuje nieciągły, dyskretny sygnał wyjściowy, a czujnik analogowy generuje sygnał, który ma charakter ciągły. Czujniki analogowe mają na celu przekształcenie na sygnał elektryczny parametrów środowiskowych, które nie są związane z elektrycznością. Sygnały wyjściowe obu rodzajów czujników muszą być przekształcone na format cyfrowy (procesor przetwarza tylko cyfrowe dane).

### **5.2.1. Sensory cyfrowe**

Sensory te są sensorami dwustanowymi przypominające proste przełączniki. Jeżeli sensor jest aktywny zostaje zamknięty obwód elektryczny umożliwiający przepływ prądu do mikrokontrolera a następnie podjęcie odpowiednio wcześniej zdefiniowanego zadania. Sensory cyfrowe podłączone do urządzenia to m.in.: ciśnienie oleju, temperatura silnika, poziom paliwa, obroty silnika.

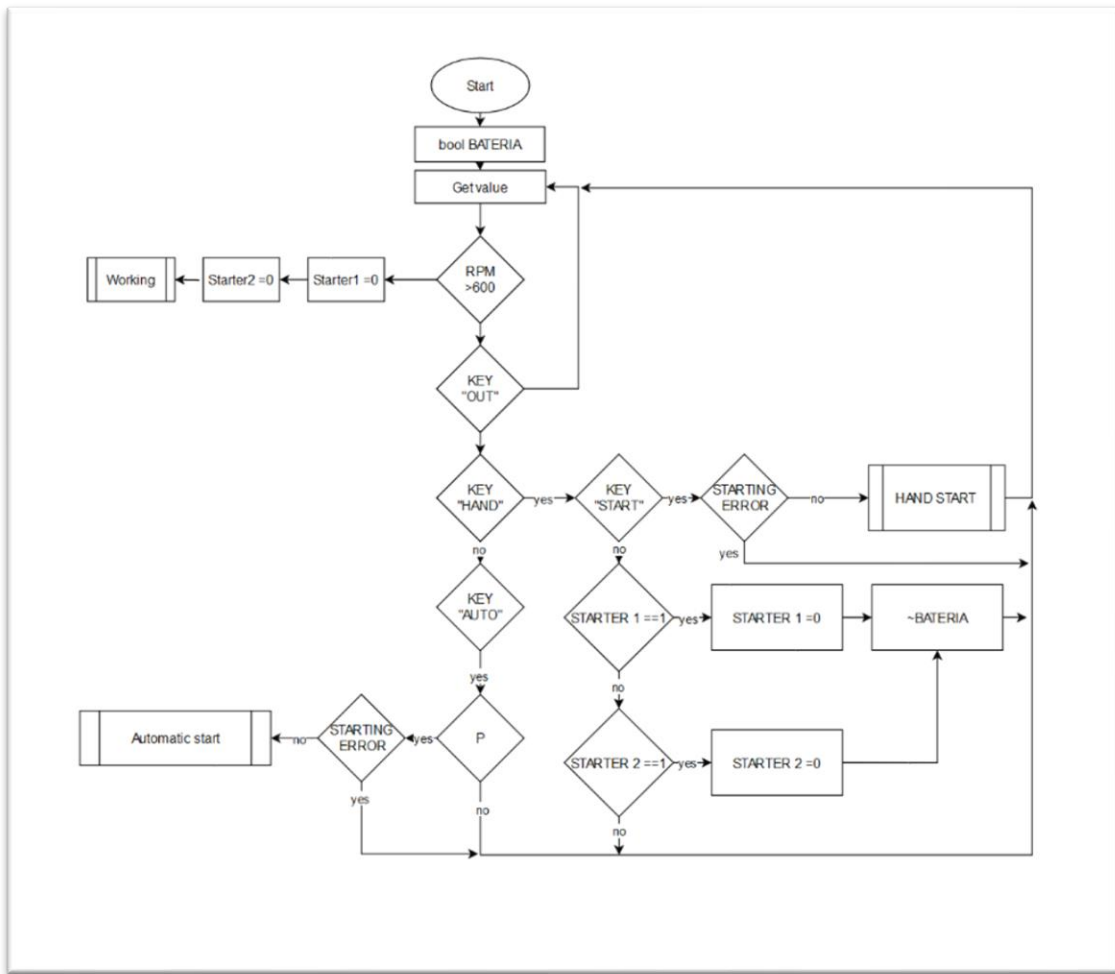
### **5.2.2. Sensory analogowe**

Analogowe czujniki generują sygnał wyjściowy w postaci określonego napięcia. Mają za zadanie przekształcić na sygnał elektryczny parametrów które nie są związane z elektrycznością takich jak: temperatura, ciśnienie oleju czy poziom paliwa. Każdy sensor ma zdefiniowane podstawowe parametry jak wartość *minimalna* i *maksymalna* po przekroczeniu których, zostanie włączony alarm oraz wartość *range* służącą określaniu wielkości pola wyświetlania.

## **5.3. Kontroler. Oprogramowanie**

Głównym założeniem oprócz podstawowej funkcjonalności jest, aby programowanie kontrolera pracowało w sposób nieblokujący wykonywanie innych zadań.

### 5.3.1. Program główny

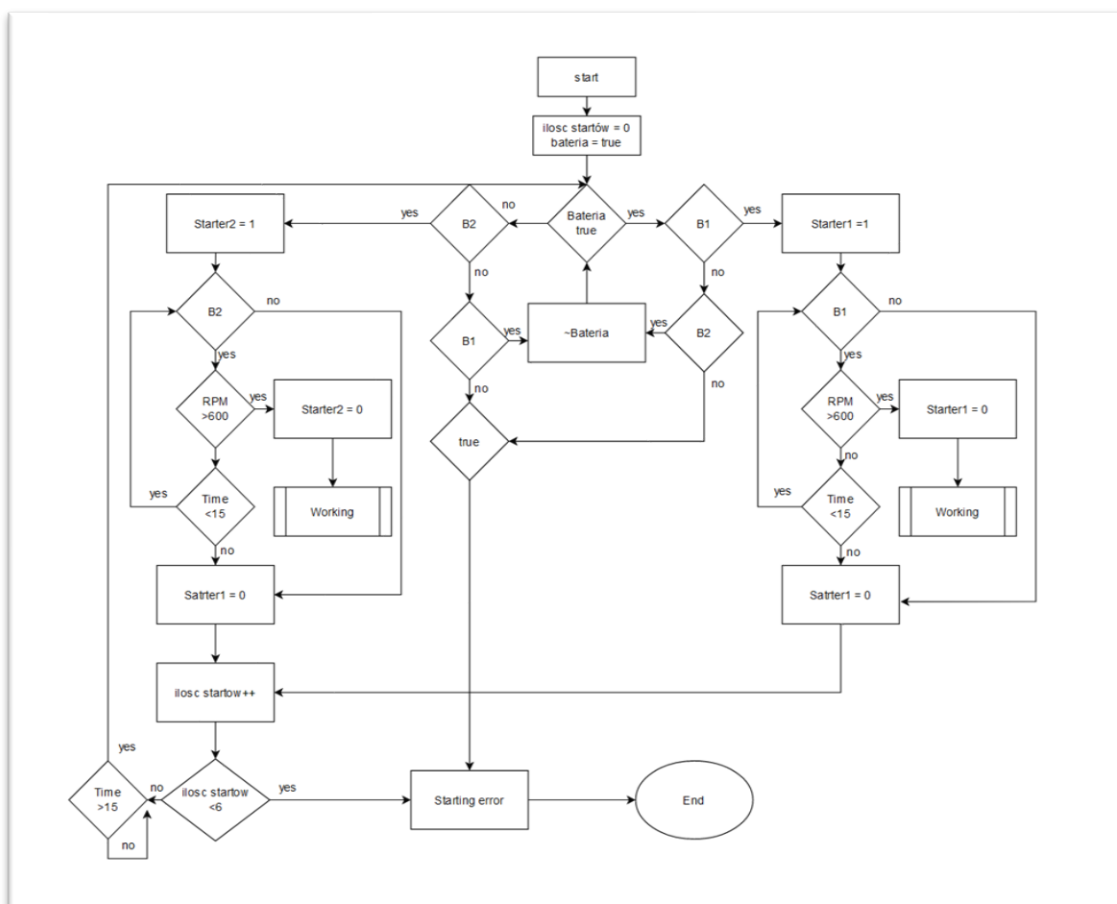


Rysunek 19. Program główny

Program główny wykonywany jest, jeżeli system jest w stanie gotowości. Zasada działania algorytmu polega wstępnym pobraniu wszelkich wartości a następnie na sprawdzeniu pozycji przełącznika głównego. Jeżeli wartość RPM jest większa niż 600 obrotów na minutę oznacza to, że silnik pracuje. Startery zostają wyłączone a program przechodzi w tryb WORKING. Następnie sprawdzana jest pozycja przełącznika głównego. Jeżeli przełącznik jest w trybie OUT to sprawdzane są tylko podstawowe wartości. W trybie HAND możliwe jest ręczne uruchomienie maszyny za pomocą przycisku START. Jeżeli zmienna STARTING\_ERROR zwraca wartość *false* jest możliwy rozruch ręczny. Algorytm HAND START opisany poniżej. Jeżeli przycisk start nie jest wciśnięty program sprawdza czy starter 1 lub starter 2 jest włączony. Jeżeli tak oznacza to, że był rozruch ręczny i wartość zmiennej BATERIA zostaje ustawiona na przeciwną a wybrany starter zostaje wyłączony. Jeżeli przycisk START jest wciśnięty i silnik osiągnie wymagane minimalne obroty program przejdzie w tryb

WORKING. Jeżeli przełącznik jest ustawiony w pozycji AUTO w przypadku spadku ciśnienia w instalacji głównej systemu oraz brak zgłoszonego błędu STARTING ERROR zostaje wywołany program startu automatycznego.

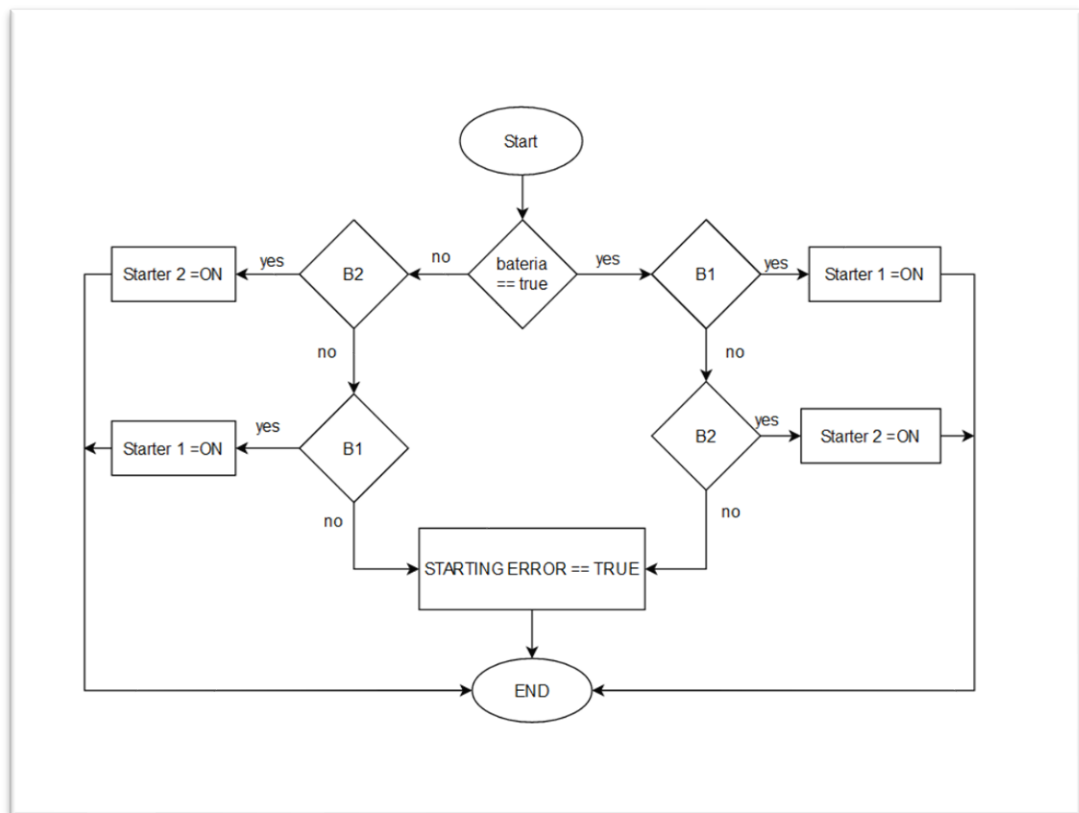
### 5.3.2. Start automatyczny



Rysunek 20. Start Automatyczny

Start odbywa się bez udziału operatora a głównym czynnikiem jest ciśnienie wody w instalacji. Zasada działania algorytmu. Zmienna bateria przechowuje informacje na temat ostatnio użytej baterii. Do rozruch każdorazowo wybierana jest inna bateria, chyba że, aktualnie wybrana bateria zgłasza błąd. W takim wypadku używana jest jedyna sprawna bateria. Ilość prób uruchomienia maszyny wynosi maksymalnie 6. Każda próba trwa maksymalnie 15s. Pomiędzy każdą próbą następuje przerwa trwająca 15 sekund. W wypadku osiągniętej wymaganej ilości obrotów procedura zostaje przerwana a program przechodzi w tryb pracuje. W przypadku w przypadku niepowodzenia włączony zostaje alarm a zmienna STARTING\_ERROR przyjmuje wartość *true*.

### 5.3.3. Start ręczny

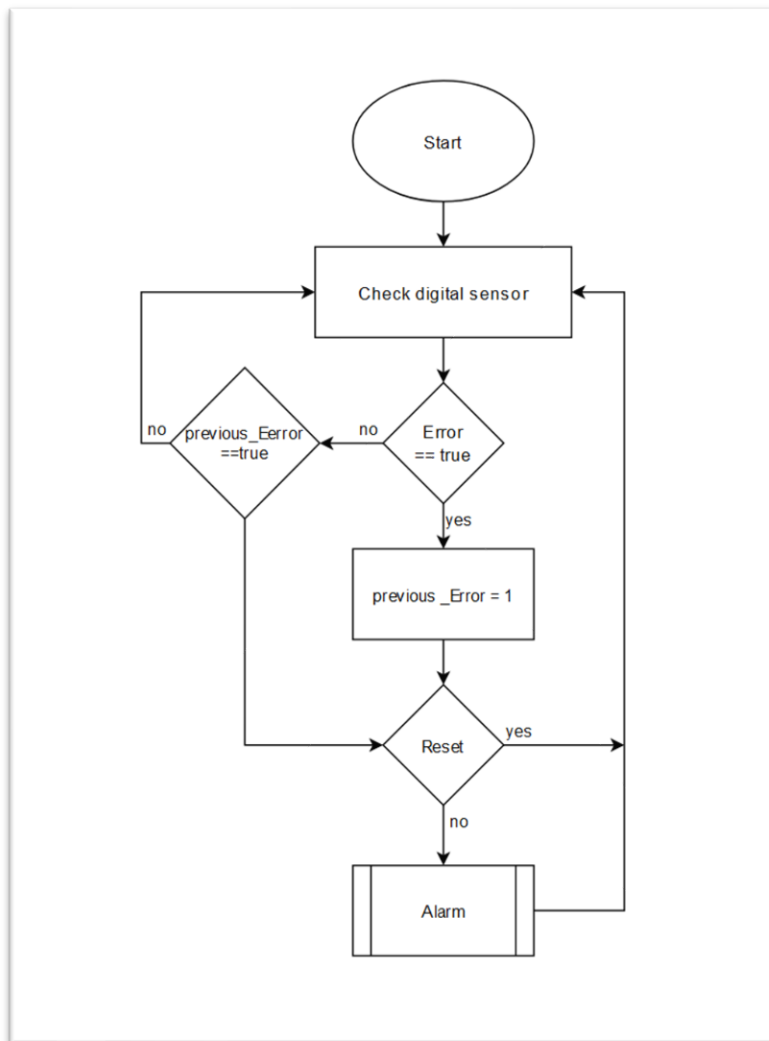


Rysunek 21. start ręczny

Start ręczny odbywa się poprzez wciśnięcie przycisku start na panelu sterowania. W zmiennej *bateria* przechowywana jest informacja na temat użycia poprzedniej baterii. Program sprawdza zmienną i w przypadku wartości *true* jako pierwsza zostanie wybrana do rozruchu bateria 1 (B1). Jeżeli bateria zgłosi błąd to zostanie wybrana bateria 2 (B2). W przypadku gdy obie baterie zgłaszają błąd zmienna *STARTING\_ERROR* ustawiana jest na wartość *true*. Za każdym razem po wciśnięciu przycisku *START* zostanie wybrana inna bateria.



#### 5.3.4. Pobierz wartości sensorów cyfrowych



Rysunek 22. Pobierz wartości.

Zasada działania algorytmu. Każdorazowe uruchomienie czujnika cyfrowego powoduje włączenie alarmu. Alarm pozostaje aktywny do momentu wciśnięcia przycisku RESET a stan czujnika nie ma już znaczenia. Zmienna pomocnicza *previous\_error* przechowuje stan z poprzedniego pomiaru. Jeżeli sensor jest aktywny to wartość zmiennej zostaje ustawiona na *true*. Jeżeli sensor jest już nieaktywny sprawdzana jest wartość zmiennej *previous\_error*. Jeżeli zwraca ona wartość *true* alarm zostanie włączony. Ma to zapobiec migotaniu alarmu. Raz włączony alarm można tylko zresetować.

### 5.3.5. Stop

Nie ma możliwości zatrzymania silnika w sposób automatyczny. Aby wyłączyć pracujący silnik należy wcisnąć przycisk STOP. Zostanie odcięty dopływ paliwa za pomocą cewki magnetycznej do momentu, kiedy obroty silnika nie osiągną wartości 0 RPM. Po upływie 5 sekund cewka zostanie wyłączona i będzie możliwy ponowny rozruch silnika.

### 5.3.6. Reset

Raz załączony alarm nie wyłączy się samoistnie. Należy wcisnąć przycisk RESET, aby wyłączyć wszystkie błędy. Kontroler dopiero po upływie 5 sekund ponownie zmierzy wartość każdego sensora. Każdy błąd musi zostać odnotowany i zresetowany.

### 5.3.7. Zapis na kartę SD

Kontroler zapisuje podstawowe dane na kartę SD co 5 minut uwzględniając wszystkie dane analogowe. Dane są zapisywane również w przypadku wystąpienia zdarzenia takiego jak: wciśnięty przycisk HAND, OUT, AUTO, START, STOP, RESET na panelu sterowania bądź aktywacji czujnika cyfrowego. Dane zapisywane są w pliku **event.csv**. Format zapisu danych:

<Date;Time;Event;Analog1;Analog2.....AnalogN;>

### 5.3.8. Pomiar RPM

Transoptor jest podłączony do pinu przerwania mikrokontrolera. Każdy impuls na kole zębatym powoduje wywołanie odpowiedniej funkcji w programie a następnie zwiększenie odpowiedniej wartości o 1. Co jedną sekundę obliczane są aktualne obroty ze wzoru:

$$\underline{\text{Ilość zębów} / \text{ilość impulsów} * 60}$$

### 5.3.9. Pamięć eeprom

Użytkownik może samodzielnie zdefiniować podstawowe parametry sensorów takie jak nazwa oraz w przypadku sensorów analogowych dodatkowo jednostkę, wartość minimalną, maksymalną oraz zasięg. Można również przypisać nowy numer seryjny. Wszystkie dane zostaną zapisane do pamięci eeprom. W przypadku ponownego uruchomienia żadne dane nie zostaną utracone. Istnieje również możliwość przywrócenia ustawień fabrycznych. Do pamięci eeprom zostaną przypisane ustawienia początkowe.

#### **5.3.9.1. Ustawienia sensora analogowego**

Każdy sensora analogowy posiada właściwości, które można w dowolny sposób zmienić. Są to nazwa, składająca się maksymalnie z 15 znaków, jednostka w jakich wyrażona jest ta wartość, składająca się z maksymalnie 5 znaków, wartość minimalna, maksymalna oraz zasięg składające się maksymalnie z 3 znaków. Sensor analogowy ma do dyspozycji 30 bajtów. Adres pierwszego sensora zaczyna się od pozycji 1400.

#### **5.3.9.2. Ustawienia sensora cyfrowego**

Cyfrowe wejścia, wyjścia oraz status silnika i sensorów analogowych podlegają tej samej zasadzie zapisów do pamięci Eeprom. Różni je jedynie adres numeru startowego. Cyfrowe wejścia zaczynają swoją numerację od adresu 200 pod którym znajduje się nazwa sensora składająca się maksymalnie z 15 znaków. Następnie znajduje się jeden bajt określający tym kontaktu. Każdy sensor ma do dyspozycji 20 bajtów. Adres pierwszego cyfrowego wyjścia to 520 a adres pierwszego stanu silnika to 840.

#### **5.3.9.3. Numer seryjny**

Numer seryjny zapisany jest na maksymalnie 20 bajtach począwszy od adresu 20

#### **5.3.9.4. Wersja oprogramowania**

Wersja oprogramowania zapisana jest na maksymalnie 20 bajtach począwszy od adresu 40.

#### **5.3.9.5. Ilość zębów**

Jeden bajt o adresie 3. Jest to liczba w zakresie 0 -255.

#### **5.3.9.6. Ustawienia początkowe**

Jeden bajt o adresie 0. Jeżeli wartość wynosi 0 to można wczytać ustawienia początkowe. W przeciwnym razie nie podejmuje akcji.

#### **5.3.10. Wysyłane dane**

Dane z kontrolera będą wysyłane co 500 ms. w formie bloku informacji lub jako odpowiedź na zapytanie panelu sterującego. Każdy blok zaczyna się od określonego znaku a jest zakończony znakiem gwiazdki (\*), każda informacja w bloku kończy się znakiem ukośnika (/).

#### 5.3.10.1. Ramka danych

Aktualny czas w formacie YYYY-MM-DD HH:MM-SS. Dane cyfrowe przesyłane są jako liczba 4 bajtowa w formacie HEX. Dane analogowe przesyłane są jako liczba 3 bajtowa w formacie HEX. Wartość analogowa jest pomnożona przez 10. Ostatnie dwie cyfry są liczbą kontrolną. Jest to liczba 2 bajtowa w formacie HEX oznaczającą całkowitą długość przesyłanego łańcucha znaków. Liczba ta służy sprawdzaniu poprawności przesyłanych danych. Format ramki:

**# <date time> / <digital in> / <digital out> / <engine state> / <checksum> / \***

Możliwy jest również format przesyłający do kontrolera informacje na temat poszczególnych sensorów lub innych ustawień. Format ramki:

**\$/<id>/<nazwa>/<typ>/\*** - ustawienia odczytane z pamięci Eeprom kontrolera (digital in)

**!/<id>/<nazwa>/<typ>/\*** - ustawienia odczytane z pamięci Eeprom kontrolera (digital out)

**&/<id>/<nazwa>/<typ>/\*** - ustawienia odczytane z pamięci Eeprom kontrolera (engine state)

**@/<id>/<nazwa>/<jednostka>/<min>/<max>/<range>/\*** - ustawienia odczytane z pamięci Eeprom kontrolera (sensor analogowy)

**%/S/<numer seryjny>/\*** - numer seryjny kontrolera odczytany z pamięci Eeprom

**%/V/<wersja oprogramowania>/\*** - wersja oprogramowania kontrolera odczytany z pamięci Eeprom.

#### 5.3.11. Odbierane dane

##### 5.3.11.1. Stan przycisków

Dane z panelu sterowania odbierane są co 200 ms. Dane te opisują stan poszczególnych przycisków z panelu sterowania: START, STOP, RESET, HAND, AUTO, OUT. Ramka zaczyna się znakiem startowym (!), dalej znajdują się dwie liczby jednocyfrowe a znakiem kończącym wiadomość jest gwiazdka (\*). Pierwsza liczba określa stan przycisków Start, Stop, Reset. Druga liczba określa stan przycisków HOA. Format ramki:

**!<SSR><HOA>\***

### 5.3.11.2. Ustaw nowe wartości

Komendy z panelu sterowania nie są wysyłane regularnie. Ramka zaczyna się znakiem startowym (#) następnie to rodzaj właściwości do zmiany. Może to być sensor cyfrowy, analogowy, numer seryjny odpowiednio oznaczony literą D, A, S. W przypadku sensorów kolejną wartością jest id sensora a następnie wartości do zmiany. W przypadku numeru seryjnego następną wartością jest właśnie ten numer. Format ramki:

#<D>/<id>/<nazwa>\*

#<A>/<id>/<nazwa>/<jednostka>/<min>/<max>/<range>\*

#<S>/<numer seryjny>\*

### 5.3.11.3. Odczytaj wartości

Po otrzymaniu tej komendy odpowiednie wartości są odczytywane a następnie wysyłane do panelu sterowania. Ramka zaczyna się znakiem startowym (@), następnie znajduje się informacja o rodzaju właściwości do doczytania. A, D dla sensorów, O dla pinów wyjściowych oraz Q aby wysłać wszystkie wartości do panelu sterowania. W przypadku sensorów następną zmienną jest jego id. Wiadomość kończy się gwiazdką. Wszelkie wartości oddzielone są od siebie ukośnikiem. Format ramki:

@<D>/<id>/\*

@<A>/<id>\*

@<O>/<id>\*

@<S>\*

@<Q>\*

## 6. Instrukcja obsługi projektu

Zasada działania projektu zostanie omówiona na przykładzie panelu sterowania. Jest to przykładowy interfejs użytkownika służący do komunikacji z kontrolerem. Cała logika systemu znajduje się po stronie kontrolera i została omówiona we wcześniejszej części opracowania. Panel sterowania służy do wydawania poleceń i wyświetlania wyników pracy kontrolera. Składa się z czterech podstawowych części: pasek zakładek pozwalający przełączać się pomiędzy różnymi ustawieniami programu. Ta część panelu jest zawsze widoczna i dostępna dla użytkownika. Na kolejną część panelu składają się pola wyświetlające aktualny czas, obroty na minutę oraz logiczny stan kontrolera. Również ta część jest zawsze widoczna dla użytkownika. Kolejna część wyświetla zawartość poszczególnych zakładek, w zależności od zakładki wyświetlana jest inna zawartość. Ostatnia sekcja zawiera sześć przycisków i służy do fizycznego sterowania kontrolerem. Sekcja ta jest zawsze dostępna dla użytkownika. Standardowo nazwy wszystkich czujników cyfrowych noszą nazwę pól w których się znajdują. Po nawiązaniu połączenia z kontrolerem wczytywane są właściwe nazwy. Zapobiega to konieczności ponownego ustawiania nazw czujników oraz ich wartości po każdorazowym restarcie panelu sterowania lub kontrolera.

### 6.1. Zakładka MAIN

Main		Settings	Connection																																																										
Date / Time : 2022/01/01 00:00:00		RPM : 0	Status : Not connected																																																										
<div>Digital input</div> <table><tr><td><input type="checkbox"/> Digital input</td><td><input type="checkbox"/> Digital input</td></tr><tr><td><input type="checkbox"/> Digital input</td><td><input type="checkbox"/> Digital input</td></tr><tr><td><input type="checkbox"/> Digital input</td><td><input type="checkbox"/> Digital input</td></tr><tr><td><input type="checkbox"/> Digital input</td><td><input type="checkbox"/> Digital input</td></tr><tr><td><input type="checkbox"/> Digital input</td><td><input type="checkbox"/> Digital input</td></tr><tr><td><input type="checkbox"/> Digital input</td><td><input type="checkbox"/> Digital input</td></tr><tr><td><input type="checkbox"/> Digital input</td><td><input type="checkbox"/> Digital input</td></tr><tr><td><input type="checkbox"/> Digital input</td><td><input type="checkbox"/> Digital input</td></tr></table> <div>Engine message</div> <table><tr><td><input checked="" type="checkbox"/> Engine state</td><td><input checked="" type="checkbox"/> Engine state</td></tr><tr><td><input checked="" type="checkbox"/> Engine state</td><td><input checked="" type="checkbox"/> Engine state</td></tr><tr><td><input checked="" type="checkbox"/> Engine state</td><td><input checked="" type="checkbox"/> Engine state</td></tr><tr><td><input checked="" type="checkbox"/> Engine state</td><td><input checked="" type="checkbox"/> Engine state</td></tr><tr><td><input checked="" type="checkbox"/> Engine state</td><td><input checked="" type="checkbox"/> Engine state</td></tr></table>		<input type="checkbox"/> Digital input	<input type="checkbox"/> Digital input	<input type="checkbox"/> Digital input	<input type="checkbox"/> Digital input	<input type="checkbox"/> Digital input	<input type="checkbox"/> Digital input	<input type="checkbox"/> Digital input	<input type="checkbox"/> Digital input	<input type="checkbox"/> Digital input	<input type="checkbox"/> Digital input	<input type="checkbox"/> Digital input	<input type="checkbox"/> Digital input	<input type="checkbox"/> Digital input	<input type="checkbox"/> Digital input	<input type="checkbox"/> Digital input	<input type="checkbox"/> Digital input	<input checked="" type="checkbox"/> Engine state	<input checked="" type="checkbox"/> Engine state	<input checked="" type="checkbox"/> Engine state	<input checked="" type="checkbox"/> Engine state	<input checked="" type="checkbox"/> Engine state	<input checked="" type="checkbox"/> Engine state	<input checked="" type="checkbox"/> Engine state	<input checked="" type="checkbox"/> Engine state	<input checked="" type="checkbox"/> Engine state	<input checked="" type="checkbox"/> Engine state	<div>Analog values</div> <table><tr><td>Analog value</td><td>0</td><td>V</td><td><input type="text"/></td></tr><tr><td>Analog value</td><td>0</td><td>V</td><td><input type="text"/></td></tr><tr><td>Analog value</td><td>0</td><td>V</td><td><input type="text"/></td></tr><tr><td>Analog value</td><td>0</td><td>V</td><td><input type="text"/></td></tr></table> <div>Digital output</div> <table><tr><td><input type="checkbox"/> Digital output</td><td><input type="checkbox"/> Digital output</td></tr><tr><td><input type="checkbox"/> Digital output</td><td><input type="checkbox"/> Digital output</td></tr><tr><td><input type="checkbox"/> Digital output</td><td><input type="checkbox"/> Digital output</td></tr><tr><td><input type="checkbox"/> Digital output</td><td><input type="checkbox"/> Digital output</td></tr><tr><td><input type="checkbox"/> Digital output</td><td><input type="checkbox"/> Digital output</td></tr><tr><td><input type="checkbox"/> Digital output</td><td><input type="checkbox"/> Digital output</td></tr><tr><td><input type="checkbox"/> Digital output</td><td><input type="checkbox"/> Digital output</td></tr><tr><td><input type="checkbox"/> Digital output</td><td><input type="checkbox"/> Digital output</td></tr></table>		Analog value	0	V	<input type="text"/>	Analog value	0	V	<input type="text"/>	Analog value	0	V	<input type="text"/>	Analog value	0	V	<input type="text"/>	<input type="checkbox"/> Digital output	<input type="checkbox"/> Digital output	<input type="checkbox"/> Digital output	<input type="checkbox"/> Digital output	<input type="checkbox"/> Digital output	<input type="checkbox"/> Digital output	<input type="checkbox"/> Digital output	<input type="checkbox"/> Digital output	<input type="checkbox"/> Digital output	<input type="checkbox"/> Digital output	<input type="checkbox"/> Digital output	<input type="checkbox"/> Digital output	<input type="checkbox"/> Digital output	<input type="checkbox"/> Digital output	<input type="checkbox"/> Digital output	<input type="checkbox"/> Digital output
<input type="checkbox"/> Digital input	<input type="checkbox"/> Digital input																																																												
<input type="checkbox"/> Digital input	<input type="checkbox"/> Digital input																																																												
<input type="checkbox"/> Digital input	<input type="checkbox"/> Digital input																																																												
<input type="checkbox"/> Digital input	<input type="checkbox"/> Digital input																																																												
<input type="checkbox"/> Digital input	<input type="checkbox"/> Digital input																																																												
<input type="checkbox"/> Digital input	<input type="checkbox"/> Digital input																																																												
<input type="checkbox"/> Digital input	<input type="checkbox"/> Digital input																																																												
<input type="checkbox"/> Digital input	<input type="checkbox"/> Digital input																																																												
<input checked="" type="checkbox"/> Engine state	<input checked="" type="checkbox"/> Engine state																																																												
<input checked="" type="checkbox"/> Engine state	<input checked="" type="checkbox"/> Engine state																																																												
<input checked="" type="checkbox"/> Engine state	<input checked="" type="checkbox"/> Engine state																																																												
<input checked="" type="checkbox"/> Engine state	<input checked="" type="checkbox"/> Engine state																																																												
<input checked="" type="checkbox"/> Engine state	<input checked="" type="checkbox"/> Engine state																																																												
Analog value	0	V	<input type="text"/>																																																										
Analog value	0	V	<input type="text"/>																																																										
Analog value	0	V	<input type="text"/>																																																										
Analog value	0	V	<input type="text"/>																																																										
<input type="checkbox"/> Digital output	<input type="checkbox"/> Digital output																																																												
<input type="checkbox"/> Digital output	<input type="checkbox"/> Digital output																																																												
<input type="checkbox"/> Digital output	<input type="checkbox"/> Digital output																																																												
<input type="checkbox"/> Digital output	<input type="checkbox"/> Digital output																																																												
<input type="checkbox"/> Digital output	<input type="checkbox"/> Digital output																																																												
<input type="checkbox"/> Digital output	<input type="checkbox"/> Digital output																																																												
<input type="checkbox"/> Digital output	<input type="checkbox"/> Digital output																																																												
<input type="checkbox"/> Digital output	<input type="checkbox"/> Digital output																																																												
<div>Start</div>		<div>Stop</div>	<div>Reset</div>																																																										
<div>H</div>		<div>O</div>	<div>A</div>																																																										

Rysunek 23. Panel sterowania. Zakładka MAIN.

### **6.1.1. Date / Time**

W zakładce wyświetlany jest aktualny czas przysyłany z kontrolera. Program nie ma wbudowanej obsługi czasu. Do czasu nawiązania połączenia z kontrolerem wyświetlana jest godzina „00:00:00”.

### **6.1.2. RPM**

Zakładka wyświetla aktualne obroty silnika.

### **6.1.3. Status**

Zakładka wyświetla stan silnika. Możliwe wartości:

- OUT – w przypadku, kiedy przełącznik jest ustawiony na OUT. Kolor kontrolki zmienia się na szary.
- READY – system jest gotowy i nie ma aktywnych alarmów. Kolor kontrolki zmienia się na zielony.
- STARTING – system przeprowadza procedurę startową lub podczas ręcznego rozruchu. Kolor kontrolki zmienia się na żółty
- RUNNING – silnik pracuje i nie ma aktywnych alarmów. Kolor kontrolki zmienia się na niebieski.
- ERROR- kiedy wystąpi zdarzenie alarmowe. Kolor kontrolki zmienia się na czerwony.

### **6.1.4. Digital input**

Wyświetlany jest stan wszystkich sensorów cyfrowych podłączonych do kontrolera.

### **6.1.5. Analog values**

Pole służy do wyświetlania aktualnych wartości analogowych zmierzonych przez mikrokontroler.

### **6.1.6. Engine status**

W tej zakładce wyświetlany jest stan kontrolera oraz wszystkie sygnały o błędach analogowych sensorów.

### 6.1.7. Digital output

Wyświetlany jest stan wszystkich cyfrowych pinów wyjścia mikrokontrolera, sterującymi przełącznikami.

### 6.1.8. Panel startowy

Panel służący do sterowania kontrolera przez użytkownika. Skład się z sześciu fizycznych przycisków: trzech przycisków sterujących oraz przełącznika **HOA** (HAND, OUT, AUTO) odpowiadającemu stanowi skrzynki rozdzielczej. W zależności od stanu przełącznika HOA dostępne są różne funkcjonalności przycisków sterujących. Przełącznik HOA może być ustalony tylko w jednej pozycji.

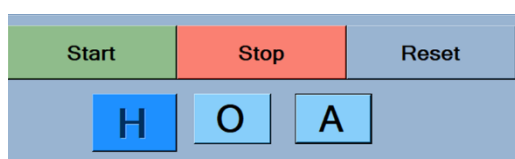
#### 6.1.8.1. Pozycja OUT



Rysunek 24. Panel sterowania. Pozycja OUT

Nie jest możliwe uruchomienie maszyny. Zbierane są podstawowe dane. Przyciski START oraz STOP są zablokowane. Można wcisnąć tylko przycisk RESET w celu zresetowania błędów.

#### 6.1.8.2. Pozycja HAND



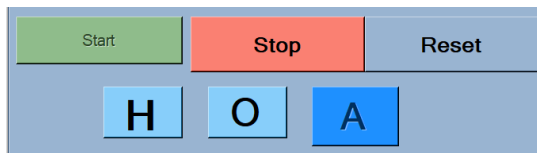
Rysunek 25. Panel sterowania. Pozycja HAND

Ustawienie przełącznika w tej pozycji odblokowuje przycisk START oraz STOP. Po wciśnięciu przycisku START następuje rozruch silnika. Po osiągnięciu wymaganych minimalnych obrotów(600rpm) kontroler przechodzi w tryb **pracuje** a starter zostaje wyłączony. Ponowne wciśnięcie przycisku START nie wywołuje żadnej akcji. Po wciśnięciu przycisku STOP następuje zatrzymanie silnika a ponowny rozruch jest możliwy po ustalonym czasie (5s.). Jeżeli nie została osiągnięta minimalna liczba obrotów po zwolnieniu przycisku



START starter zostaje wyłączony a alarm nie zostaje uruchomiony. Jest możliwe ponowne uruchomienie maszyny.

### 6.1.8.3. Pozycja AUTO



Rysunek 26. Panel sterowania. pozycja AUTO

W tej pozycji nie można uruchomić maszyny ręcznie, przycisk START jest zablokowany. Po otrzymaniu sygnału startowego (niskie ciśnienie w instalacji) następuje automatyczny rozruch urządzenia. Sekwencja startowa składa się z sześciu prób uruchomieniowych przedzielonych przerwami. Każda próba oraz przerwa trwają maksymalnie 15 sekund. Jeżeli silnik osiągnie wymaganą ilość obrotów procedura zostaje przerwana. Do rozruchu baterie są używane naprzemiennie, chyba że, bateria nie ma wymaganego poziomu napięcia. W przypadku awarii obu baterii start jest niemożliwy i zostaje włączony alarm. Jeżeli po sześciu próbach nie nastąpi poprawne uruchomienie silnika włączy się alarm a ponowny rozruch nie będzie możliwy do czasu wciśnięcia przycisku RESET. W przypadku poprawnego uruchomienia silnika, urządzenie można zatrzymać wciskając przycisk STOP dopiero jeśli ciśnienie w instalacji osiągnie wymagany poziom. Nie istnieje możliwość automatycznego zatrzymania się maszyny.

## 6.2. Zakładka SETTINGS

The screenshot shows the 'SETTINGS' tab of a control panel. At the top, there are three tabs: 'Main', 'Settings' (selected), and 'Connection'. Below the tabs, a status bar displays 'Date / Time : 2023-01-15 20:57:00', 'RPM : 0', and 'Status : Out'. The main area is divided into a left sidebar with 'Login', 'General settings' (selected), 'Analog settings', and 'Digital settings'. The 'General settings' section contains input fields for 'Serial number' (900000000) and 'Software ver.' (1.002), along with 'Default settings', 'Get settings', and 'Send settings' buttons. At the bottom, there are 'Start' (green), 'Stop' (red), and 'Reset' (blue) buttons, and three large blue buttons labeled 'H', 'O', and 'A'.

Rysunek 27. Panel sterowania. Zakładka SETTINGS

Zakładka SETTINGS służy do wyświetlania oraz modyfikowania podstawowych parametrów takich jak: nazwa, numer seryjny. Aby dokonywać zmian należy się najpierw zalogować. Przeglądanie ustawień nie wymaga logowania. Użytkownik zostanie automatycznie wylogowany po określonym czasie lub w przypadku przełączenia pomiędzy zakładkami. Bez logowania wszelkie pola edycyjne oraz przyciski SEND SETTINGS pozostają nieaktywne.

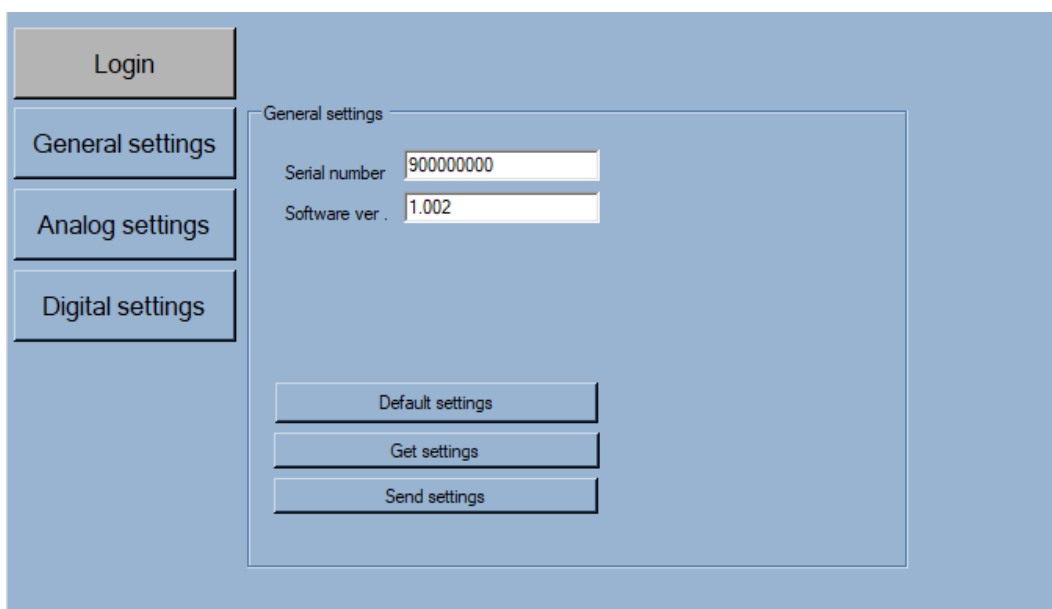
### 6.2.1. LOGIN

The screenshot shows a login window with a black box containing ten red asterisks (\*\*\*\*\*). Below this, the text 'Incorrect password.' is displayed in red. At the bottom, there are three buttons: 'Cancel', 'Reset', and 'Login'.

Rysunek 28. Panel sterowania. Okienko logowania.

Przycisk LOGIN otwiera okienko do logowania. W czarnym polu należy wpisać hasło a przyciskiem LOGIN je zatwierdzić. W przypadku wpisania błędnego hasła zostanie wyświetlony komunikat o błędzie INCORRECT PASSWORD. Przycisk RESET umożliwia ponowne wpisanie hasła a przycisk CANCEL powoduje powrót do poprzedniej strony bez logowania. Ilość prób logowania nie jest ograniczona. Długość hasła ustalona na maksymalnie 10 znaków. Każdorazowe przełączenia pomiędzy zakładkami powoduje wylogowanie się z trybu edycji. Ustawiony jest również maksymalny czas możliwości zmian, po którym użytkownik zostanie wylogowany. Jeżeli użytkownik jest zalogowany, przy ponownej próbie logowania zostanie wyświetlony komunikat YOU ARE ALREADY LOGGED IN a okienko logowania nie zostanie wyświetlone.

### 6.2.2. GENERAL SETTINGS



Rysunek 29. Panel sterowania. Okienko GENERAL SETTINGS

W zakładce GENERAL SETTINGS użytkownik ma możliwość zmiany numeru seryjnego urządzenia. Po poprawnym zalogowaniu należy wpisać nowy numer składający się maksymalnie z 20 znaków i wcisnąć przycisk SEND SETTINGS. Aby wczytać wszystkie ustawienia z kontrolera należy wcisnąć przycisk GET SETTINGS. W celu przywrócenia ustawień fabrycznych należy wcisnąć przycisk DEFAULT SETTINGS. Do pamięci eeprom kontrolera zostaną wczytane ustawienia początkowe a następnie wysłane do panelu sterowania. Wszelkie zmiany wprowadzone przez użytkownika zostaną utracone. Nie jest możliwa zmiana wersji oprogramowania.

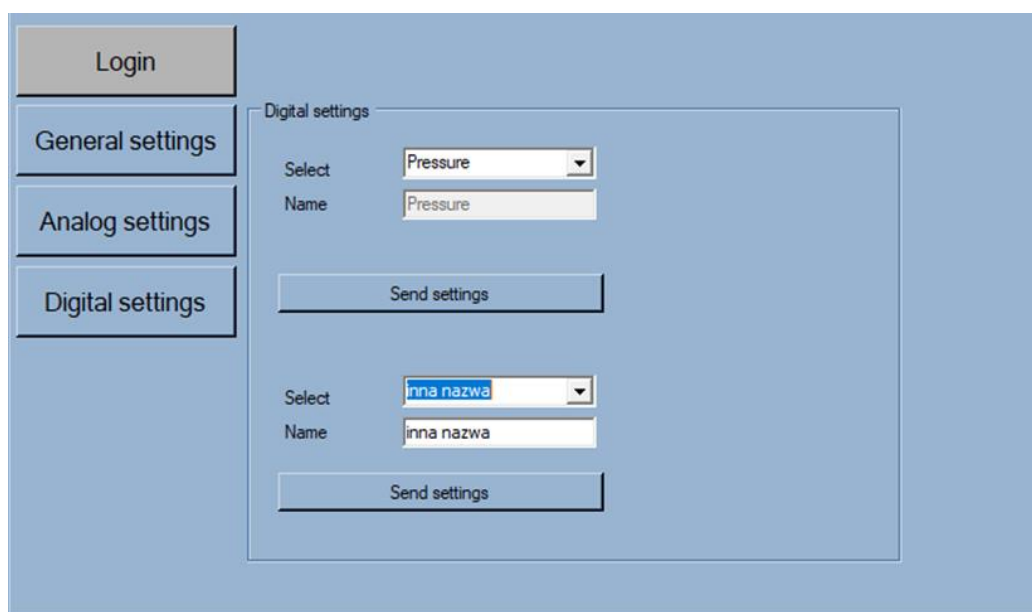
### 6.2.3. ANALOG SETTINGS

Analog settings	
Select	Battery 1
Name	New name
Unit	V
Min value	2
Max value	4
Range	5
<button>Send settings</button>	

Rysunek 30. Panel sterowania. Okienko ANALOG SETTINGS

W zakładce ANALOG SETTINGS możliwe jest modyfikowanie wszystkich ustawień sensora nałogowego. Z pola listy rozwijanej należy wybrać sensor. W odpowiednich polach zostaną wyświetlone odpowiednie wartości. Nazwa nie może przekraczać 15 znaków, jednostka maksymalnie 5 znaków. Pozostałe wartości muszą być liczbami całkowitymi. Wartość maksymalna nie może być mniejsza niż minimalna oraz większa niż zasięg. W przeciwnym wypadku przy próbie wysłania ustawień do kontrolera zostanie wyświetlony komunikat o błędzie. Po poprawnym wysłaniu danych wszystkie pola zostaną wyczyszczone i ponownie wczytane zaktualizowanymi danymi z kontrolera. Ponieważ napięcie baterii jest czynnikiem kluczowym w działaniu całego urządzenia nie jest możliwe zmiana jego nazwy oraz jednostki w jakiej jest wyrażona. Możliwa jest natomiast zmiana pozostałych parametrów uwzględniającą logikę ustawień.

#### 6.2.4. DIGITAL SETTINGS



The screenshot shows a web-based control panel with a sidebar on the left containing four buttons: 'Login', 'General settings', 'Analog settings', and 'Digital settings'. The 'Digital settings' button is highlighted. The main area is titled 'Digital settings' and contains two identical configuration blocks. Each block has a 'Select' dropdown menu and a 'Name' text input field. The first block has 'Pressure' selected in the dropdown and 'Pressure' entered in the text field. The second block has 'inna nazwa' selected in the dropdown and 'inna nazwa' entered in the text field. Below each configuration block is a 'Send settings' button.

Rysunek 31. Panel sterowania. Okienko DIGITAL SETTINGS

W zakładce DIGITAL SETTINGS możliwa jest zmiana ustawień sensorów cyfrowych (DIGITAL IN) oraz przełączników (DIGITAL OUT). Parametr jaki można zmienić to jego nazwa, która składa się maksymalnie z 15 znaków. Nie wszystkie dostępne ustawienia są do dyspozycji użytkownika. Część kluczowych sensorów oraz przełączników możliwa jest tylko do wyświetlenia. Pole edycji *name* jest w takim wypadku niedostępne do edycji. Po wciśnięciu przycisku *Send settings* nowe wartości zostaną wysłane do mikrokontrolera a wszystkie pola zostaną wyczyszczone. Nowe wartości zostaną uzupełnione w razie otrzymania odpowiedzi z mikrokontrolera.

### 6.3. Zakładka CONNECTION

Main	Settings	Connection
Date / Time : 2022/01/01 00:00:00	RPM : 0	Status : Not connected
<div>Serial settings</div> <div>Select COM Port    Select Baud Rate    Disconnect    Status <div>▼    19200    ▼</div>    Connect    not conneted Refresh</div>		
<div>Start    Stop    Reset</div> <div>H    O    A</div>		

Rysunek 32. Panel sterowania. Zakładka CONNECTION.

W tej zakładce możliwe jest zdefiniowanie ustawień połączenia ze sterownikiem. Polu Select COM Port należy wybrać odpowiedni port. Przycisk *Refresh* odświeża listę dostępnych portów com. W polu Select *Baud Rate* należy wybrać odpowiednią prędkość połączenia. Standardowo kontroler jest ustawiony na prędkość 19200. Przycisk Connect łączy nas z urządzeniem.

Serial settings

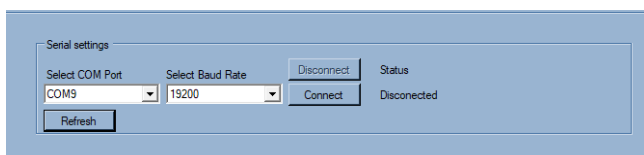
Select COM Port    Select Baud Rate    Disconnect    Status  

COM9    19200    ▼

    Connect    Connected  
Refresh

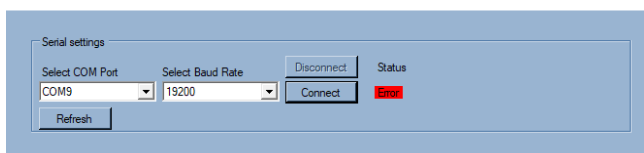
Rysunek 33. Panel sterowania. Connected.

W przypadku poprawnego połączenia przycisk CONNECT zmienia się na nieaktywny a przycisk DISCONNECT zmienia się w aktywny. Status połączenia zmienia się na CONNECTED w zielonym polu.



Rysunek 34. Panel sterowania. Disconnected.

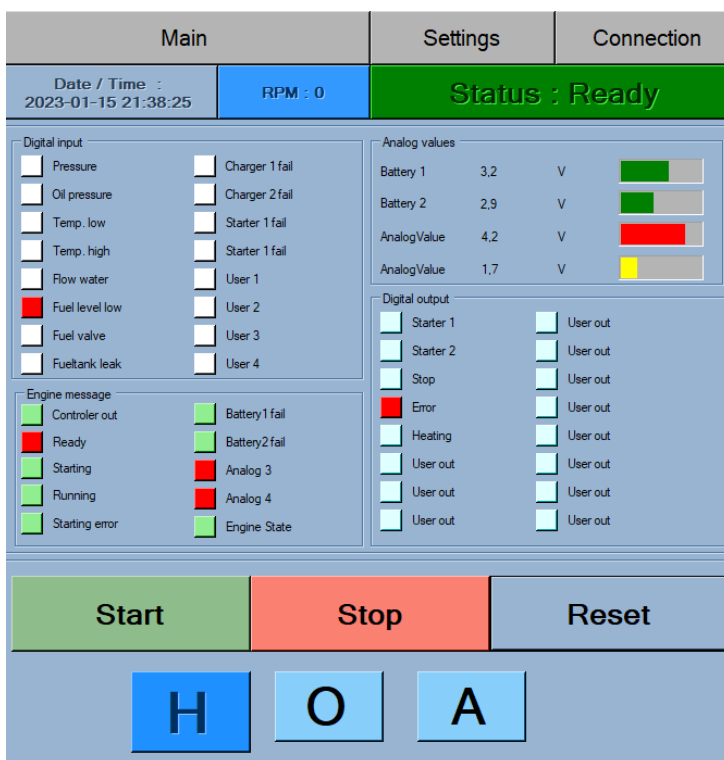
Przycisk DISCONNECT służy do odłączenia urządzenia od wybranego portu. Przycisk Connect jest znowu aktywny a Status zmienia się na DISCONNECTED na neutralnym polu.



Rysunek 35. Panel sterowania. Connection error.

W przypadku niepowodzenia Status połączenia zmieniany jest na Error na czerwonym tle.

## 6.4. Komunikacja z kontrolerem



Rysunek 36. Panel sterowania. Komunikacja z kontrolerem.

Po poprawnym połączeniu się z kontrolerem wszelkie nazwy zostaną wczytane z pamięci eeprom kontrolera. W polu Digital input wyświetlany jest stan czujników cyfrowych. Podczas

pracy silnika wyświetlany jest stan wszystkich czujników. W przeciwnym wypadku niektóre wartości nie są mierzone, ponieważ ich parametry dotyczą pracującego silnika takie jak: ciśnienie oleju, wysoka temperatura oraz przepływ wody w układzie chłodzenia. Z wyjątkiem czujnika *Temp. Low*. Każda zmiana stanu powoduje włączenie się alarmu oraz zapalana jest kontrolka ERROR. W przypadku wykrycia niskiej temperatury silnika włączane jest natychmiast jego ogrzewanie do czasu osiągnięcia wymaganej minimalnej temperatury. Natomiast alarm nie jest uruchamiany a kontrolka ERROR nie jest włączana. Po osiągnięciu wymaganej temperatury ogrzewanie jest automatycznie wyłączane, ale kontrolka *Temp. Low* musi zostać zresetowana za pomocą przycisku RESET. W polu *Engine message* znajdują się informacje na temat stanu maszyny takie jak:

- **Ready** - urządzenie jest gotowe do startu, przynajmniej jedna bateria jest sprawna
- **Out** – urządzenie jest wyłączone, ale zbiera informacje na temat wszystkich sensorów. Start jest niemożliwy.
- **Starting** – urządzenie znajduje się w trybie rozruch.
- **Running** – urządzenie pracuje i nie ma błędów
- **Error** – silnik nie wystartować, ponieważ obie baterie są niesprawne lub w trybie automatycznym po sześciu nieudanych próbach rozruch.

Dodatkowo w tym polu są wyświetlane także informacje na temat stanu baterii lub czujników analogowych. W polu *Analog values* wyświetlany jest stan czujników analogowych. Każdy sensor ma podstawowe właściwości, które można wyświetlić na pasku stanu. Każda wartość może być wyświetlana w trzech kolorach. Kolor żółty informuje, że sensor nie osiągnął minimalnej wymaganej wartości, po której jest uruchamiany alarm i zostaje włączona odpowiednia kontrolka symbolizująca problem. Wyświetlanie w kolorze zielonym informuje, że sensor działa we wcześniej zdefiniowanych wartościach. Kolor czerwony oznacza, że wartość zmierzona na sensorze przekracza wartość maksymalną i jest włączany alarm. W polu *Digital out* zdefiniowany jest stan wszystkich przekaźników podłączonych do kontrolera. Użytkownik może zmienić niektóre ustawienia w zakładce SETTINGS.



## 7. Wnioski

Celem projektu było stworzenie uniwersalnej platformy do testowania oraz wdrażania nowych rozwiązań przy projektowaniu sterowników pompy wodnej w instalacji tryskaczowej. Projekt cechuje się dużą elastycznością w kwestii konfiguracji ustawień. Daje również duże możliwości, jeżeli chodzi o dalszy rozwój projektu we wprowadzaniu nowych funkcjonalności. Wyprowadzone w tym celu zostały piny zasilania 5V oraz 3.3V, piny sterujące magistralą TWI czy nadmiarowa ilość pinów cyfrowych mikrokontrolera. Zastosowanie multipleksera dało możliwość pomiaru 16 wartości analogowych lub dodatkowych pinów cyfrowych. W przypadku uszkodzenia poszczególnych komponentów bardzo łatwo je wymienić, ponieważ projekt zbudowany jest z niezależnych modułów. Panel sterowania został opracowany z myślą o jak najszerszym widoku wszelkich podłączonych komponentów oraz łatwości modyfikowania poszczególnych ustawień mikrokontrolera. Panel sterowania nie był skonstruowany z myślą o końcowym użytkowniku gotowego systemu, ale raczej projektanta danego systemu, dlatego szata graficzna jest bardzo oszczędna.

## 8. Literatura

1. Mark Michaelis - C# 7.0 Kompletny przewodnik dla praktyków
2. Jerzy Grębosz – Opus Magnum
3. Mirosław Kardaś – Mikrokontrolery AVR. Język C

## 9. Źródło

1. [https://en.wikipedia.org/wiki/Visual\\_Studio](https://en.wikipedia.org/wiki/Visual_Studio)
2. <https://www.arduino.cc/en/Guide/Introduction>
3. <https://strefainzyniera.pl/artukul/1037/czujniki-analogowe-i-cyfrowe>
4. <https://www.supon.rzeszow.pl/content/60-instalacje-tryskaczowe>
5. <https://pl.wikipedia.org/wiki/Arduino>
6. <https://poradnikprojektanta.pl/instalacje-tryskaczowe-wytyczne-i-normy/>
7. [https://pl.wikipedia.org/wiki/Przetwornica\\_napi%C4%99cia](https://pl.wikipedia.org/wiki/Przetwornica_napi%C4%99cia)

8. [https://pl.wikipedia.org/wiki/Uniwersalny\\_asynchroniczny\\_nadajnik-odbiornik](https://pl.wikipedia.org/wiki/Uniwersalny_asynchroniczny_nadajnik-odbiornik)
9. <https://www.jarzebski.pl/arduino/komponenty/zegar-czasu-rzeczywistego-rtc-ds1307.html>
10. <https://pl.wikipedia.org/wiki/I%C2%B2C>
11. <https://pl.wikipedia.org/wiki/Tryskacz>

## 10. Spis ilustracji

Rysunek 1. Pompa wodna w instalacji tryskaczowej .....	2
Rysunek 2. Typowa instalacja tryskaczowa.....	6
Rysunek 3. Tryskacz ze szklaną ampułką. ....	6
Rysunek 4. Skrzynka kontrolna WBFP-D4 w instalacji tryskaczowej z wykorzystaniem silnika spalinowego.....	7
Rysunek 5. Płytki testowa .....	11
Rysunek 6. Program testowy .....	12
Rysunek 7. Podłączenie całego układu testowego. ....	14
Rysunek 8. Schemat ideowy kontrolera .....	15
Rysunek 9. Widok gotowej płytki .....	16
Rysunek 10. Gotowa płytki z zainstalowanymi komponentami. ....	16
Rysunek 11. Moduł karty SD.....	17
Rysunek 12. Multiplexer CD74HC4067 .....	17
Rysunek 13. Podłączenie RS 232.....	17
Rysunek 14. Widok koła zamachowego z wieńcem zębatym.....	18
Rysunek 15. Moduł RPM kontrolera.....	19
Rysunek 16. Moduł RTC kontrolera .....	19
Rysunek 17. DC - DC .....	19
Rysunek 18. Widok Arduino mega PRO.....	20
Rysunek 19. Program główny .....	22
Rysunek 20. Start Automatyczny .....	23
Rysunek 21. start ręczny.....	24
Rysunek 22. Pobierz wartości. ....	25
Rysunek 23. Panel sterowania. Zakładka MAIN.....	30
Rysunek 24. Panel sterowania. Pozycja OUT .....	32
Rysunek 25. Panel sterowania. Pozycja HAND.....	32
Rysunek 26. Panel sterowania. pozycja AUTO.....	33
Rysunek 27. Panel sterowania. Zakładka SETTINGS .....	34
Rysunek 28. Panel sterowania. Okienko logowania. ....	34
Rysunek 29. Panel sterowania. Okienko GENERAL SETTINGS .....	35
Rysunek 30. Panel sterowania. Okienko ANALOG SETTINGS.....	36
Rysunek 31. Panel sterowania. Okienko DIGITAL SETTINGS.....	37
Rysunek 32. Panel sterowania. Zakładka CONNECTION.....	38
Rysunek 33. Panel sterowania. Connected.....	38
Rysunek 34. Panel sterowania. Disconnected. ....	39
Rysunek 35. Panel sterowania. Connection error.....	39

Rysunek 36. Panel sterowania. Komunikacja z kontrolerem. ....	39
---	----

## 11. Spis treści

<b>1. WSTĘP .....</b>	<b>2</b>
<b>2. ZAKRES PRACY .....</b>	<b>3</b>
<b>3. CHARAKTERYSTYKA WYBRANYCH NARZĘDZI DO WYKONANIA PRACY .....</b>	<b>3</b>
3.1. VISUAL STUDIO .....	3
3.2. EASY EDA .....	3
3.3. DIAGRAMS.NET .....	4
3.4. PLATFORMA ARDUINO .....	4
3.5. VISUAL MICRO.....	5
<b>4. PROJEKT SYSTEMU .....</b>	<b>6</b>
4.1. ZASADA DZIAŁANIA.....	6
4.2. SKRZYNKA ROZDZIELCZA .....	7
4.3. PODSTAWOWE ZAŁOŻENIA PROJEKTU .....	7
4.4. ZAŁOŻENIA PROJEKTU .....	8
4.4.1. PODSTAWOWE ZAŁOŻENIA KONTROLERA.....	8
4.4.1.1. ZAŁOŻENIA SPRZĘTOWE .....	8
4.4.1.2. ZAŁOŻENIA PROGRAMOWE .....	8
4.4.2. PODSTAWOWE ZAŁOŻENIA PANELU STEROWANIA.....	10
4.5. TESTOWANIE.....	11
4.5.1. PŁYTKA TESTOWA .....	11
4.5.2. DEDYKOWANE OPROGRAMOWANIE PŁYTKI TESTOWEJ.....	12
<b>5. INSTALACJA PROJEKTU – WDROŻENIE .....</b>	<b>15</b>
5.1. KONTROLER - PROJEKT PŁYTKI.....	15
5.1.1. SCHEMAT PŁYTKI .....	15
5.1.2. ROZMIESZCZENIE ELEMENTÓW NA PŁYTCE .....	16
5.1.3. MODUŁ KARTY SD.....	17
5.1.4. MULTIPLEXER .....	17
5.1.5. TRANSMISJA SZEREGOWA RS – 232.....	17
5.1.6. POMIAR PRĘDKOŚCI OBROTÓW SILNIKA .....	18
5.1.7. MODUŁ RTC DS1307 .....	19
5.1.8. PRZETWORNICA DC – DC STEP DOWN 3A.....	19
5.1.9. ARDUINO MEGA PRO.....	20
5.2. SENSORY.....	20
5.2.1. SENSORY CYFROWE .....	21
5.2.2. SENSORY ANALOGOWE .....	21
5.3. KONTROLER. OPROGRAMOWANIE .....	21
5.3.1. PROGRAM GŁÓWNY .....	22
5.3.2. START AUTOMATYCZNY .....	23
5.3.3. START RĘCZNY.....	24
5.3.4. POBIERZ WARTOŚCI SENSORÓW CYFROWYCH.....	25
5.3.5. STOP.....	26
5.3.6. RESET .....	26
5.3.7. ZAPIS NA KARTĘ SD .....	26
5.3.8. POMIAR RPM .....	26

5.3.9.	PAMIĘĆ EEPROM .....	26
5.3.9.1.	USTAWIENIA SENSORA ANALOGOWEGO.....	27
5.3.9.2.	USTAWIENIA SENSORA CYFROWEGO .....	27
5.3.9.3.	NUMER SERYJNY.....	27
5.3.9.4.	WERSJA OPROGRAMOWANIA.....	27
5.3.9.5.	IŁOŚĆ ZĘBÓW.....	27
5.3.9.6.	USTAWIENIA POCZĄTKOWE .....	27
5.3.10.	WYSYŁANE DANE.....	27
5.3.10.1.	RAMKA DANYCH .....	28
5.3.11.	ODBIERANE DANE.....	28
5.3.11.1.	STAN PRZYCISKÓW .....	28
5.3.11.2.	USTAW NOWE WARTOŚCI.....	29
5.3.11.3.	ODCZYTAJ WARTOŚCI .....	29
<b>6.</b>	<b>INSTRUKCJA OBSŁUGI PROJEKTU .....</b>	<b>30</b>
6.1.	ZAKŁADKA MAIN .....	30
6.1.1.	DATE / TIME.....	31
6.1.2.	RPM .....	31
6.1.3.	STATUS.....	31
6.1.4.	DIGITAL INPUT .....	31
6.1.5.	ANALOG VALUES.....	31
6.1.6.	ENGINE STATUS .....	31
6.1.7.	DIGITAL OUTPUT .....	32
6.1.8.	PANEL STARTOWY .....	32
6.1.8.1.	POZYCJA OUT .....	32
6.1.8.2.	POZYCJA HAND.....	32
6.1.8.3.	POZYCJA AUTO .....	33
6.2.	ZAKŁADKA SETTINGS.....	34
6.2.1.	LOGIN .....	34
6.2.2.	GENERAL SETTINGS .....	35
6.2.3.	ANALOG SETTINGS .....	36
6.2.4.	DIGITAL SETTINGS.....	37
6.3.	ZAKŁADKA CONNECTION .....	38
6.4.	KOMUNIKACJA Z KONTROLEREM .....	39
<b>7.</b>	<b>WNIOSKI .....</b>	<b>41</b>
<b>8.</b>	<b>LITERATURA .....</b>	<b>41</b>
<b>9.</b>	<b>ŹRÓDŁO .....</b>	<b>41</b>
<b>10.</b>	<b>SPIS ILUSTRACJI .....</b>	<b>42</b>
<b>11.</b>	<b>SPIS TREŚCI .....</b>	<b>43</b>