

**Wydział Informatyki, Elektroniki i Telekomunikacji**

**Dokumentacja projektowa**

**System testowania poprawności generacji sygnału PWM przez mikrokontrolery**

Autorzy: Michał Markiewicz, Łukasz Piasecki

Kierunek studiów: *Elektronika* *i Telekomunikacja*

Przedmiot: Testowanie i niezawodność

Kraków, 2023

Spis treści

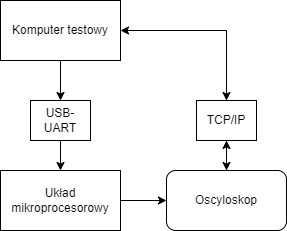
**Nie znaleziono żadnych pozycji spisu treści.**

# Wprowadzenie

# System testowania poprawności generacji sygnału PWM przez mikrokontrolery został stworzony w celu weryfikacji i walidacji poprawności działania mikrokontrolera pod względem generowania sygnałów. System ten ma na celu sprawdzenie, czy mikrokontroler jest w stanie generować oczekiwane sygnały zgodnie z założeniami projektowymi. W ramach testowania poprawności wykorzystano porównywanie sygnałów generowanych przez mikrokontroler z oczekiwanymi wzorcami. W obecnej formie system obejmuje tylko porównanie częstotliwości sygnałów, ale łatwość dodania nowych funkcji umożliwia łatwy rozwój przypadków testowych.

# Metodyka testów

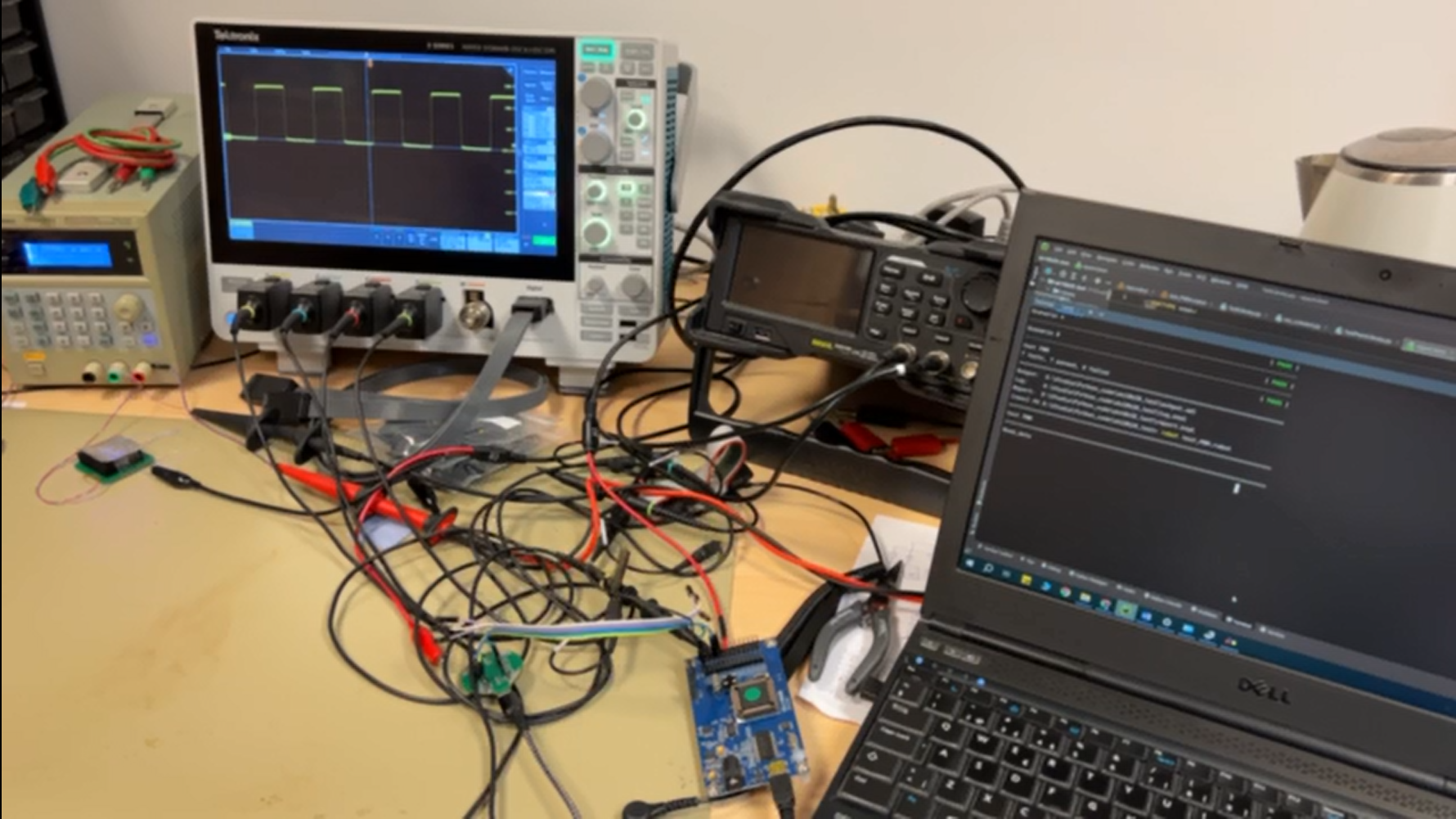
Do przygotowania stanowiska testowego wykorzystano moduł USB-UART, płytkę rozwojową PADUAK 5S-I-S02B[1], oraz oscyloskop.



Zdj. 1 Schemat ideowy

Komputer testowy komunikuje się z płytką rozwojową przez interfejs UART kontrolując sygnał generowany przez mikrokontroler. Zależnie od informacji wysyłanej przez komputer mikrokontroler generuje inny sygnał, który podawany jest na wyjścia płytki rozwojowej.

Do wyjść podpięta jest sonda podłączona do oscyloskopu, która odczytuje sygnał generowany. Oscyloskop jest połączony z komputerem testowym poprzez interfejs TCP/IP. Dzięki temu możliwe jest odczytywanie niezbędnych parametrów z oscyloskopu i porównywanie ich z informacjami wysyłanymi do płytki rozwojowej.



Zdj. 2 Setup testowy

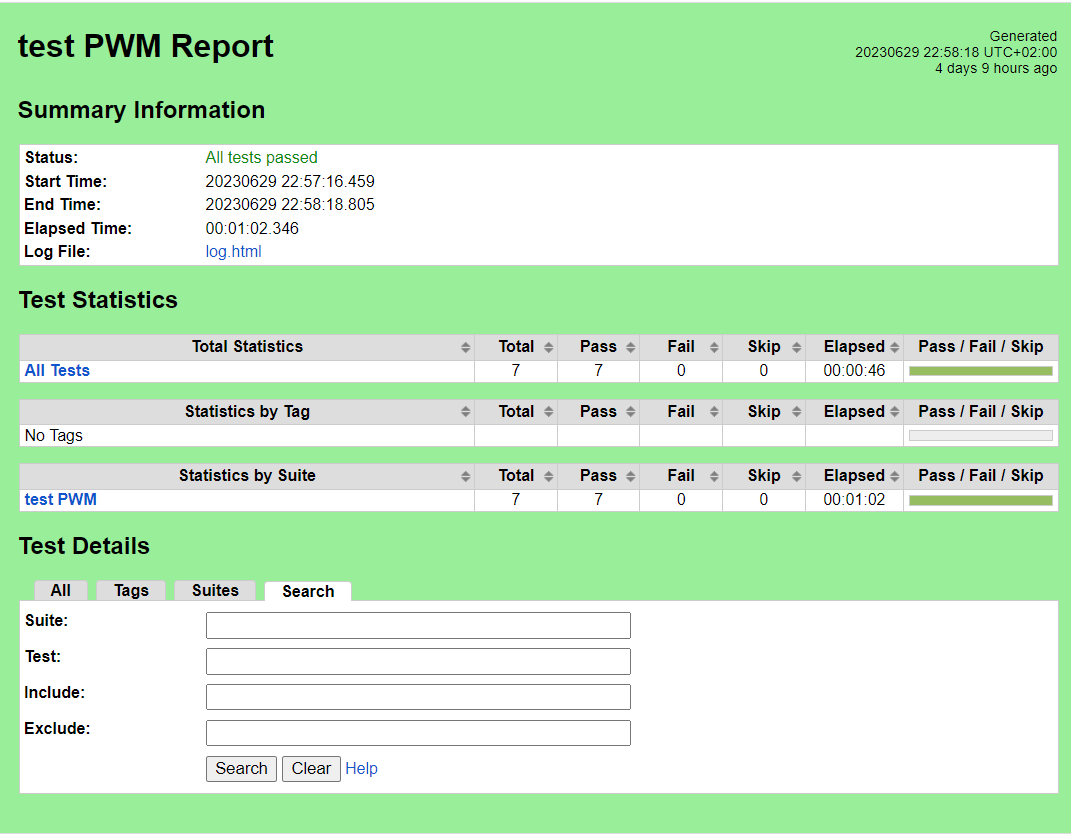
## Robot Framework

W celu automatyzacji testów wykorzystano narzędzie Robot Framework. Robot Framework[2] to otwarte narzędzie do automatyzacji testów, które wykorzystuje język Python. Jest to framework, który umożliwia tworzenie, zarządzanie i wykonywanie testów w sposób zautomatyzowany oraz pozwala pisać czytelne i łatwe do zrozumienia scenariusze testowe, korzystając z czytelnej składni i dostępnych bibliotek. Pozwala on na testowanie różnych typów aplikacji, w tym interfejsów użytkownika, usług sieciowych czy baz danych.

W momencie rozpoczęcia testów nawiązywana jest komunikacja z płytką rozwojową poprzez interfejs UART. Następnie wykonywane są kolejne scenariusze testowe.

## Raport

Program po zakończeniu generuje rozbudowane raporty, które zawierają szczegółowe informacje o przebiegu testów oraz wynikach. Raporty te zawierają m.in. listę wykonanych testów, wyniki dla poszczególnych testów (sukces, niepowodzenie, pominięcie) oraz ewentualne komunikaty błędów. Generowane są również pliki z logami, które zawierają szczegółowe informacje diagnostyczne na temat wykonania testów, takie jak ścieżki dostępu, wartości parametrów czy wykonywane akcje. Dzięki tym generowanym dokumentom można dokładnie analizować przebieg testów, zidentyfikować ewentualne problemy i podjąć odpowiednie działania naprawcze.



Zdj.3. Przykład raportu Robot Framework

# Testy

W celu przetestowania poprawności działania systemu testującego przeprowadzono testy. Przebieg testów można zobaczyć na nagrania umieszczonych w załączniku.

Rezultat testów

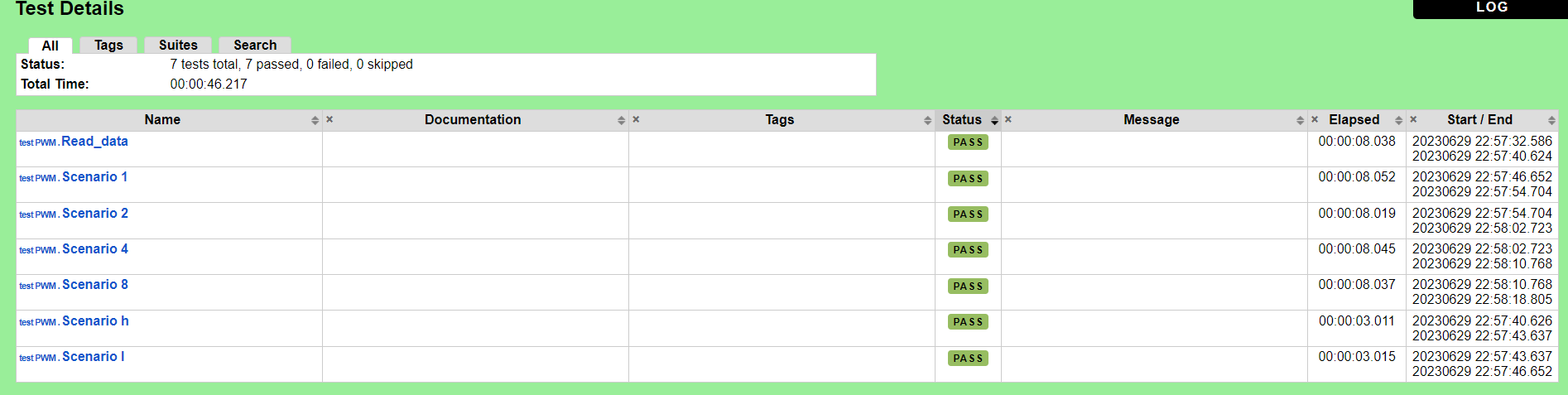
## Scenariusze testowe

Podczas testów sprawdzane są kolejne scenariusze testowe, podczas których wysyłane są za pomocą USB-UART komendy do mikrokontrolera. Po odebraniu polecenia mikrokontroler generuje konkretny sygnał PWM na wyjście.

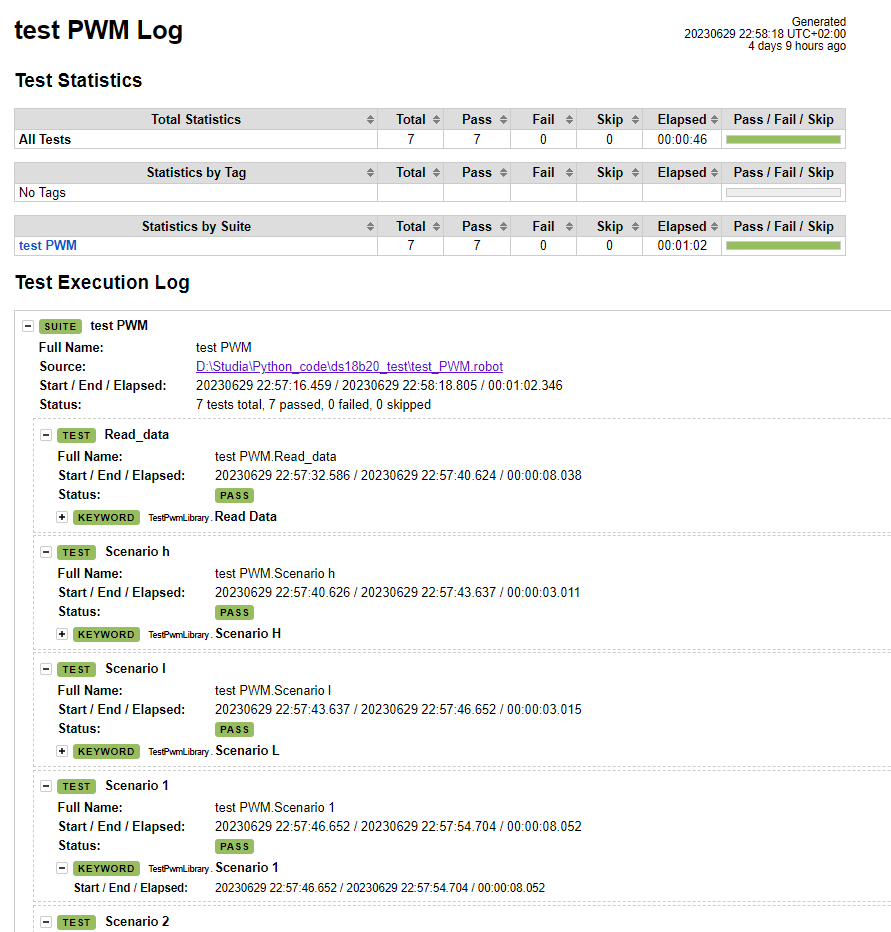
|  |  |
| --- | --- |
| **Scenariusze testowe** | **Opis** |
| Read\_data | Sprawdzenie poprawności komunikacji z oscyloskopem |
| Scenario 1 | Ustawienie sygnału PWM o częstotliwości 4kHz na wyjściu mikrokontrolera i sprawdzenie odczytu oscyloskopu |
| Scenario 2 | Ustawienie sygnału PWM o częstotliwości 2kHz na wyjściu mikrokontrolera i sprawdzenie odczytu oscyloskopu |
| Scenario 4 | Ustawienie sygnału PWM o częstotliwości 1kHz na wyjściu mikrokontrolera i sprawdzenie odczytu oscyloskopu |
| Scenario 8 | Ustawienie sygnału PWM o częstotliwości 500Hz na wyjściu mikrokontrolera i sprawdzenie odczytu oscyloskopu |
| Scenario h | Ustawienie stanu wysokiego na wyjściu mikrokontrolera i sprawdzenie odczytu oscyloskopu |
| Scenario l | Ustawienie stanu niskiego na wyjściu mikrokontrolera i sprawdzenie odczytu oscyloskopu |

## Raport

Po przeprowadzeniu testów został wygenerowany raport. Plik html z raportem znajduje się w załączniku.



Zdj.4. Strona główna raportu



Zdj.5. Strona raportu zawierająca logi z każdego testu

# Podsumowanie

System został stworzony w celu weryfikacji i walidacji poprawności działania mikrokontrolera pod względem generowania sygnałów. Dzięki wykorzystaniu narzędzia Robot Framework możliwe było pisanie czytelnych i łatwych do zrozumienia scenariuszy testowych.

Wyniki testów są porównywane z oczekiwanymi wzorcami oraz prezentowane w rozbudowanych raportach, które zawierają szczegółowe informacje o przebiegu testów oraz wynikach. Generowane są również pliki z logami, umożliwiające analizę przebiegu testów i identyfikację ewentualnych problemów. Scenariusze testowe pozostawiają możliwość rozbudowy, która zdecydowanie jest niezbędna w celu lepszej walidacji poprawności działania badanego mikrokontrolera.

# Załączniki

* PMS150C-TestUnit.C - Program w języku C służący do obsługi zapytań oraz generacji sygnału PWM w mikrokontrolerze
* test\_PWM.robot wraz bibliotekami TestPwmLibrary.py, osc\_connect.py – skrypty służące do automatyzacji przypadków testowych
* log.html, report.html – pliki wynikowe zawierające logi oraz raport z testów.
* Działanie komunikacji komputera z mikrokontrolerem.mp4 – film prezentujący działanie komunikacji, poprzez którą ustawiany był sygnał generowany przez mikrokontroler. Na oscyloskopie widzimy podgląd generowanego sygnału.
* Przebieg testów.mp4 – nagranie przebiegu testów.

# Źródła

[1] <https://datasheet.lcsc.com/lcsc/1912111437_PADAUK-Tech-5S-I-S02B_C394018.pdf>

[2] https://robotframework.org/robotframework/2.1.2/RobotFrameworkUserGuide.html