

# Raport

**Przedmiot:** Zaawansowane zastosowania układów mikroprocesorowych

**Temat:** Symulacja przycisku wiertarki za pomocą płytki nucleo z opcją sterowania za pomocą aplikacji.

**Wykonawcy:**

1. Jakub Bajek
2. Zuzanna Baran
3. Wiktor Kacprzak

**Planowany zakres pracy:**

Analiza działania fizycznego przycisku w układzie rozłączonym oraz w układzie pracy – z wpiętym sterownikiem silnika. Sprawdzenie zakresów napięć i ustalenie materiałów potrzebnych do odwzorowania działania przycisku. Przygotowanie programu mającego dać możliwość uruchomienia silnika, wyboru kierunku obrotów oraz zadanie prędkości obrotowej w zakresie od 0 do 100% wciśnięcia przycisku przez połączenie aplikacji komputerowej z płytką sterującą embedded oraz elementem pośrednim, wykonawczym mającym zasymulować działanie przycisku dla sterownika, o ile taki element pośredni okaże się niezbędny.

**Wykonanie projektu:**

Zajęcia rozpoczęliśmy od podzielenia się pracą i przygotowaniem wykresu Gantta. Podział pierwszych obowiązków prezentował się następująco:

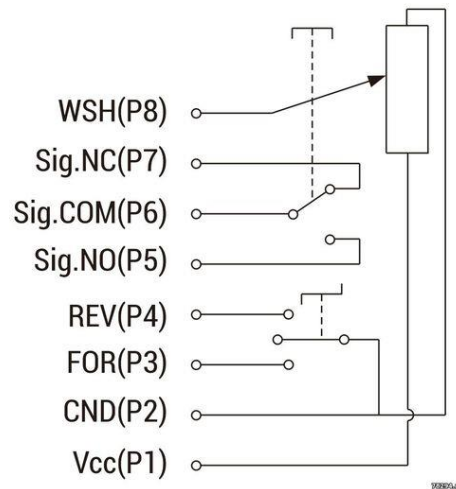
*Jakub Bajek – wykonanie aplikacji, odbiór danych na płytce i przekazanie ich w czytelnej formie do programu.*

*Zuzanna Baran – zaprogramowanie płytki nucleo, sterowanie przetwornikiem cyfrowo-analogowym, zgodnie z otrzymanymi danymi, oraz sterowanie wyjściami GPIO do zmiany kierunku oraz załączenia silnika.*

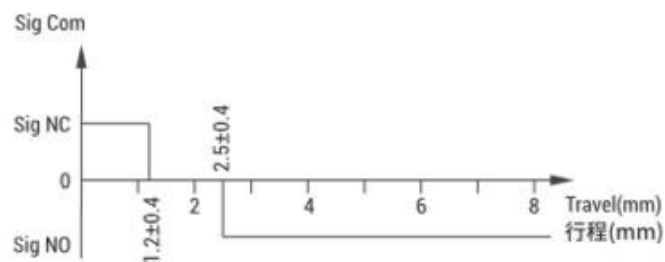
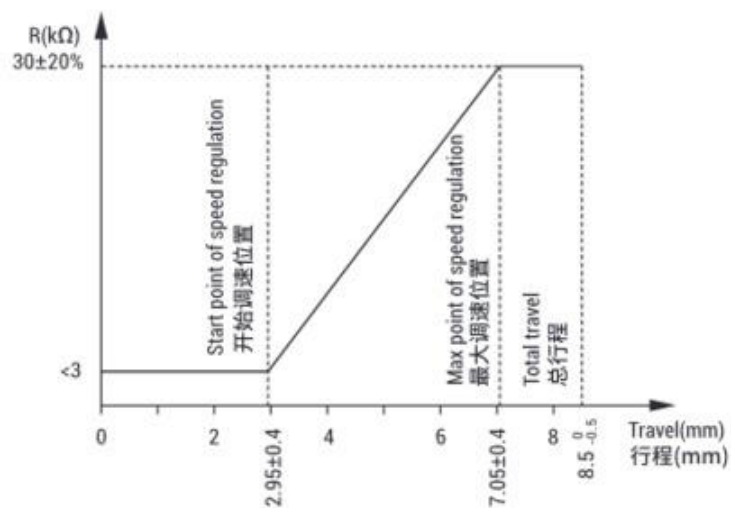
*Wiktor Kacprzak – przeprowadzenie reverse-engineering, research oraz analiza działania fizycznego przycisku, przygotowanie elektroniki niezbędnej do symulacji działania przycisku oraz połączenie układu.*

W czasie laboratorium nie trzymaliśmy się stricte wyznaczonych podczas podziału zadań. Wspólnie pomagaliśmy sobie w razie potrzeby w pomiarach przycisku, problemach z działaniem programu, czy finalnych testach i poprawkach. Podział obowiązków jednak pomógł nam zrównoleglić, a tym samym znacznie przyspieszyć pracę.

Podczas zajęć przeprowadziliśmy pełną analizę działania przycisku. Dużą rolę odegrała tutaj również znaleziona rosyjska dokumentacja, której poprawność potwierdziliśmy później podczas pomiarów.



Rys. 1. Pinout przycisku



78294.ru

Rys. 2. Rezystancja kanału w zależności od poziomu wciśnięcia

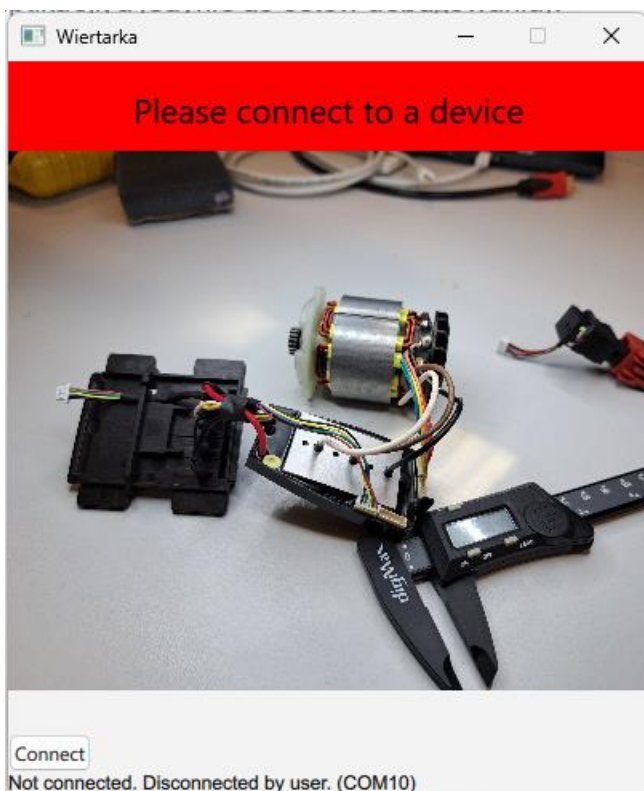
Ze strony aplikacji zaimplementowaliśmy komunikację z płytką za pomocą uarta. Dodaliśmy suwak od 0 do 100% z możliwością wpisania wartości wciśnięcia przycisku również bezpośrednio z klawiatury. Oprócz tego zaimplementowaliśmy 4 przyciski – *POWER ON*, *POWER OFF*, *DIR\_LEFT*, *DIR\_RIGHT* umożliwiające pełne sterowanie tak jak za pomocą fizycznego przycisku.

Komunikację zaprogramowaliśmy tylko jednostronnie (z aplikacji do płytki) z powodu ograniczonych zasobów czasowych. Długość ramki danych ustaliliśmy na 1 bajt ułatwiając tym samym implementację odbioru danych. Liczby od 0 do 100 zaprogramowaliśmy jako bezpośrednie ustalenie poziomu wciśnięcia przycisku w procentach, natomiast wolne możliwości jako:

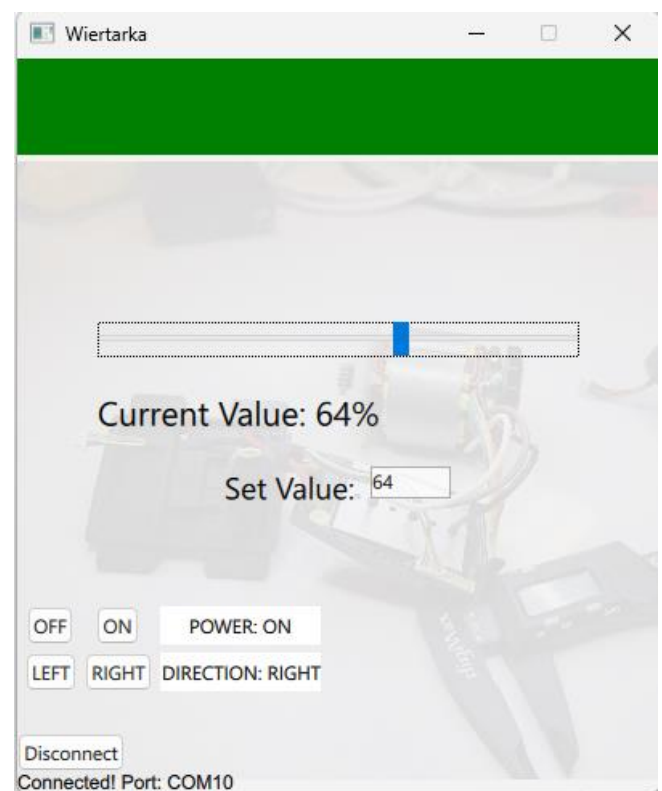
- 200 -> *power on*,
- 201 -> *power off*,
- 202 -> *direction right*,
- 203 -> *direction left*.

W celach debugowania program uwzględnia również:

- zapalanie diody w przypadku kliknięcia *power on* lub *direction\_right*
- zgaszenie diody w przypadku kliknięcia *power\_off* lub *direction\_left*
- echo na uarcie, zwracające bez zmiany i natychmiastowo otrzymany bajt (komunikacja zwrotna nie jest jednak wykorzystywana w aplikacji, a jedynie do celów debugowania).



Rys. 3. Interfejs aplikacji przed połączeniem



Rys. 4. Interfejs aplikacji po połączeniu

Zaprogramowaliśmy wystawianie napięcia na odpowiedni pin płytki z DAC zgodnie z otrzymaną procentową wartością, którą przeliczyliśmy na pełny zakres 12 bitowego przetwornika umożliwiając sterowanie w zakresie 0 – 3.3V.

Po przeanalizowaniu działania przycisku okazało się, że nie będziemy potrzebowali żadnego skomplikowanego modułu wykonawczego pomiędzy płytką, a sterownikiem silnika. Poziomy napięcie nie przekraczały tych, które mogliśmy bezpośrednioysterować za pomocą nucleo.

Mimo to potrzebowaliśmy specjalnie zaprojektowanej płytki PCB z wyprowadzonymi odpowiednimi gniazdami oraz pinami do wpięcia płytki oraz wyprowadzenia miejsc z łatwym dostępem do wykonywania pomiarów. Nie wymagane były natomiast na niej dodatkowe elementy jak rezystory czy tranzystory.

## ZDJĘCIE PŁYTKI!

Przeprowadziliśmy research mający na celu znalezienie oraz zakup odpowiednich gniazd oraz przewodów. Były one na tyle rzadkie, że ich zakup był możliwy tylko z zagranicy – z USA. Długi czas dostarczenia paczki spowolnił nasze prace.

Głównym problemem z naszym projektem mógł się okazać zbyt duży pobór prądu przez sterownik z pinów, którymi symulowaliśmy sygnał załączenia silnika oraz kierunek obrotów. Dostęp do wpięcia multimetru był jednak na tyle utrudniony, że założyliśmy iż prąd nie przekroczy limitów płytki nucleo.

Ostatecznie połączyliśmy wszystko w jedną całość i przeprowadziliśmy testy naszego rozwiązania.

**Wyniki:**

Zaprogramowany system sterowania w pełni spełnił nasze założenia, które postawiliśmy sobie podczas procesu inżynierii odwrotnej.

<działanie silnika>

**Podsumowanie:**

Praca nad projektem pozwoliła nam się zetknąć z fizycznie postawionym problemem, którego rozwiązanie spłynęło na nas bez żadnej instrukcji, która by nas prowadziła krok po kroku jak w przypadku wielu zajęć na studiach.

Pokazała nam jak ciężko czasem jest znaleźć odpowiednie, niezbędne komponenty czy nawet dokumentacje do niektórych elementów elektronicznych.

Nauczyliśmy się planowania, dotrzymywania terminów i wyznaczaniu kamieni milowych, a także poprawiliśmy swoje umiejętności pracy oraz komunikacji w grupie.