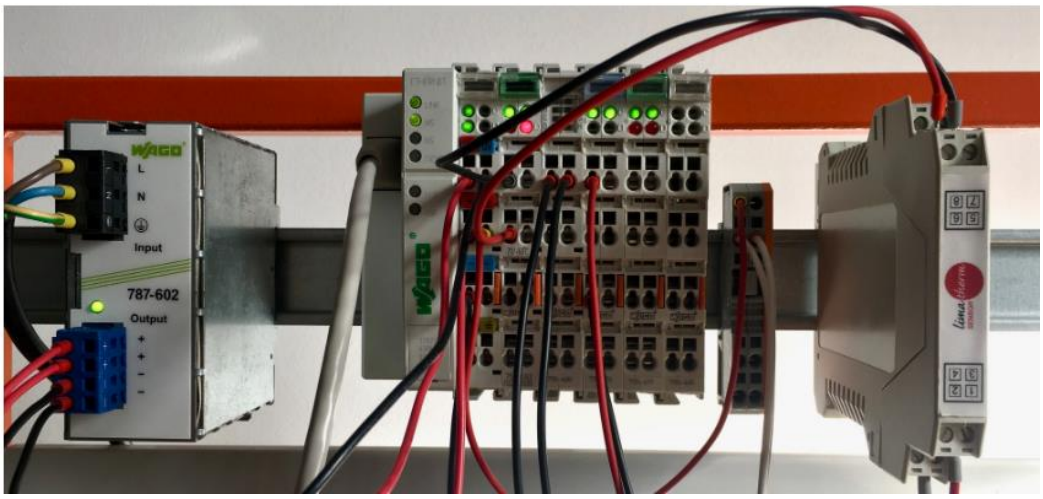
Protokół HART			
Julita Wójcik **Jakub Szczypek**	**25 V 2022**	**wtorek, 19:45**	**3A**

1. Cel ćwiczenia

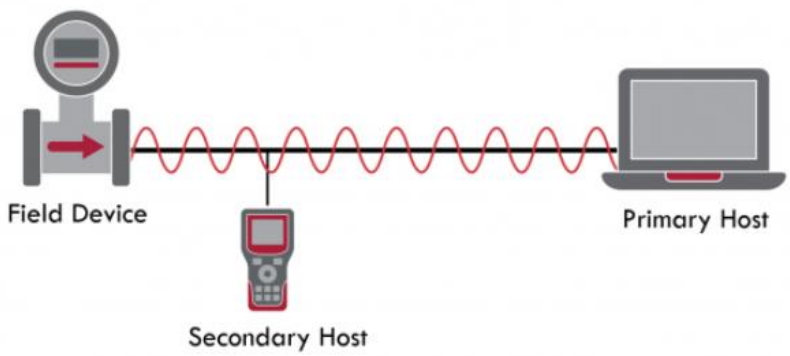
Celem ćwiczenia jest zapoznanie się z protokołem HART na przykładzie komunikacji przetwornika temperatury TxIsoRail-HART ze sterownikiem Wago 750-841, wyposażonym w dwukanałowy, analogowy moduł wejściowy 750-482/025-000. Elementem pomiarowym jest termopara typu K.



Rysunek 1. Stanowisko pracy

2. Wstęp teoretyczny

Protokół HART (*Highway Addressable Remote Transducer*) umożliwia dwukierunkową komunikację w trudnych warunkach środowiskowych. Jest to protokół typu request-response pomiędzy inteligentnym urządzeniem polowym a systemem nadrzędnym. Przesyłanie danych odbywa się po liniach transmisyjnych, wykorzystując w warstwie fizycznej pętlę prądową 4-20 mA. Dzięki temu możliwe jest parametryzowanie urządzeń w sieci, uruchamianie ich w wygodny sposób, a także odczytywanie wyników pomiarów i zapisywanie danych w ich pamięci. Urządzenia wykorzystujące protokół HART dzielą się na urządzenia nadrzędne (*host/master*) i podrzędne (*field/slave*). Do urządzeń nadrzędnych możemy zaliczyć m.in. sterowniki PLC, przenośne terminale ręczne, jak i stacje robocze, a jako *slave* pracować mogą: czujniki, nadajniki i rozmaite urządzenia wykonawcze.



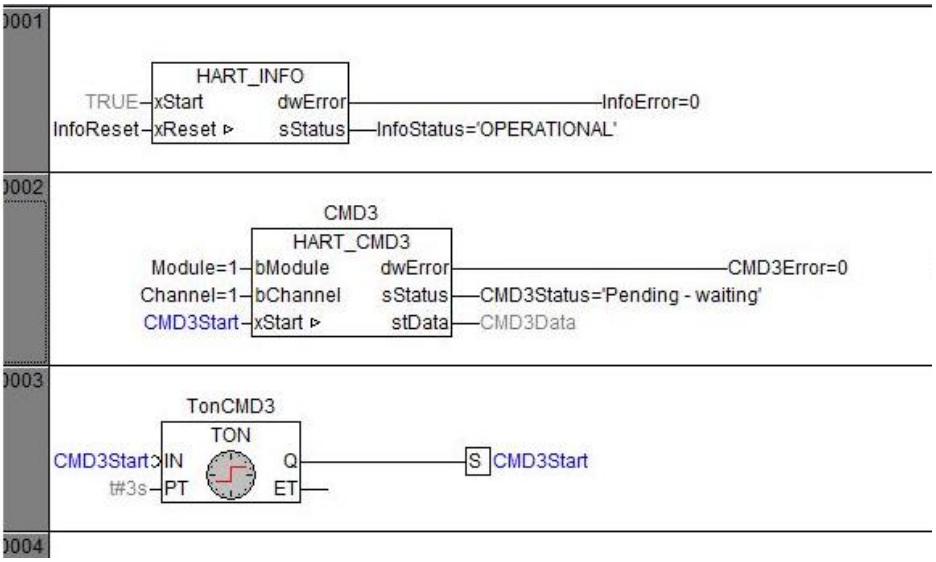
Rysunek 2. Schemat połączenia

3. Przebieg ćwiczenia

W środowisku CoDeDys utworzono nowy projekt, wybierając potrzebny sterownik WAGO_750-841_(FW12-...). Zgodnie z instrukcją określono język programowania - FBD, dodano moduły wejścia/wyjścia, skonfigurowano parametry komunikacji ze sterownikiem PLC oraz zainstalowano bibliotekę do obsługi komunikacji HART.

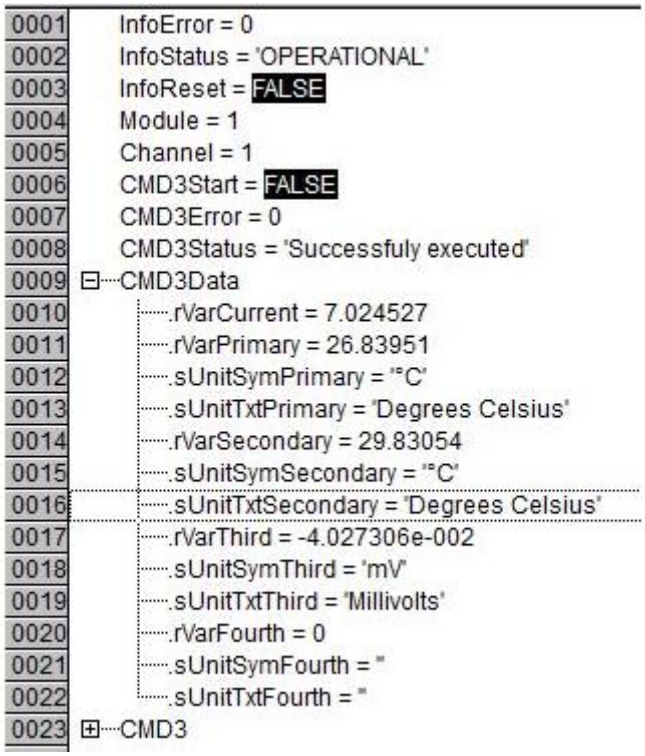
Po dokonaniu sprawdzenia poprawności konfiguracji PLC przystąpiono do napisania programu obsługującego moduł wejść analogowych HART. Pracę rozpoczęto od wstawienia do programu funkcji HART_INFO. Zmienne do wejść i wyjść bloczka można przypisać, wybierając odpowiednie miejsce na bloczku, bądź definiować je w górnej części okna, a później przypisywać do poszczególnych składowych programu.

Do odczytu zmiennych z modułu HART wykorzystano uniwersalną komendę HART nr 3 – w bibliotece WagoLibHART służy do tego blok funkcyjny HART_CMD3. Przypisano do niego wszystkie wejścia i wyjścia, pamiętając o odpowiednich typach. W celu poprawnego wywołania funkcji należało ustawić poprawną wartość Module oraz Channel. Blok HART_CMD3 jest wywoływany rosnącym zboczem, więc ręczne wywoływanie bloku za pomocą zmiany wartości CMD3Start zastąpiono timerem, który robił to cyklicznie co 3 sekundy. Na rysunku 3. przedstawiono kompletny program z funkcjonalnościami opisanymi wyżej.



Rysunek 3. Przygotowany program

Program umożliwił nam odczytanie wartości takich jak prąd, wartość procesowa, jednostka, opis jednostki. Odczytane wartości zmiennych znajdują się w strukturze CMD3Data. Odczyt przedstawiono na rysunku 4.



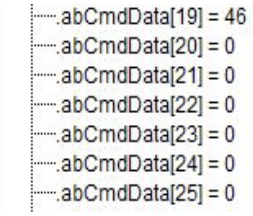
Rysunek 4 Odczytane wartości zmiennych

Dzięki sprawdzeniu pola CMD3.bCmdDataCount uzyskano liczbę bajtów odpowiedzi od urządzenia – 19. Porównując tę wartość z tabelą 1. W dokumentacji wywnioskowano, że przetwornik obsługuje trzy zmienne – PV, SV, TV.

Tabela 1 . Obsługiwane zmienne w zależności od liczby bajtów odpowiedzi.

Dynamic Variables Supported	No. of Response Data Bytes
PV	9
PV, SV	14
PV, SV, TV	19
PV, SV, TV, QV	24

Wszystkie pola w tablicy CMD.abCmdData powyżej 19 zostały wyzerowane, co przedstawiono na rysunku 5.



Rysunek 5. Wyzerowane pola tablicy

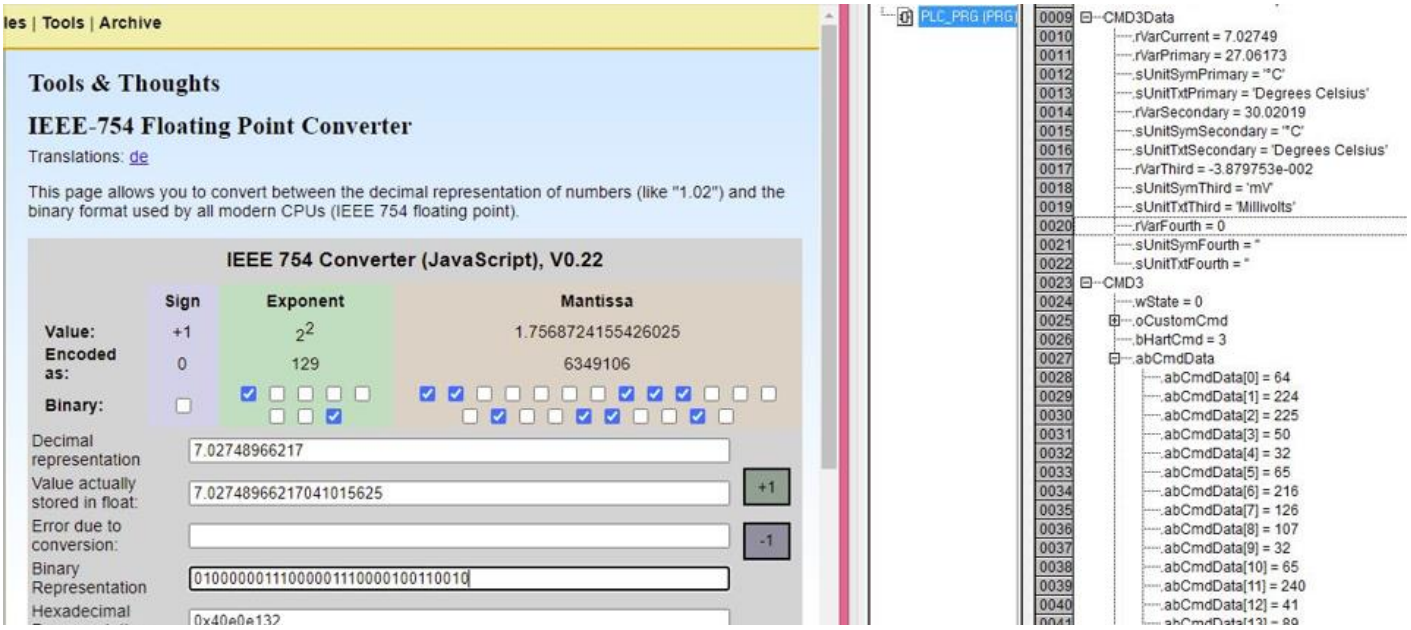
Na podstawie danych w tablicy CMD.abCmdData zdekodowano wartość PV, jednostkę PV oraz wartość prądu (Loop Current). Porównano zdekodowane wartości z danymi w strukturze CMD3DATA. Wykorzystano konwerter online, który pozwala na dokonanie konwersji liczby binarnej do typu float, zgodnie ze standardem IEEE-754.

Wykorzystując tabelę 2. z dokumentacji, sprawdzono co oznaczają poszczególne pola tablicy CMD.abCmdData

Tabela 2. Oznaczenia poszczególnych bitów

Byte	Format	Description
0-3	Float	Primary Variable Loop Current (units of milliamps^2)
4	Enum-8	Primary Variable Units Code (refer to *Common Tables Specification*)
5-8	Float	Primary Variable
9	Enum-8	Secondary Variable Units Code (refer to *Common Tables Specification*)
10-13	Float	Secondary Variable
14	Enum-8	Tertiary Variable Units Code (refer to *Common Tables Specification*)
15-18	Float	Tertiary Variable
19	Enum-8	Quaternary Variable Units Code (refer to *Common Tables Specification*)
20-23	Float	Quaternary Variable

Zdekodowano bity 0-3. Każdą z wartości decymalnych zamieniano na wartość binarną. Uzyskany ciąg wartości wpisano do konwertera, co przedstawiono na rysunku 6.



Rysunek 6. Zdekodowanie bitów 0-3

Decymalna reprezentacja wpisanej liczby binarnej pokrywa się z tą w polu CMD3Data, co świadczy o poprawności wykonania ćwiczenia oraz odpowiedniego działania urządzenia.

Zgodnie z Tabelą 3. Porównano wartości bitu nr 5 – 32, co odpowiada jednostce °C. Jednostka ta pokryła się z informacją z pola .sUnitSymPrimary w zakładce CMD3Data.

Tabela 3. Kody jednostek inżynierskich - temperatura

Unit Code	Symbol / Description	
32	°C	degrees Celsius
33	°F	degrees Fahrenheit
34	°R	degrees Rankine
35	K	kelvin

Ostatnim punktem zadania było zdekodowanie bitów 5-8. Rezultat przedstawiono na rysunku 7.

Rysunek 7. Zdekodowanie bitów 5-8

Ponownie, uzyskana wartość pokryła się z odpowiedzią urządzenia.

4. Wnioski

Ćwiczenie pozwoliło na zapoznanie się z protokołem HART - standardem komunikacji urządzeń pomiarowych w przemyśle. Wykonanie laboratorium usprawniło tworzenie prostych programów w języku blokowym FBD oraz zwiększyło świadomość wyszukiwania potrzebnych informacji w dokumentacji od producentów