Sprawozdanie z laboratorium 6

HART (WAGO)

Łukasz Janusz
Marek Generowicz
09.03.2025

AGH UNIVERSITY OF KRAKOW

1 Wstęp

Na laboratoriach należało zapoznać się z protokołem HART, zasadami komunikacji oraz praktycznymi aspektami wykorzystania go w przemyśle. W trakcie zajęć przeprowadzono ćwiczenia z wykorzystaniem sterownika *WAGO 750-841* wyposażonym w dwukanałowy analogowy moduł wejścia, który pozwala na komunikację z urządzeniami HART. Elementem pomiarowym natomiast jest *termopara typu K*, która została połączona z modułem WAGO za pomocą przetwornika temperatury *TxIsoRail-HART*.

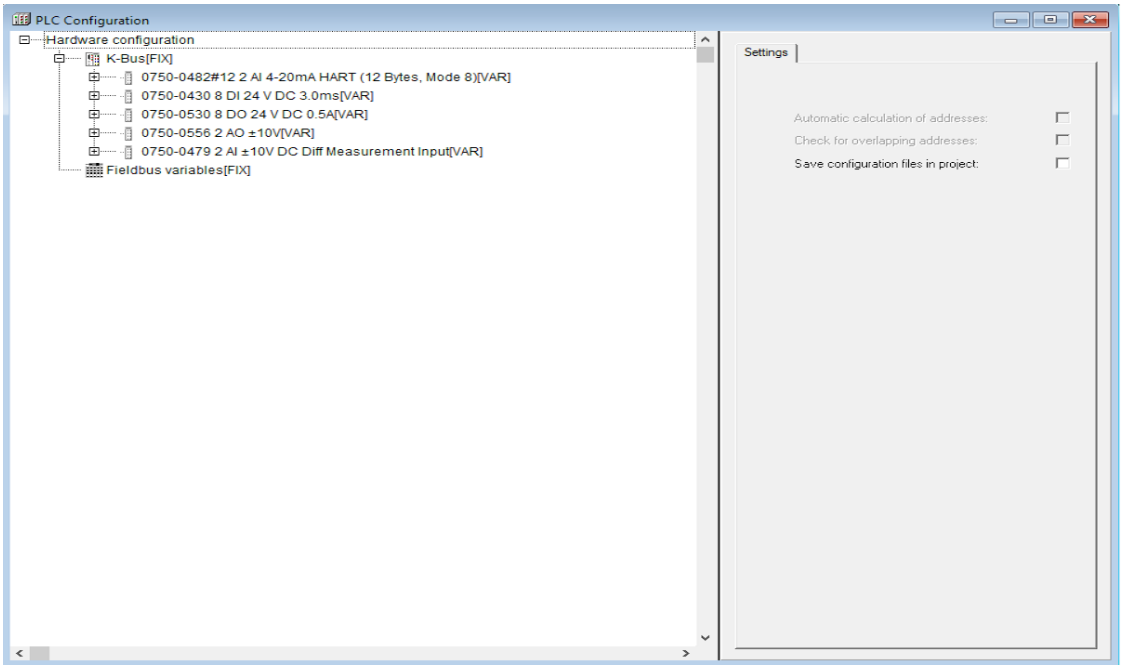
1.1 Protokół HART

Protokół HART *(High Addressable Remote Transducer)* jest standardem komunikacyjnym stosowanym w przemyśle, który pozwala na komunikację z urządzeniami pomiarowymi, takimi jak czujniki, przetworniki, zawory, itp. Protokół HART umożliwia przesyłanie danych cyfrowych i analogowych w jednym przewodzie. Komunikacja odbywa się za pomocą sygnałów modulowanych na sygnale prądu stałego, co pozwala na przesyłanie danych cyfrowych wraz z sygnałem analogowym. Protokół HART jest kompatybilny z większością urządzeń pomiarowych, co pozwala na łatwe wdrożenie w istniejących systemach. Urządzenia, które wykorzystują protokół HART, są podzielone na nadrzędne (np. sterowniki PLC) i podrzędne (np. czujniki).

2 Przebieg ćwiczenia

2.1 Konfiguracja PLC

W pierwszej części zadania należało zaprogramować sterownik *WAGO*. W tym celu należało skorzystać z aplikacji *CoDeSys*. Ważne aby w nowo stworzonym projekcie ustawić *Type od POU* na *Program* a język programowania na *FBD* ze względy na konieczność wykorzystania biblioteki do obsługi *POU* napisanej w tym właśnie języku. Następnie należało dodać moduły wejścia i wyjścia w wirtualnym wnętrzu magistrali. Uzupełniona magistrala wyglądała jak na zdjęcie 1.

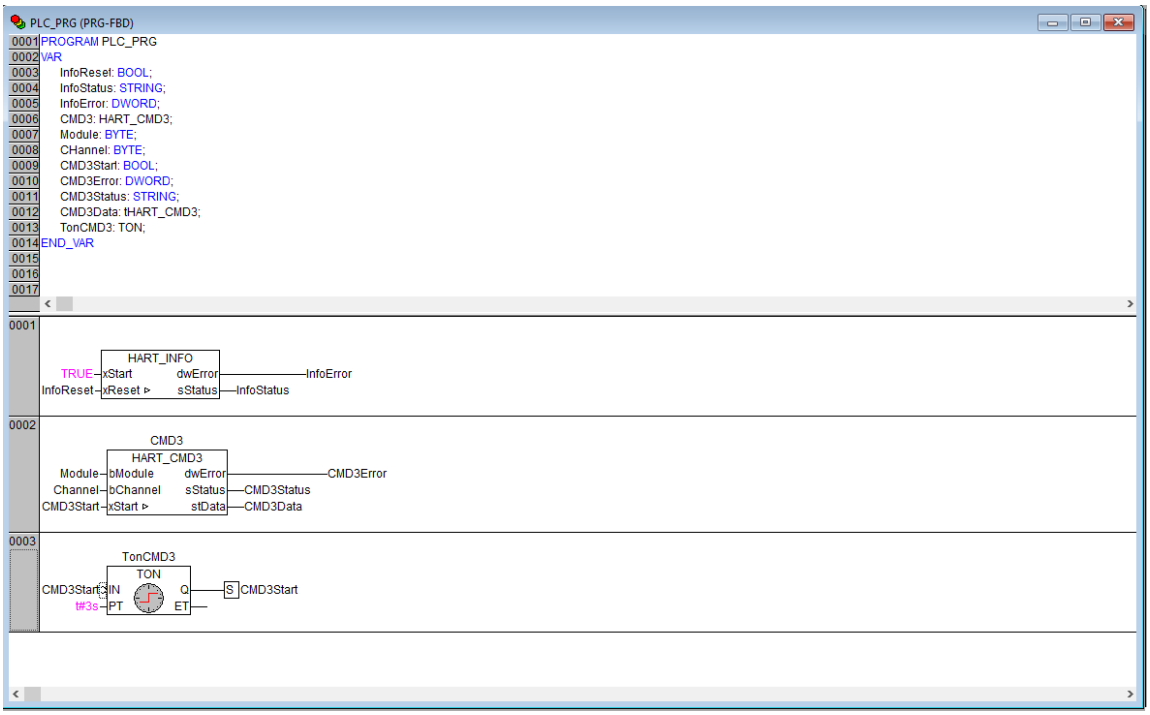


Rysunek 1: Wnętrze magistrali w aplikacji *CoDeSys*.

Przed przystąpieniem do programowania należało skonfigurować parametry komunikacji oraz, w razie gdyby jej nie było, dodać bibliotekę do obsługi komunikacji HART *WagoLibHART_03.lib*

2.2 Program - HART_INFO

Po skonfigurowaniu PLC należało przejść do napisania kodu obsługującego moduł HART. Okno programu jest podzielone na dwie części, w górnej znajdują się zmienne używane w kodzie, a w dolnej znajduje się logika kodu. Zdjęcie 2 przedstawia całość kodu napisanej na zajęciach aplikacji. Jak widać logika zależy od trzech bloków logicznych.



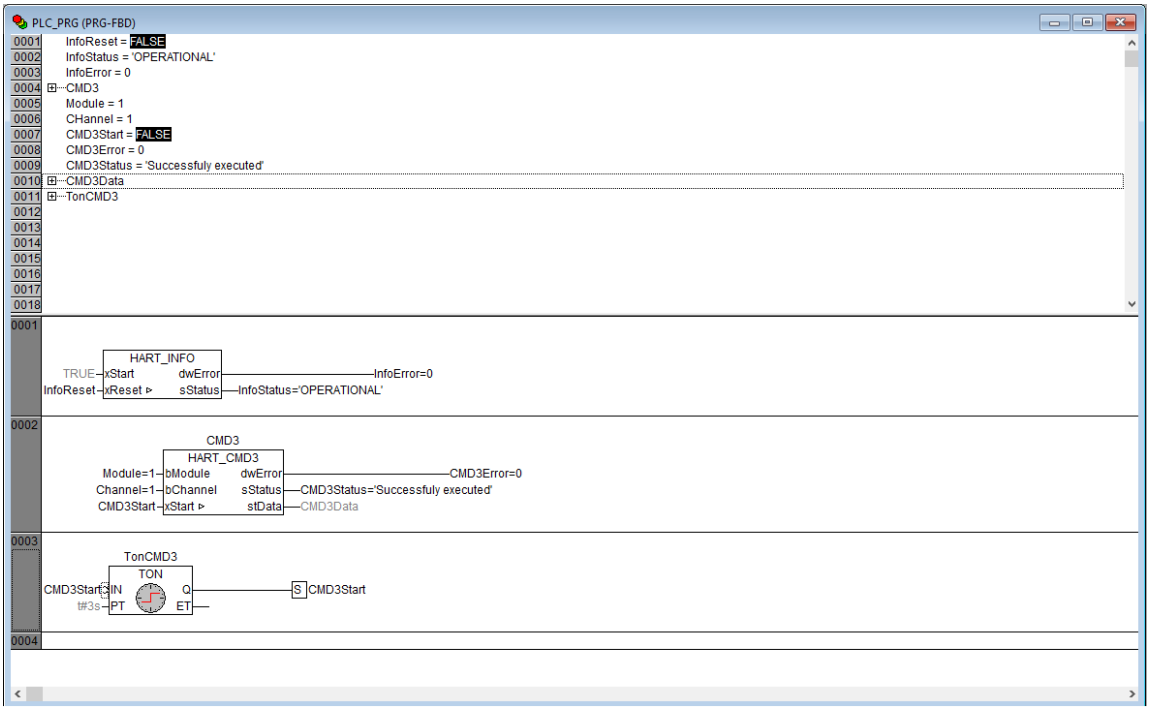
Rysunek 2: Program obsługujący wejście analogowe modułu HART.

Pierwszy z nich *HART_INFO* służy do odczytu czy cały układ jest w stanie komunikować się z modułem HART.

Drugi blok *HART_CMD3* za pomocą uniwersalnej komendy HART numer 3 służy do odczytu zmiennych w module HART.

Trzeci blok *TON* jest zegarem, który pozwala na cykliczne wykonywanie logiki drugiego bloku, ponieważ blok ten jest uruchamiany rosnącym zboczem parametru *CMD3Start*. Ważne jest aby wejście timera zanegować aby zegar działał zgodnie z tym co oczekujemy.

Stworzenie takiego programu daje nam możliwość odczytu zmiennych z modułu HART, co jest niezbędne do dalszych działań. W trakcie uruchomiania programu wartości należało pamiętać aby ustawić wartości zmiennych *Module* oraz *Channel* na wartości 1. Wygląd uruchomionego programu przedstawia zdjęcie 3.

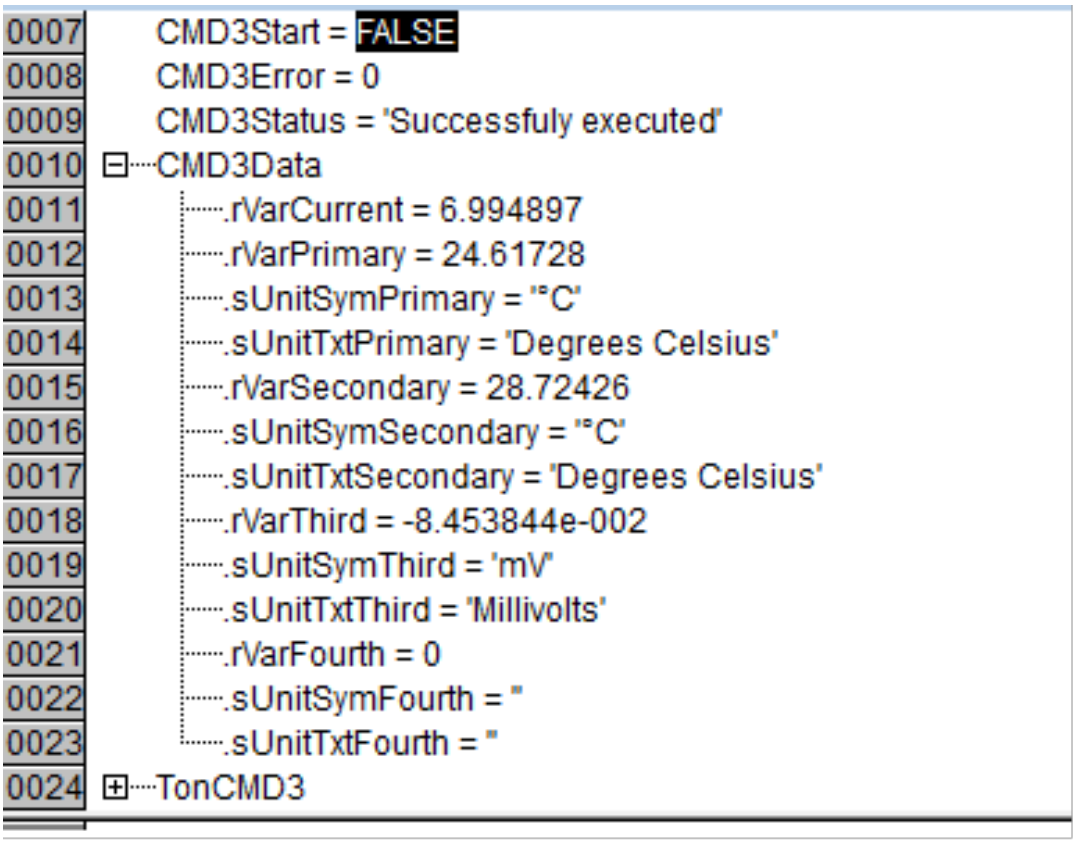


Rysunek 3: Uruchomiony program do obsługi modułu HART.

2.3 Program - odczyt zmiennych HART

Po poprawnej konfiguracji oraz zaprogramowaniu sterownika, można przystąpić do odczytu zmiennych z modułu HART. W tym celu należy uruchomić program i ustawić poprawne wartości zmiennych *Module* oraz *Channel* na wartości 1. Po uruchomieniu programu, dzięki blokowi *TON* program będzie cyklicznie co 6 sekund aktualizował dane, które moduł HART będzie otrzymywał od termopary typu K.

Aby odczytać wartości odebrane przez termoparę typu K oraz przekazane przez moduł HART, należy w części programu gdzie znajdują się parametry układu rozwinąć część *CMD3Data*. Zdjęcie 4 przedstawia jak wygląda odczyt zmiennych z modułu HART.



Rysunek 4: Dane otrzymane z modułu HART.

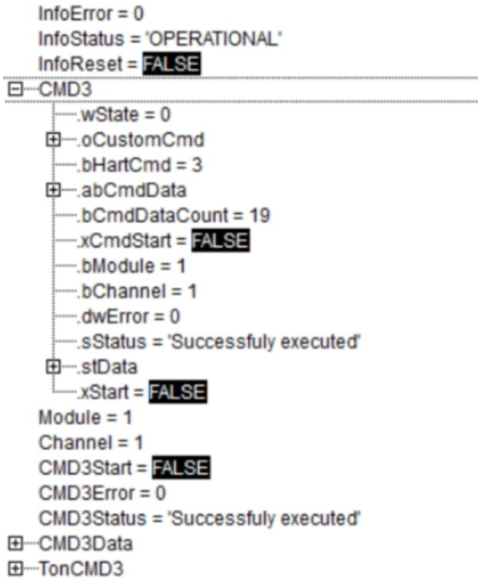
2.4 Dane HART

W celu sprawdzenia ilości zmiennych dynamicznych należ skorzystać z tabeli dostępnej w dokumentacji urządzenia.

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

Tabela 1: Zależność między ilością zmiennych dynamicznych a długością tablicy CMD.abCmdData

Aby tabela była użyteczna sprawdzamy wartość pola CMD3.bCmdDataCount, która w naszym przypadku wynosi 19.



Rysunek 5: Wartość pola CMD3.bCmdDataCount

Dzięki temu wiadomo, że nasze urządzenie obsługuje dokładnie 3 zmienne dynamiczne.

Następnie należy sprawdzić jakie informacje są przekazywane w poszczególnych bitach oraz zdekodować i porównać znalezione wartości do danych widniejących w strukturze CMD3DATA. W celu sprawdzenia informacji o poszczególnych polach należy skorzystać z dokumentacji urządzenia.

Request Data Bytes

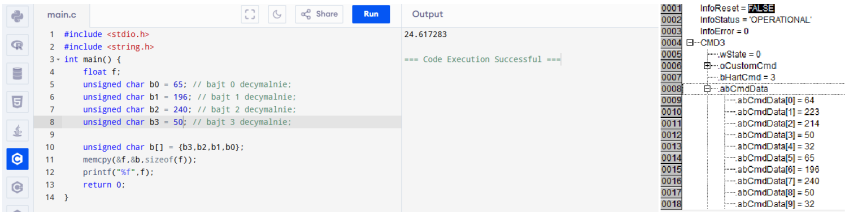
Byte	Format	Description
None		

Response Data Bytes

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

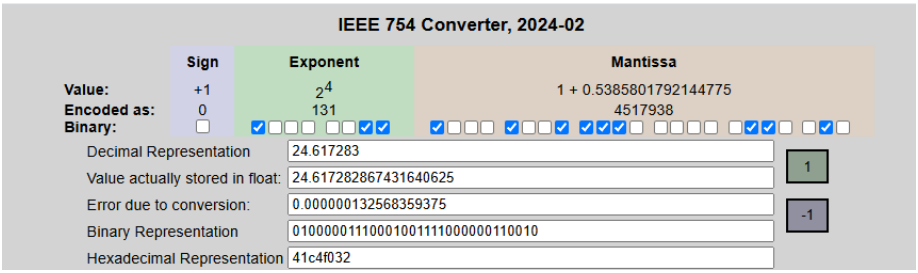
Tabela 2: Znaczenie poszczególnych bitów w tabeli abCmdData

W celu sprawdzenia wartości *PV (Primary Value)* został użyty prosty program w języku C.



Rysunek 6: Program w języku C do sprawdzenia wartości PV

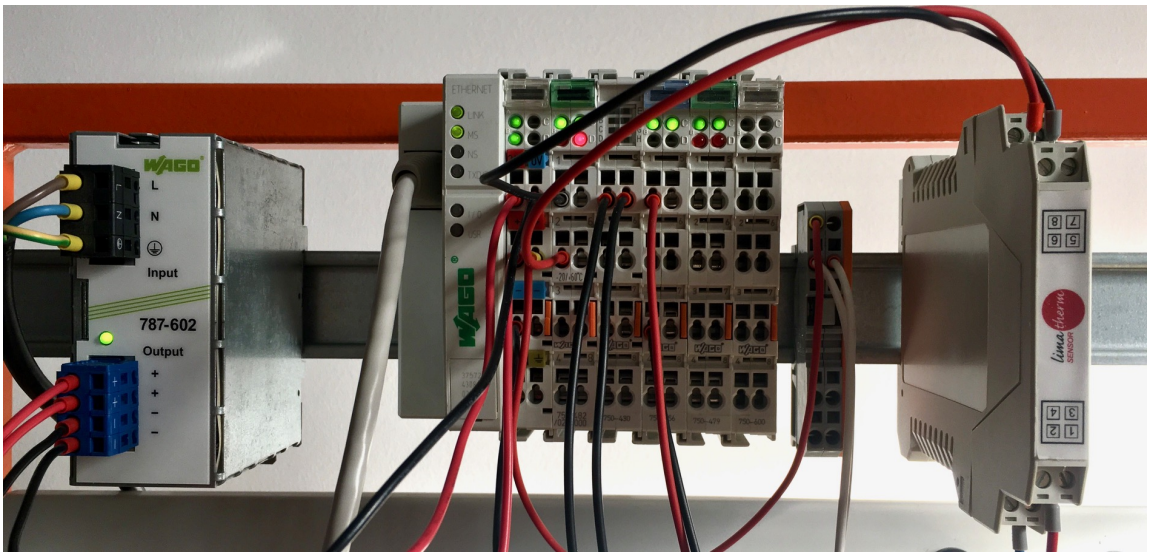
Jak widać, jego odpowiedź jest zgodna (co do ilości wyświetlanych miejsc po przecinku) z danymi wyświetlanymi w polu CMD3Data.rVarPrimary (Rysunek 4).
Drugim sposobem sprawdzenia wartości *PV* jest internetowy konwerter dostępny pod linkiem https://www.h-schmidt.net/FloatConverter/IEEE754.html. Po wprowadzeniu odczytanych wartości otrzymujemy identyczny wynik.



Rysunek 7: Konwersja wartości binarnych na wartość PV

3 Podsumowanie

Nauka programowania modułu HART jest bardzo przydatna ponieważ pozwala na komunikację z urządzeniami pomiarowymi w przemyśle ponieważ daje możliwość komunikacji dwukierunkowej nawet w niesprzyjających warunkach. W trakcie laboratoriów udało nam się zaprogramować sterownik *WAGO 750-841*, odczytać dane z modułu HART oraz nauczyć się odczytywać "surowe"dane aby odczytać poszukiwane informacje. Dzięki temu zdobyliśmy praktyczną wiedzę na temat protokołu HART co umożliwi nam w przyszłości praktyczne wykorzystanie tej wiedzy w przemyśle.



Rysunek 8: Wygląd modułu HART