Fizyka i przetwarzanie danych w systemach wbudowanych

Dr Michał Tanaś (http://mtanas.home.amu.edu.pl/)

- CoDeSys (Controller Development System)
 - Środowisko developerskie dla systemów wbudowanych
 - Tworzone od 1994 przez Codesys GMBH w Bawarii
 - Niezależne od producentów PLC (wspiera większość dostępnych, a nie konkretną markę)
 - Licencje
 - Środowisko developerskie darmowe
 - Symulator PLC (SoftPLC) płatne
 - Do pobrania z <u>https://www.codesys.com/download.html</u> (wymagana rejestracja na e-maila)

- Projekt zbiór obiektów stanowiących kompletne oprogramowanie sterownika
- Wzorowany na projektach w środowiskach developerskich Visual C++

| Funkcja | CoDeSys | Visual C++ |
|---------------------------|-----------------------|-----------------|
| Spis plików | Projekt | Projekt |
| Kod źródłowy (definicje) | Moduł | Plik źródłowy |
| Kod źródłowy (deklaracje) | Definicja typu danych | Plik nagłówkowy |
| Wizualizacja | Wizualizacja | Wizualizacja |
| Punkt startowy programu | Funkcja main() | Moduł PLC_PRG |

- Tryb pracy sterownika
 - Polling
 - Sterowanie przekazywane jest do modułu PLC_PRG
 - Moduł PLC_PRG wywołuje pozostałe moduły i funkcje sterownika wg. własnego uznania
 - PLC_PRG powinien działać w nieskończonej pętli
 - Wyjście z PLC_PRG oznacza zakończenie działania sterownika (analogicznie do funkcji main() w C / C++)
 - Odpowiednik kompilacji C++ do programu wykonywalnego

- Tryb pracy sterownika
 - Event driven
 - Sterownik składa się z niezależnych od siebie modułów
 - Każdy z modułów jest wykonywany gdy
 - Zajdzie odpowiednie zdarzenie
 - Spełnione są odpowiednie warunki startu
 - Brak modułu głównego PLC_PRG
 - Odpowiednik kompilacji C++ do biblioteki dynamicznej

• Tworzenie projektu

- Wybrać system docelowy
- Skonfigurować sterownik (wejścia i wyjścia)
- Stworzyć moduły (w ramach jednego projektu można tworzyć moduły w różnych językach)
- Skompilować projekt

Testowanie projektu

- Uruchomić symulator PLC
- Załadować skompilowany projekt do symulowanego sterownika
- Otworzyć okno podglądu konfiguracji sterownika
- Ustawić ręcznie wartości na wejściach
- Sprawdzić czy wartości na wyjściach są prawidłowe
- Można ustawiać breakpointy
- Można wykonywać program krok po kroku
- Można wyświetlić przebieg czasowy wejść i wyjść

- Engineering interface (ENI)
 - Repozytorium modułów CoDeSys (z wersjonowaniem)
 - Może współpracować z wieloma środowiskami developerskimi na raz
 - Przeznaczony do tworzenia projektów przez większe zespoły
- Funkcje związane z dokumentacją
 - Automatyczne generowanie dokumentacji projektu
 - Eksport projektu do plików tekstowych

- Projekt
 - Zawiera wszystkie obiekty programu sterownika
 - Jest przechowywany w pliku o nazwie takiej jak projekt
 - Zawiera m.in.:
 - Moduły
 - Typy danych
 - Wizualizacje
 - Zasoby
 - Biblioteki

- Moduł
 - Fragment oprogramowania sterownika realizujący określoną funkcjonalność
 - Składa się z części deklaratywnej i z kodu
 - Typy modułów:
 - Funkcje
 - Bloki funkcji
 - Programy
 - Moduł może wywołać inny moduł
 - ... ale bez rekursji

- Funkcja
 - Moduł, który zwraca dokładnie jedną wartość
 - ... przy czym może zwracać wartość typu złożonego (np. strukturę)
 - Deklaracja funkcji:
 - FUNCTION nazwa_funkcji : typ_wyniku

• Wywołanie funkcji

• ST

- wynik:=nazwa_funkcji(1,2,3)
- IL
 - LD 1
 - nazwa_funkcji 2,3
 - ST wynik



- Blok funkcji
 - Moduł, który zwraca jedną lub więcej wartości
 - ... w sposób podobny do zwracania wartości przez wskaźnik lub przez referencję w C/C++
 - Żeby wywołać blok funkcji należy utworzyć jego instancję
 - Pojedynczy blok funkcji może mieć dowolną liczbę instancji.
 - Uwaga! Instancja przechowuje wartości zmiennych, więc kolejne wywołania z tymi samymi parametrami mogą dać różne wyniki

| CoDeSys | C++ |
|--------------|--------|
| Blok funkcji | Klasa |
| Instancja | Obiekt |

 Blok funkcji FUNCTION BLOCK **BLOKFUN** VAR INPUT PAR1:INT; PAR2:INT; END_VAR VAR OUTPUT WYN1:INT; WYN2:INT; END VAR

Instancja
INSTANCJA: BLOKFUN;

Wywołanie bloku funkcji

• ST

INSTANCJA(PAR1:=1,PAR2:=2) WYNIK1:=INSTANCJA.WYN1; WYNIK2:=INSTANCJA.WYN2;

• FBD



• Program

- Blok funkcji, który
 - Jest globalny w całym projekcie
 - Nie posiada instancji
 - ... ale podobnie jak instancja przechowuje wartości zmiennych

| CoDeSys | C++ |
|---------|-----------------|
| Program | Klasa statyczna |

• Program **PROGRAM PROG** INPUT VAR PAR1:INT; PAR1:INT; END VAR OUTPUT VAR WYN1:INT; WYN1:INT; END VAR

- Wywołanie programu
 - ST

PROG(PAR1:=1,PAR2:=2) WYNIK1:=PROG.WYN1; WYNIK2:=PROG.WYN2;





- Akcja
 - Blok funkcji lub program, który jest składową innego bloku funkcji lub programu
 - Akcja ma dostęp do zmiennych bloku/programu nadrzędnego
 - Żeby stworzyć akcję należy:
 - Stworzyć standardowy pogram lub blok funkcji
 - Ustawić go jako akcję przy pomocy "Projekt / Dodaj Akcję"

| CoDeSys | C++ |
|---------|--------|
| Akcja | Metoda |

- Debugowanie
 - Breakpoint
 - Punkt wstrzymania wykonania programu
 - Umieszczany
 - W językach tekstowych w wierszu kodu źródłowego
 - W językach graficznych w punkcie schematu
 - Dojście do breakpoint'a powoduje wstrzymanie wykonania programu
 - ... przy zachowaniu dotychczas obliczonych wartości zmiennych
 - ... co daje możliwość obserwacji wartości zmiennych w trakcie wykonania
 - Można kontynuować wykonanie programu

- Debugowanie
 - Breakpoint
 - Punkt wstrzymania wykonania programu
 - Umieszczany
 - W językach tekstowych w wierszu kodu źródłowego
 - W językach graficznych w wierzchołku schematu
 - Dojście do breakpoint'a powoduje wstrzymanie wykonania programu
 - ... przy zachowaniu dotychczas obliczonych wartości zmiennych
 - ... co daje możliwość obserwacji wartości zmiennych w trakcie wykonania
 - Można kontynuować wykonanie programu

- Debugowanie
 - Praca krokowa
 - Umożliwia wykonanie programu krok po kroku
 - Gdzie krok oznacza:
 - W językach tekstowych pojedynczy wiersz kodu źródłowego
 - W językach graficznych pojedynczy wierzchołek schematu
 - Odpowiada wstawieniu breakpointa w każdy wiersz (lub w każdy wierzchołek)
 - Można kontynuować wykonanie programu

• Debugowanie

- Pojedynczy cykl
 - Wykonanie pojedynczego obiegu pętli sterującej
 - Odpowiada ustawieniu breakpointu po ostatnim wierszu kodu źródłowego lub po ostatnim wierzchołku schematu
- Monitoring
 - Normalne wykonanie programu sterownika
 - ... ale z wyświetlaniem wartości zmiennych
 - Możliwa jest również "ręczna" zmiana wartości zmiennych
- Dziennik
 - Zapis zdarzeń w trakcie wykonywania (np. wywołań funkcji) oraz interakcji użytkownika