A. Informacje o zespole realizującym ćwiczenie

Nazwa przedmiotu:	Automatyka pojazdowa
Nazwa ćwiczenia:	Architektura systemów sterowania w samochodach
Data ćwiczenia:	2022-03-23
Czas ćwiczenia:	15:00– 16:30
Zespół realizujący ćwiczenie:	• Błażej Szczur • Jakub Szczypek • Julita Wójcik

B. Sformułowanie problemu

Celem zajęć jest budowa i symulacja działania prostej architektury systemowej zawierającej co najmniej dwa komunikujące się ze sobą moduły. Architekturę należy zbudować i przeprowadzić symulację działania w środowisku CANoe. Jeden z dwóch stworzonych węzłów powinien co 100ms zwiększać zadany sygnał x o wartość 0.5 i wysyłać na magistralę w wiadomości, która będzie odbierana przez drugi z węzłów. Jego zadaniem jest odczytywanie sygnału x oraz wyliczanie sygnału y na podstawie danej formuły $y = \sin(x)\cos(\frac{x}{20})$. Wartości te należy wysłać na magistralę. Poprawność zadania można sprawdzić obserwując przesyłane wiadomości sygnały oraz graficzne wyrysowanie sygnału y. Dopełnieniem zadania jest zdefiniowanie panelu z 4 elementami: LED 'em, SWITCH 'em, wskaźnikiem oraz wskaźnikiem cyfrowym. Kolor diody LED powinien sygnalizować stan węzła A: aktywny – kolor czerwony nieaktywny – kolor szary. Wskaźnik ma reprezentować obliczoną wartość sygnału y, a wskaźnik cyfrowy x .

C. Sposób rozwiązania problemu

Wykonanie zadania podzielono na 3 etapy, które pozwoliły na kontrolowanie postępu, naprawianie błędów: wysyłanie danych w węźle pierwszym, odczyt i wysyłanie danych w węźle drugim oraz odczyt danych na stworzonym panelu.

Zdefiniowano dwie wiadomości - Msg1 oraz Msg2. Do wiadomości Msg1 przypisano sygnał x, a do Msg2 – sygnały x i y. Ustawiono odpowiednie typ – zmiennoprzecinkowy. Stworzono system zmiennych, w którym uwzględniono: $x, y, svSwitch$.

W węźle pierwszym – A, wykorzystując podany w instrukcji szkielet programu okresowo wywołujący funkcję, zdefiniowano timer, który co 100 ms tworzył wiadomość ‚typu' Msg1, zwiększał wartość sygnału x oraz wysyłał jego wartość na magistralę (etap zakończono obserwacjami poprawności danych na magistrali).

Drugi z węzłów zaprogramowano tak, aby odbierał wiadomość z węzła A, tworzył wiadomość ‚typu' Msg2. Dane z odebranej wiadomości odczytano za pomocą słowa kluczowego $this$. Wprowadzono formułę obliczającą wartość y. Przypisano wartości x, y do stworzonej wcześniej wiadomości. Dane wysłano na magistralę – ponownie przetestowano działanie tej funkcjonalność.

Do każdego elementu tworzonego panelu przypisano zmienną: do LED 'a i SWITCHA 'a – zmienną svSwitch, wskaźnika – y , wskaźnika cyfrowego – x . Dodano odpowiedni fragment kodu ‚aktywujący' działanie funkcji w węźle pierwszym, gdy SWITCH będzie aktywny (czyli svSwitch = 1).

```
7  variables
8  {
9    msTimer tm1;
10 }
11
12 on start {
13   setTimerCyclic(tm1, 100);
14 }
15
16 on timer tm1 {
17   if(sysGetVariableInt(sysvar::Var::svSwitch) == 1 )   {
18
19     float signal_x = 1;
20     message Msg1 msg1;
21     signal_x += 0.5;
22     msg1.x = signal_x;
23     output(msg1);
```

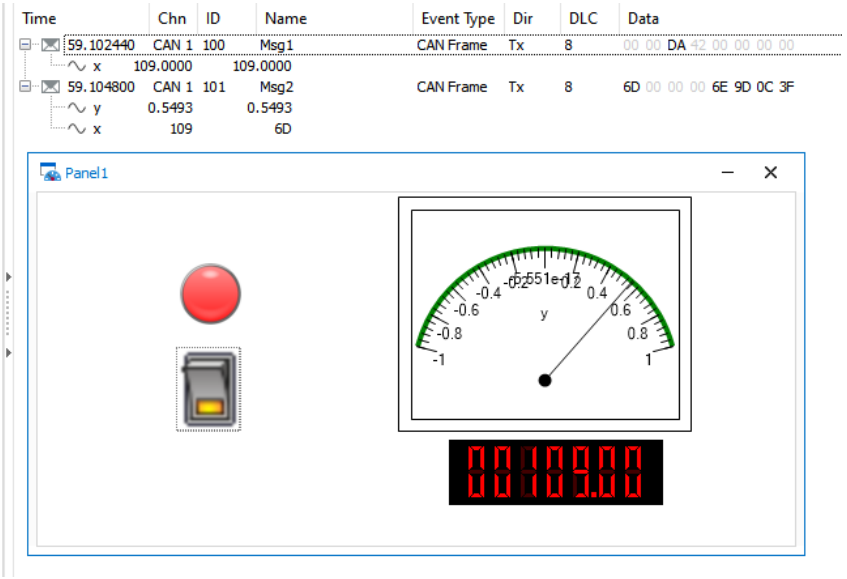
Rys1. kluczowy fragment kodu węzła nr 1

```
12 on message Msg1
13 {
14   message Msg2 msg2;
15   msg2.y = sin(this.x)*cos(this.x/20);
16   msg2.x = this.x;
17   output(msg2);
18 }
```

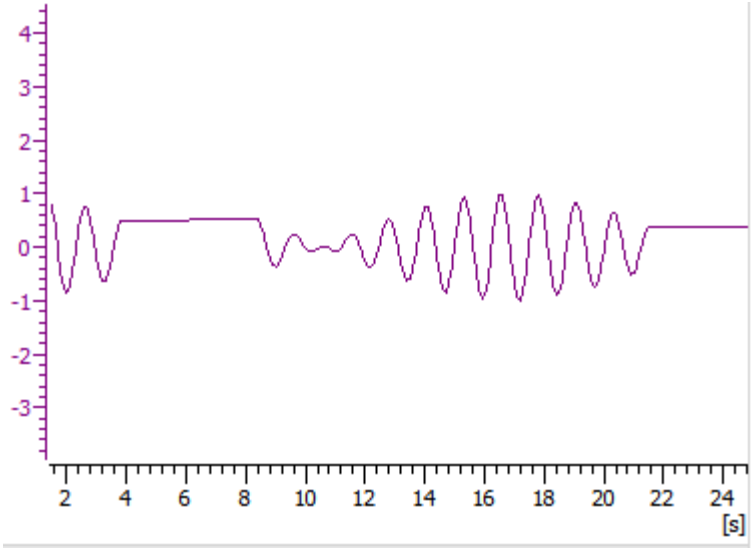
Rys2. kluczowy fragment kodu węzła nr 2

D. Wyniki

Podczas laboratorium poprawnie zbudowano wszystkie elementy sieci opisane w instrukcji.



Rys3. - Uruchomienie symulacji – SWITCH aktywny



Rys4. – wykres przebiegu wartości y

Na wykresie możliwe jest dostrzeżenie momentu zatrzymanie symulacji – od 4. do 8. sekundy oraz po ok. 22. sekundzie. Poza tym, widzimy oscylacje w zakresie od -1 do 1, co świadczy o poprawności wykonania zadania.

E. Wnioski

- Zaprojektowanie działania prostej architektury systemowej zawierającej najmniej dwa komunikujące się ze sobą moduły pozwoliło nam na utrwalenie podstawowych operacji w programie CANoe, zwiększyło pewność i biegłość działania w tym środowisku oraz zbudowało większą świadomość wykonywanych działań.
- Wykonując zadanie, powtórzyliśmy poznane już elementy składni języka CAPL, zwłaszcza operowanie z timerami i wywoływanie określonych funkcji w danym czasie.
- Uświadomiliśmy sobie, jak ważny jest podział zadanego problemu na mniejszy, w celu kontrolowania postępu i usuwania pojawiających się błędów, które wtedy stają się łatwiejsze do naprawy.