

A. Informacje o zespole realizującym ćwiczenie

Nazwa przedmiotu: Automatyka pojazdowa	
Nazwa ćwiczenia:	Architektura systemów sterowania w samochodach
Data ćwiczenia:	2019-03-06
Czas ćwiczenia:	09:30 – 11:00
Zespół realizujący ćwiczenie:	<ul style="list-style-type: none">• Sonia Wittek• Anna Gęca• Barbara Kaczorowska



B. Sformułowanie problemu

Celem laboratorium było stworzenie prostej architektury składającej się z dwóch komunikujących się ze sobą modułów. Węzły te powinny co 10 ms sprawdzać, czy jest między nimi połączenie i wysyłać potwierdzenie. Dodatkowo trzeba było zdefiniować wiadomość zawierającą sygnały x i y . Węzeł A co 100 ms powinien zwiększać wartość sygnału x i wysyłać jego wartość do węzła B, gdzie sygnał y wyliczany był z poniższego wzoru, a następnie odsyłany do węzła B.

$$y = \sin(x)\cos(x/20)$$

Należało również przedstawić wartości y w formie wykresu.

C. Sposób rozwiązania problemu

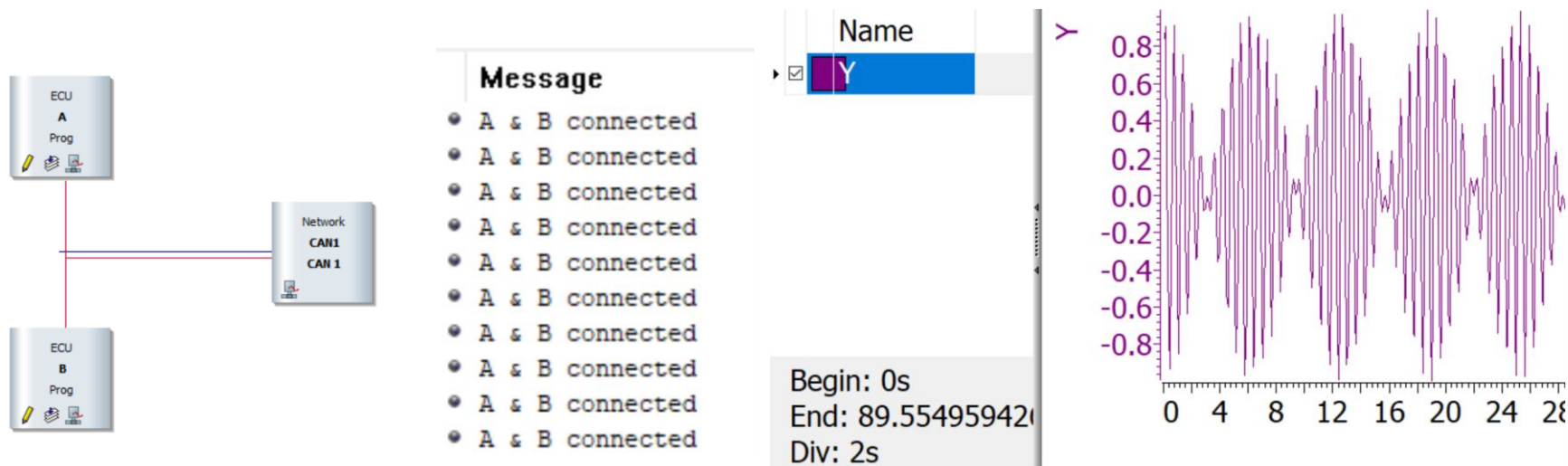
Ćwiczenie zostało przeprowadzone w programie CANoe, gdzie zasymulowana została magistrala CAN, a funkcjonalność poszczególnych węzłów została opisana przy pomocy języka CAPL.

```
includes
{
variables
{
    float x;
    mstimer connectionTimer;
    msTimer messageTimer;
}
on start
{
    setTimer(connectionTimer, 10);
    setTimer(messageTimer, 100);
    x = 0;
}
on message Connection_Message
{
    if(this.BIsConnected) write ("A & B connected");
}
on timer connectionTimer
{
    message Connection_Message A_connected;
    A_connected.AIsConnected = 1;
    output(A_connected);
}
on timer messageTimer{
    message Signal_Message_X msg_x;
    x += 1;
    msg_x.Signal_x = x;
    output(msg_x);
    setTimer(messageTimer, 100);
}
on message Signal_Message_y{
    message Output y;
    y.Y = this.Signal_y;
    output(y);
}
```

```
includes
{
variables
{
    msTimer connection_timer;
}
on start
{
    setTimer(connection_timer, 10);
}
on timer connection_timer
{
    message Connection_Message B_connected;
    B_connected.BIsConnected = 1;
    output(B_connected);
}
on message Signal_Message_X
{
    message Signal_Message_y msg_y;
    msg_y.Signal_y =
    sin(this.Signal_x)*cos(this.Signal_x/20);
    output(msg_y);
}
```

D. Wyniki

Odpowiednio połączone i zaprogramowane węzły wyświetlały wiadomość o ich połączeniu co 10ms, a w oknie graficznym wyświetlono przebieg sygnału y.



E. Wnioski

Zaznajomiłyśmy się z budową sieci CAN, sposobem przesyłania informacji, w jaki sposób definiowane są sygnały oraz wiadomości. Zapoznaliśmy się z podstawowymi funkcjami programu CANoe. Mimo drobnych napotkanych trudności, które wynikały z pierwszego kontaktu z programem udało nam się w pełni zrealizować zakres przewidzianego ćwiczenia: nie napotkaliśmy błędów przy dokonywaniu pomiarów, a przesłanie wiadomości zakończył się sukcesem. Otrzymałyśmy wykres sygnału y zgodny z oczekiwaniami. Prezentuje on iloczyn dwóch funkcji trygonometrycznych – obwiednię $\cos(x/20)$, a wewnątrz niej funkcję $\sin(x)$ dla wartości x zmiennej w czasie.