

A. Informacje o zespole realizującym ćwiczenie

Nazwa przedmiotu: Automatyka pojazdowa	
Nazwa ćwiczenia:	Architektura systemów sterowania w samochodach
Data ćwiczenia:	2019-03-20
Czas ćwiczenia:	08:00 – 09:30
Zespół realizujący ćwiczenie:	<ul style="list-style-type: none">• Katarzyna Wątorska• Jacek Wójtowicz• Bartłomiej Mróz



B. Sformułowanie problemu

Zajęcia miały na celu zbudowanie w środowisku CANoe prostej architektury systemowej zawierającej dwa komunikujące się ze sobą moduły.

Węzły powinny wysyłać wiadomości co 10 ms w celu sprawdzenia połączenia.

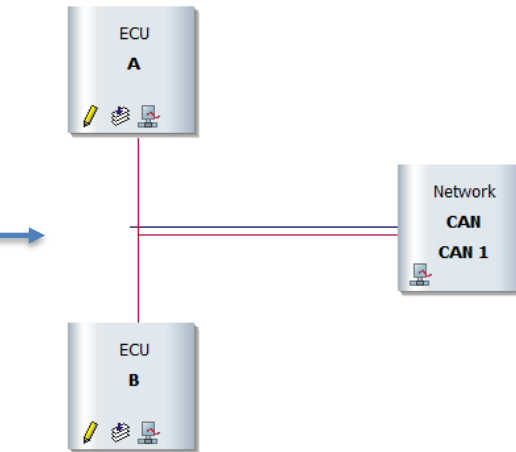
Składa się ona z sygnałów x i y. Węzeł A co 100 ms ma zwiększać wartość sygnału x, a węzeł B po otrzymaniu sygnału x od A wylicza sygnał y z poniższego wzoru i przesyła go do A:

$$y = \sin(x) \cdot \cos\left(\frac{x}{20}\right)$$

Zaobserwowano przesyłane sygnały i graficznie przedstawiono wartości sygnału y.

C. Sposób rozwiązania problemu

Schemat



Kod węzła B

```

1  /*@!Encoding:1250*/
2  #includes
3  {
4    #include "nodeA.can"
5  }
6
7  variables
8  {
9    message 0x400 msg_B;
10 }
11
12
13 on start{
14   msg_B.y = 0;
15 }
16
17 on message connection{
18   output(connection);
19   write("There's a connection between nodes!~B");
20 }
21
22 on message msg_A{
23   msg_B.y = sin(msg_A.x)*cos(msg_A.x/20);
24   output(msg_B);
25 }
26

```

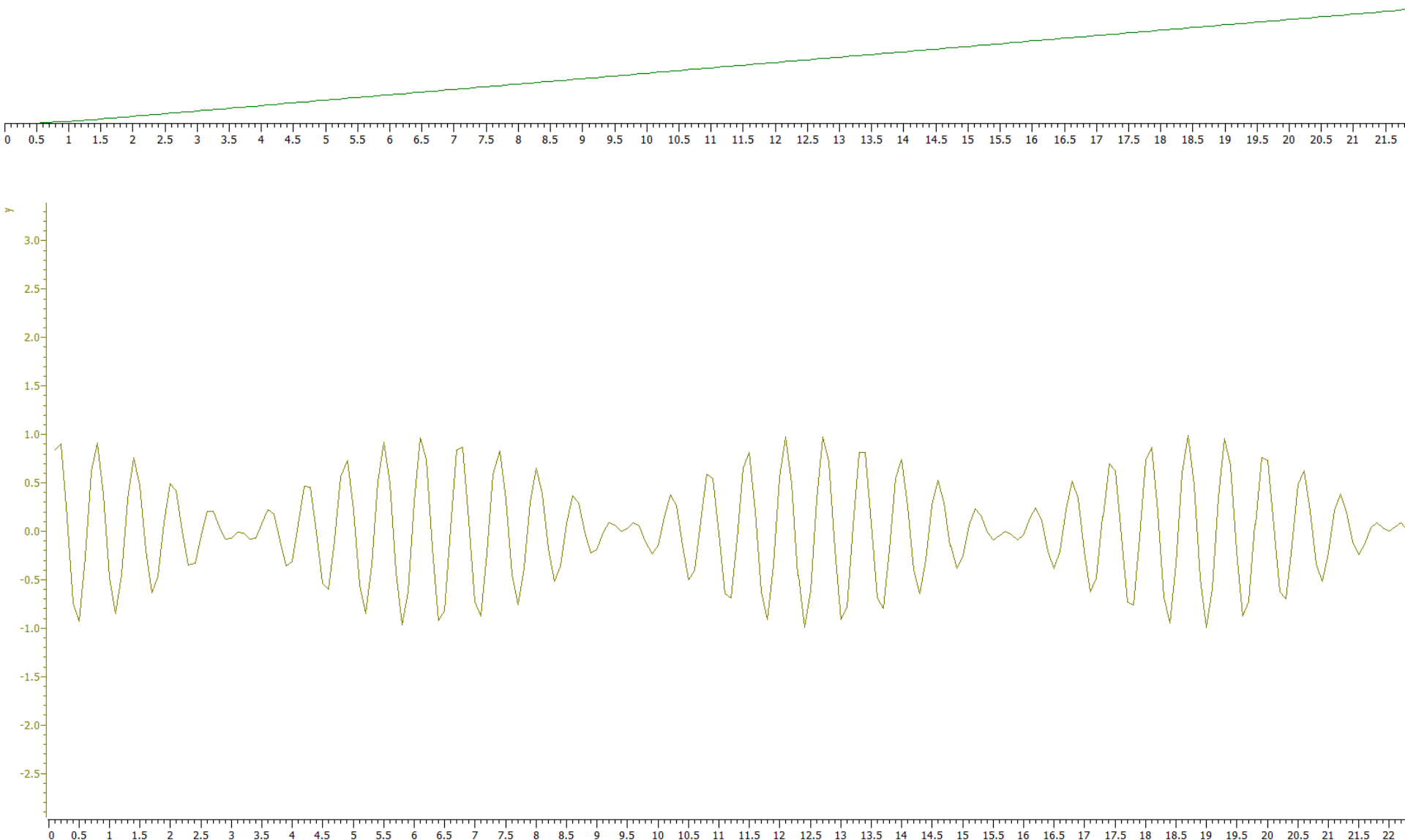
```

1  /*@!Encoding:1250*/
2  #includes
3  {
4  }
5
6
7  variables
8  {
9    message 0x200 msg_A;
10   message 0x500 connecti
11   msTimer timer1;
12   msTimer timer2;
13 }
14
15 on start{
16   output(connection);
17   setTimer(timer1, 10);
18   setTimer(timer2,100);
19 }
20
21 on timer timer1{
22   setTimer(timer1, 10);
23   output(connection);
24 }
25
26 on timer timer2{
27   setTimer(timer2,100);
28   msg_A.x = msg_A.x+1;
29   output(msg_A);
30 }

```

Kod węzła B

D. Wyniki



E. Wnioski

Wykorzystując środowisko CANoe utworzono dwa komunikujące się z sobą moduły, a następnie przeprowadzono symulację.

Mieliśmy okazję zaznajomić się z samym środowiskiem CANoe i poznać podstawy operowania w nim, takie jak tworzenie i wyświetlanie zmiennych, podłączanie bibliotek, komunikacja między modułami.

Zebrana wiedza i doświadczenie będzie przydatna przy tworzeniu bardziej skomplikowanych systemów i symulacji związanych z automatyką pojazdową w środowisku CANoe.