



# **Systemu czasu rzeczywistego**

**Sprawozdanie z projektu pt.  
„Sterownik roweru elektrycznego”**

**Autor: Piotr Kaniuk**

E-mail: baniuk@student.agh.edu.pl

## I. Opis modelowanego systemu.

### 1. Opis ogólny:

Model opisuje uproszczony elektroniczny układ sterowania roweru elektrycznego. System został przedstawiony przy użyciu języka AADL i składa się z zestawu urządzeń wejściowych, procesu odpowiedzialnego za przetwarzanie danych oraz elementu wykonawczego sterującego silnikiem. Komunikacja pomiędzy komponentami odbywa się poprzez porty danych, a struktura systemu została uporządkowana na trzech poziomach: urządzenia (device), proces (process) wraz z wątkami (threads) oraz elementy platformy (processor, memory, bus).

W modelu założono, że każdy sensor generuje dane okresowo, a główny proces kontrolny pobiera informacje z manetki (throttle), baterii oraz sensora prędkości. Na podstawie tych danych wyliczana jest komenda sterująca wysłana do sterownika silnika. Całość systemu jest osadzona na prostym modelu platformy obliczeniowej składającej się z procesora, pamięci i magistrali komunikacyjnej.

### 2. Opis dla użytkownika (opis funkcjonalny):

System sterowania roweru elektrycznego umożliwia użytkownikowi płynną kontrolę nad pracą silnika w zależności od pozycji manetki, stanu baterii i aktualnej prędkości. Działanie przebiega w następujący sposób:

1. Manetka (*Throttle\_Device*) przekazuje informację o żądanej mocy.
2. Moduł baterii (*Battery\_Device*) dostarcza aktualny poziom naładowania.
3. Czujnik prędkości (*Speed\_Sensor*) wysyła bieżące informacje o prędkości roweru.
4. Wszystkie te sygnały trafiają do procesu sterującego *E\_Bike\_Process*, który zawiera trzy wątki:
  - odczyt manetki,
  - monitorowanie baterii,
  - logika sterowania silnikiem.
1. Wątek sterowania łączy dane z pozostałych modułów i generuje sygnał sterujący *Motor\_Command*, który trafia do urządzenia wykonawczego *Motor\_Driver*.

Użytkownik końcowy widzi jedynie efekt w postaci odpowiedniej reakcji silnika na zmianę położenia manetki, z uwzględnieniem stanu baterii i prędkości pojazdu.

## **II. Spis komponentów AADL z komentarzem.**

### **1. Typy danych:**

*Battery\_Level*, *Motor\_Command*, *Throttle\_Input*, *Speed\_Signal* to typy danych reprezentujące sygnały przesyłane między modułami.

### **2. Urządzenia:**

*Throttle\_Device* - Urządzenie reprezentujące manetkę. Generuje wartość wejściową dla układu.

*Battery\_Device* - Źródło informacji o stanie baterii.

*Speed\_Sensor* - Urządzenie pomiarowe przekazujące prędkość.

*Motor\_Driver* - Element wykonawczy realizujący komendy sterujące silnikiem.

### **3. Wątki:**

*Throttle\_Reader* - Pobiera sygnał z manetki, ewentualnie filtry i przekazuje dalej.

*Battery\_Monitor* - Odbiera i propaguje stan baterii.

*Motor\_Control* - Główny wątek decyzyjny systemu. Łączy dane z manetki, baterii i prędkości, generując komendę dla silnika.

### **4. Procesy:**

*E\_Bike\_Process* - Łączy wszystkie trzy wątki i definiuje interfejs komunikacyjny z resztą systemu. Przyjmuje sygnały wejściowe z urządzeń i wysyła sterowanie na silnik.

### **5. Procesor, pamięć, magistrala:**

*ECU / ECU.impl* - Reprezentacja jednostki obliczeniowej.

*ECU\_RAM* - Pamięć systemu.

*CommBus* - Magistrala obsługująca połączenia pomiędzy elementami systemu.

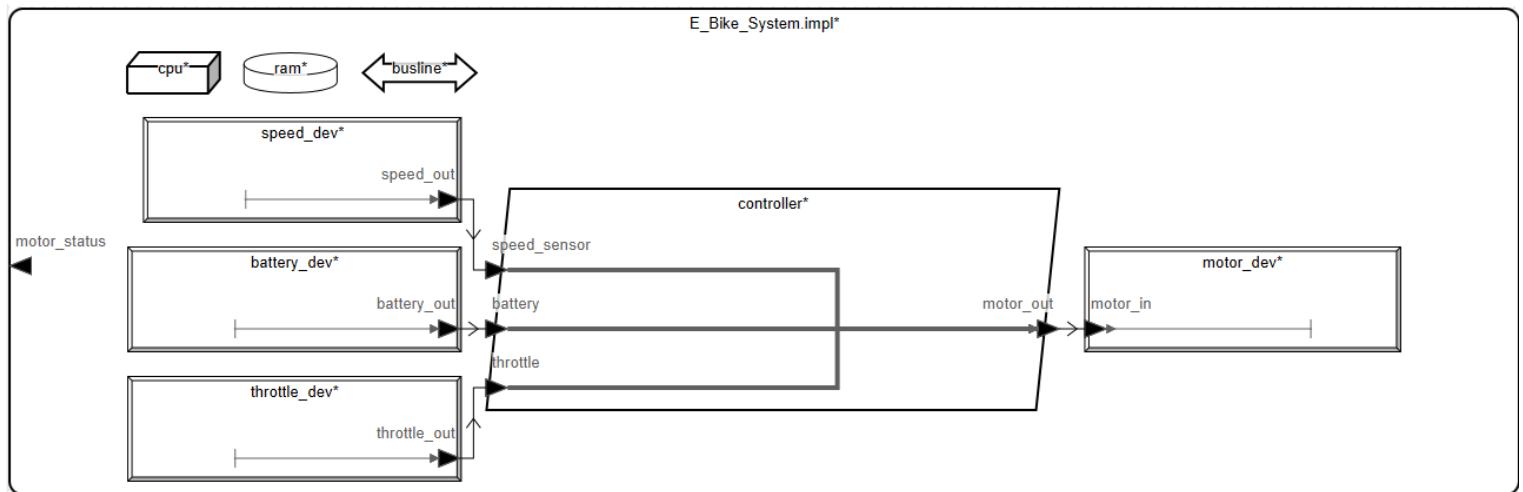
### **6. System i implementacja:**

*E\_Bike\_System* - Wysokopoziomowa definicja systemu e-bike.

*E\_Bike\_System.impl* - Konkretna instancja systemu zawierająca urządzenia, proces oraz powiązania komunikacyjne. Definiuje połączenia:

- manetka → proces,
- bateria → proces,
- sensor prędkości → proces,
- proces → sterownik silnika, oraz przypisania platformowe (processor binding, memory binding, bus binding).

### III. Model – Rysunek.



### IV. Inne informacje zależne od tematu.

W modelu założono, że:

- urządzenia wejściowe (manetka, bateria, sensor prędkości) generują dane okresowo i niezależnie,
- proces sterujący jest odpowiedzialny za łączenie tych informacji oraz generowanie komendy sterującej silnikiem,
- komunikacja odbywa się wyłącznie poprzez porty danych, bez modelowania opóźnień transmisyjnych,
- platforma obliczeniowa została uproszczona i nie uwzględnia wielowątkowego planowania ani czasu wykonania.

Model ten może stanowić bazę do dalszych rozszerzeń, takich jak:

- dodanie analiz schedulability,
- wprowadzenie flow paths i flow latency,
- modelowanie awarii czujników,
- rozszerzenie procesu o dodatkowe algorytmy wspomagania jazdy.

### V. Literatura.

- SAE AS5506C – Architecture Analysis & Design Language (AADL) Standard** - Podstawowy dokument definiujący składnię i semantykę AADL.
- OSATE User Guide & Reference** - Dokumentacja narzędzia OSATE zawierająca opis dostępnych analiz i sposobu pracy z modelem instancji.