



AKADEMIA GÓRNICZO-HUTNICZA  
IM. STANISŁAWA STASZICA W KRAKOWIE



# LEGO Segway

**Budowa i analiza strukturalnie niestabilnego  
robota mobilnego na bazie projektu Segway**



**Wydział Elektrotechniki, Automatyki, Informatyki i Elektroniki**

**Kraków, 29.05.2009**



# Projekt LEGO Segway

Projekt realizowany w ramach  
koła naukowego INTEGRA



Inspiracja i opieka:  
dr inż. Adam Piłat

Zespół:  
Łukasz Bondyra  
Paweł Górka  
Jakub Tutro  
Krzysztof Wesółowski

Prezentuje:  
Krzysztof Wesółowski

# Plan prezentacji



- **Co to jest Segway?**
- **Układy strukturalnie niestabilne**
- **Wahadło odwrócone**
- **Klocki LEGO Mindstorms NXT**
- **Budowa i działanie NXT Brick**
- **Systemy RTOS**
- **Pakiet Matlab/SIMULINK**
- **ECRobot NXT**
- **Inspiracja – pierwszy projekt**
- **Przegląd wykonanych zadań**
- **Podsumowanie**

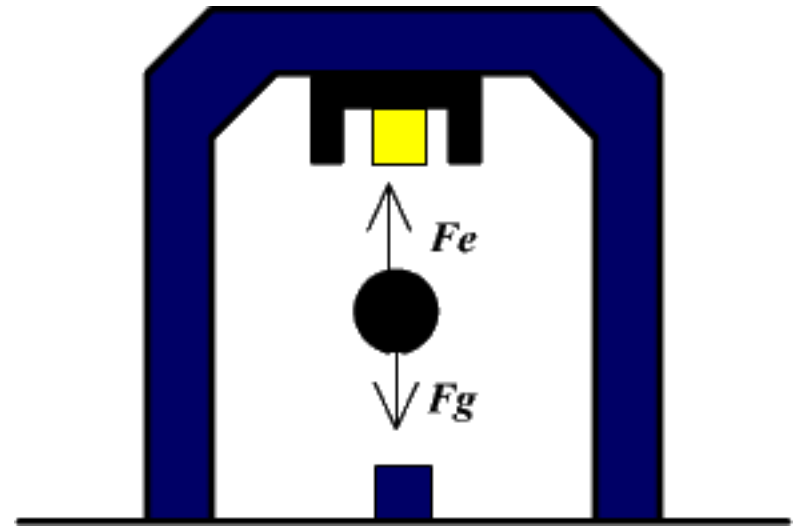
# Co to jest Segway?

- Przeznaczony do transportu ludzi  
udźwig dochodzi **do 200 kg**
- Waga  
w zależności od wersji **48 – 56 kg**
- Maksymalna moc  
2 x 2KM czyli ponad **3000 W**
- Prędkość  
dochodzi do **20 km/h**
- Koszt badań nad rozwojem  
**100 000 000 \$**



# Układy strukturalnie niestabilne

- Współczesne samoloty wojskowe
- Kula lewitująca w polu magnetycznym
- Wahadło odwrócone

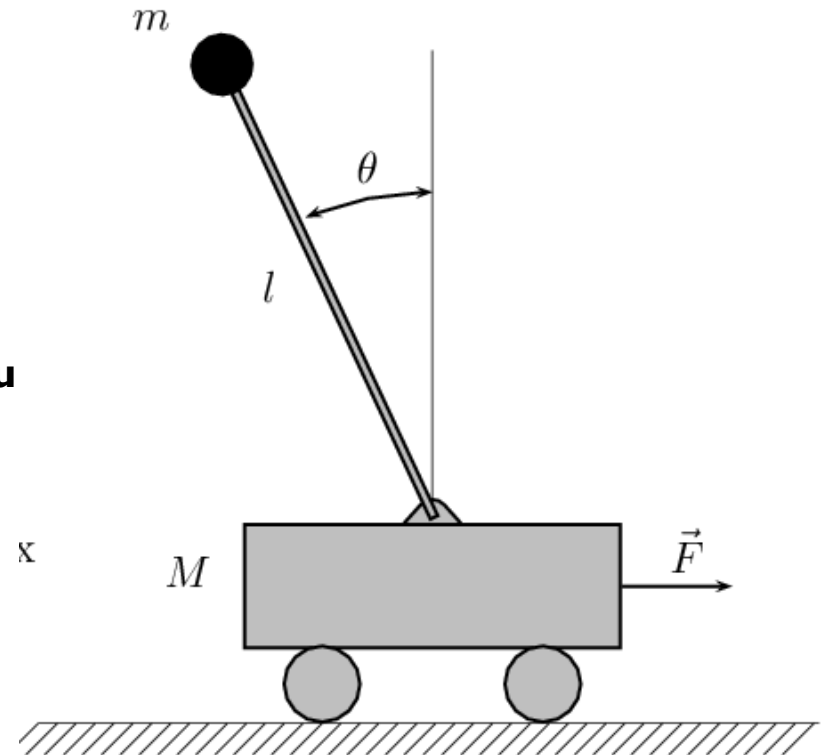


# Wahadło odwrócone

## Podstawowy model – wahadło na wózku

- **Cztery zmienne stanu**
  - Prędkość kątowa ramienia
  - Kąt odchylenia ramienia
  - Położenie wózka
  - Prędkość wózka
- **Zależności między nimi**
- **Szczegóły, opis matematyczny modelu**

Andrzej Turnau - Target Control of Nonlinear Systems in Real Time - Intelligent and Time-optimal Algorithms, wydawnictwo AGH Kraków 2002



# Wahadło odwrócone - przykłady



- Celowa niestabilność
- Porównanie z wahadłem

# Klocki LEGO Mindstorms NXT

## •Sensory

- Odległości (ultradźwiękowy)
- Dźwięku (natężenie)
- Jasności
- Dotykowy
- Pozostałe

## •Silniki

- Zasilane prądem stałym
- Regulowana prędkość (PWM)
- Wbudowana przekładnia
- Wbudowane enkodery



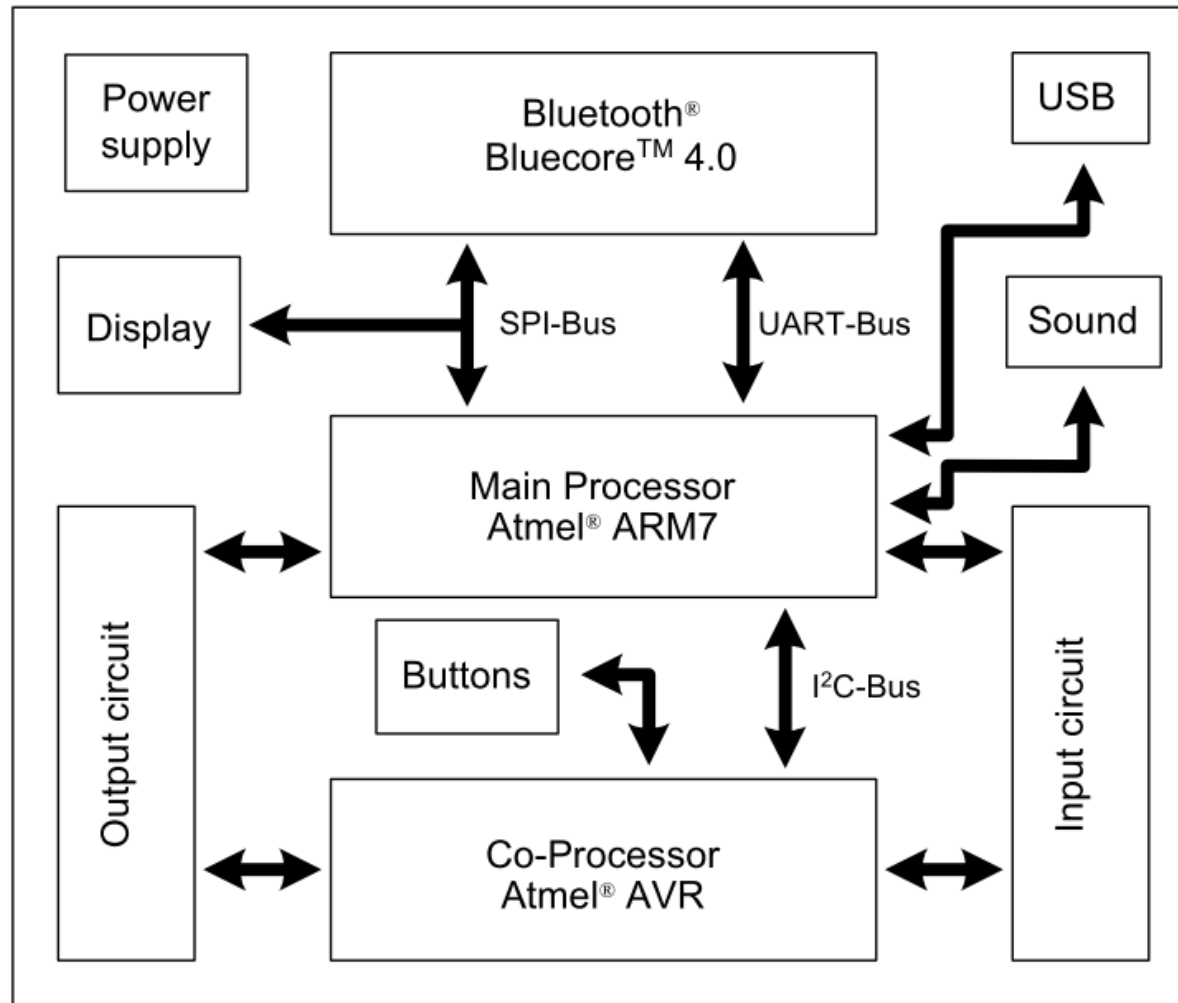
## NXT Brick – serce zestawu

- Cztery wejścia na czujniki
- Trzy porty silników
- Obsługa komunikacji
- Wyświetlacz i głośnik



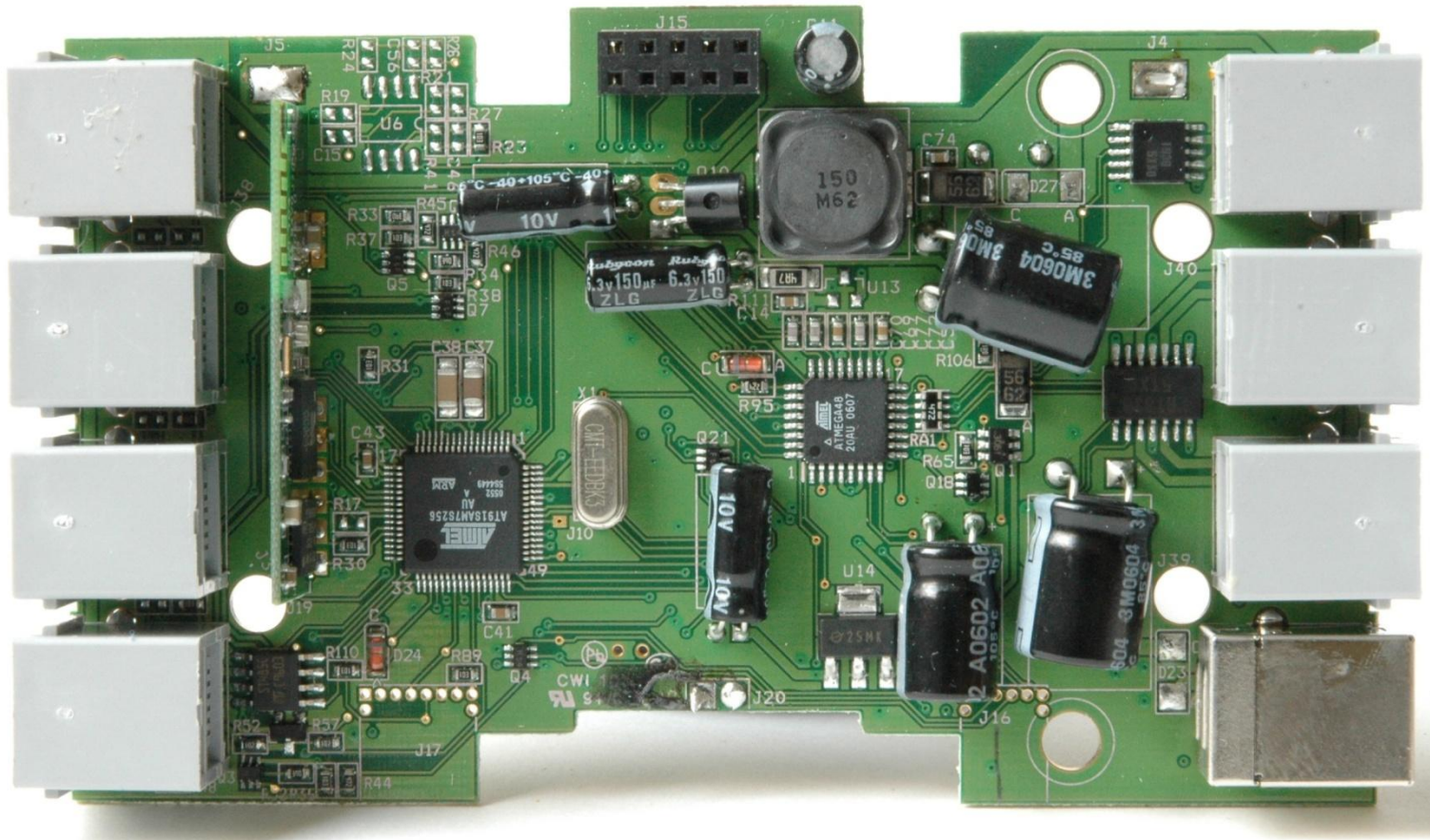
- Wbudowane mikrokontrolery:
- ARM7, AVR

# Schemat ideowy NXT



# Płytką drukowaną

Płytką PCB znajdującą się wewnątrz NXT Brick





# Sposoby programowania NXT

- **Next Byte Codes**
- **Not eXactly C**
- **LeJOS + NXJ**
- **LabVIEW Toolkit for LEGO**
  
- **NxtOSEK**
- **MATLAB/Simulink + ECRobot NXT**

# Systemy operacyjne czasu rzeczywistego

- Zwykłe systemy operacyjne
- Systemy czasu rzeczywistego (RTOS)
  - O twardych wymaganiach (Hard)
  - O miękkich wymaganiach (Soft)

**Obiekty niestabilne można sterować tylko  
za pomocą systemów Hard Real Time!**

## Przykłady RTOS

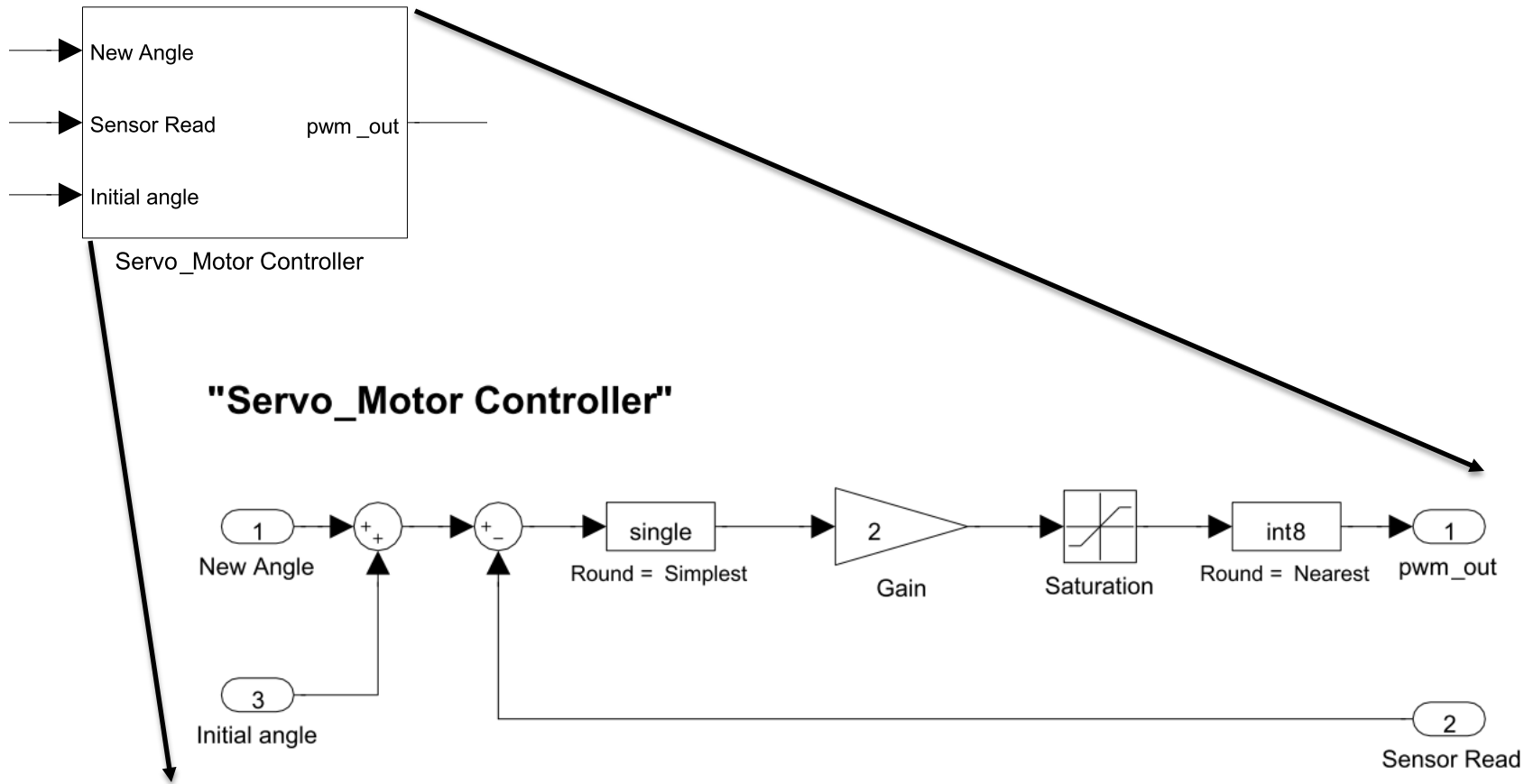
<b>System</b>	<b>Platforma</b>	<b>Opis</b>
<b>FreeRTOS</b>	<b>AVR32, ARM7, ARM9, Cortex M3, MicroBlaze</b>	<b>Prosty darmowy RTOS na wiele platform</b>
<b>Phoenix-RTOS</b>	<b>ARM7</b>	<b>Darmowy RTOS początkowo stworzony na mikrokontrolery LPC2000 firmy Philips</b>
<b>LynxOS</b>	<b>ARM7, ARM9, PowerPC, PC</b>	<b>Przemysłowy system dla urządzeń wbudowanych, płatna licencja</b>
<b>uCLinux</b>	<b>ARM7, MicroBlaze, Motorola DragonBall</b>	<b>OpenSource'owy system oparty na jądrze linux'a</b>
<b>TOPPERS OSEK</b>	<b>ARM7, PIC</b>	<b>RTOS używany w motoryzacji w systemach typu ABS, ESP itp.</b>



# **nxtOSEK**

- **Architektura TOPPERS OSEK**
- **Przygotowany specjalnie dla NXT**
- **Dostarcza API dla czujników i silników**
- **Łączony z kodem C/C++ (zadania)**
- **Utrzymujący reżimy czasowe.**
- **Wydajny, stabilny**

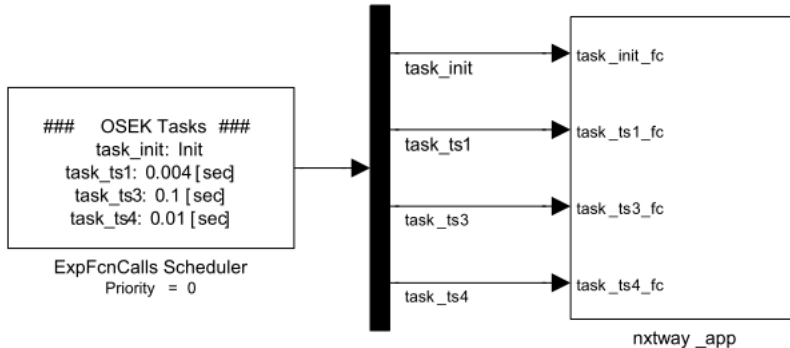
# Matlab/Simulink



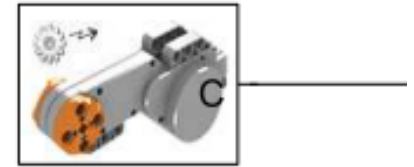
# Embedded Coder Robot NXT

- **Bloki wejścia/wyjścia**
- **Blok uruchamiający zadania**
- **Wsparcie dla Model Based Design**

## Planowanie zadań

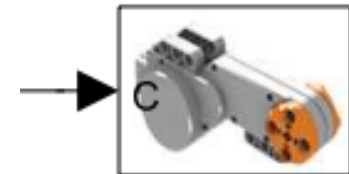


## Odczyt sensora



Revolution Sensor Read  
Port = C

## Zapis wyjścia



Servo Motor Write  
Port = C



# Automatyczna generacja kodu

- Wykorzystanie Toolkitów Matlab
- Łatwość „pisania” aplikacji
- Kod w języku C
- Zgodność z wieloma platformami
- Stosowanie w komercyjnych zastosowaniach
- Integracja z ECRobot NXT



# Opis zadania

- **Sterowanie robotem „Segway”**
- **Zapoznanie się z**
  - **Platformą NXT**
  - **Systemami OSEK**
  - **Toolkitem Embedded Coder**
- **Modyfikacja i ulepszanie istniejącego modelu.**
- **Uodpornienie algorytmu na zakłócenia**

# Inspiracja i pierwszy model

## Model autorstwa pana Yorihisu Yamamoto

### Cechy:

- **Lekka, prosta konstrukcja**
- **Stały moment bezwładności**
- **Duże, sztywne koła**
- **Sterowanie z poziomu komputera lub prosty tryb autonomiczny**
- **Realizacja logiki i sterowania w jednym zadaniu**



# Rozbudowa i modyfikacje

- Odtworzenie na dostępnych klockach istniejącego robota
- Poprawienie parametrów istniejącego algorytmu
- Zmodyfikowanie konstrukcji mechanicznej
- Dodanie zmiennego momentu bezwładności
- Przygotowanie robota do występów w konkursie



# Kolejne etapy projektu

- Konfiguracja wymaganych narzędzi
- Modyfikacja firmware'u robota
- Uruchomienie robota na małych kołach
- Zmiana konstrukcji na masywniejszą
- Ruchoma głowa – zmiana parametrów fizycznych w czasie jazdy
- Uniezależnienie robota od otoczenia
- Pozostałe





## Zadawanie trasy - taniec

- **Korzystanie ze zmiennych współdzielonych**
  - **Prędkość postępową**
  - **Kąt względem początkowej pozycji**
- **Łatwe rysowanie toru robota**
- **Integracja tras z oprogramowaniem**

## Konkurs w Bratysławie

- Konkurs robotów autonomicznych
- Kategoria Freestyle
- Udany taniec zakończony II miejscem



**ISTROBOT**  
**2009**

# Podsumowanie

- **Plany na przyszłość**
  - **Własna platforma sprzętowa**
  - **Budowa prototypu Segway'a**
  - **Poprawa jakości sterowania**

