

# Tytuł: Cardog

## Autorzy: Gabriel Płasiński (GP), Michał Wałek (MW)

Ostatnia modyfikacja: 02.09.2024

### Spis treści

1. Repozytorium git.....	2
2. Wstęp .....	2
3. Specyfikacja .....	2
3.1. Opis ogólny algorytmu .....	2
3.2. Tabela zdarzeń .....	3
4. Architektura.....	4
4.1. Moduł: top .....	4
4.1.1. Schemat blokowy .....	5
4.1.2. Porty.....	6
a) jstk2spi / spi2jstk – input/output.....	6
b) proxy – input.....	6
c) led – output .....	6
d) sseg – output .....	6
e) servo_y - output .....	6
f) servo_x - output.....	6
4.1.3. Interfejsy .....	7
a) spi2track – JSTK_SPI_if to Backtrack .....	7
b) spi2led – JSTK_SPI_if to Led_Ctrl.....	7
c) spi2sseg – JSTK_SPI_if to sseg_ctrl .....	7
d) t2sseg – Backtrack to sseg_ctrl.....	7
e) track2sens – Backtrack to Sensor_ctrl .....	7
f) sens2x – Sensor_ctrl to Steering_X .....	7
g) track2y – Backtrack to Steering_Y.....	7
4.2. Rozprowadzenie sygnału zegara .....	8
5. Implementacja.....	8
5.1. Lista zignorowanych ostrzeżeń Vivado.....	8
5.2. Wykorzystanie zasobów .....	9
5.3. Marginesy czasowe .....	9
6. Film .....	9
7. Informacje dodatkowe.....	9

## 1. Repozytorium git

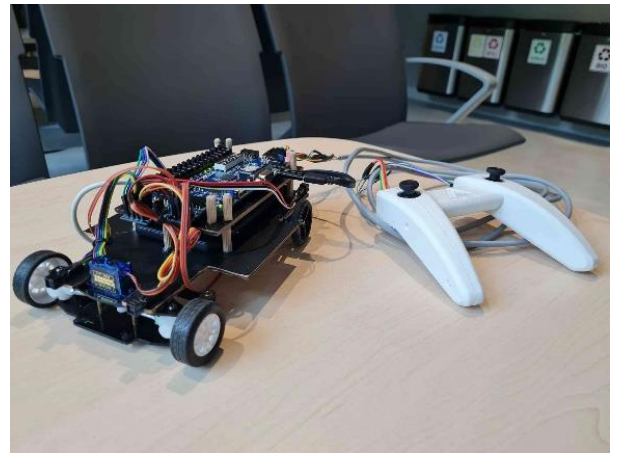
Adres repozytorium GITa (jeżeli używane):

<https://github.com/MichalWlk/UEC2-Servo-SPI-steering>

W przypadku repozytorium prywatnego należy zaprosić użytkownika zewnętrznego o adresie mailowym: [kaczmarczyk@agh.edu.pl](mailto:kaczmarczyk@agh.edu.pl)

## 2. Wstęp

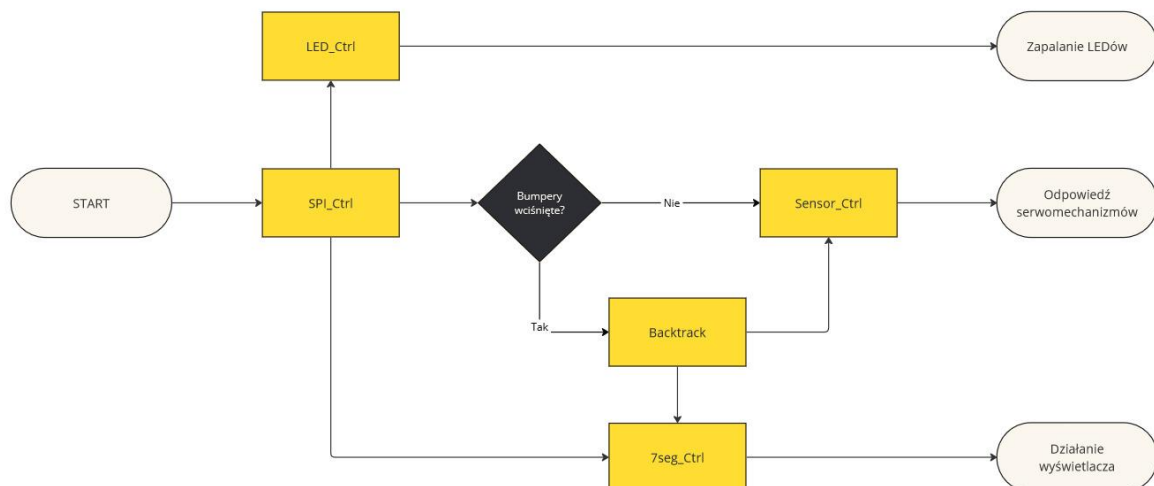
Pomysł na projekt wziął się z „zajawki” do motoryzacji i chęci stworzenia czegoś na wzór zabawki z dzieciństwa. W ramach projektu zbudowaliśmy pojazd sterowany przewodowo padem wydrukowanym przez drukarkę 3D. „Cardog” bo tak nazwaliśmy nasze dzieło posiada tył napęd który składa się z 2 serwomechanizmów oraz osi skrętnej umiejscowionej z przodu. Serce stanowi Basys3 a źródłem zasilania jest powerbank i obie te rzeczy wraz z czujnikami IR są zamontowane na platformie wykonanej z płyty HDF.



## 3. Specyfikacja

### 3.1. Opis ogólny algorytmu

Uproszczony schemat blokowy działania implementowanego algorytmu. Co się dzieje po starcie, jak wygląda przebieg działania, kiedy i pod jakimi warunkami się kończy.



### 3.2. Tabela zdarzeń

Opis zdarzeń występujących podczas działania programu/urządzenia, zarówno zewnętrznych (interakcje z użytkownikiem), jak i wewnętrznych (specyficzne stany w algorytmie). Zdarzenia podzielone są na kategorie dotyczący różnych stanów działania programu. Kategorie powinny odpowiadać stanom ze schematu z pkt. 2.1.

Zdarzenie	Kategoria	Reakcja systemu
Lewy joystick do góry	Pad	Pojazd jedzie do przodu
Lewy joystick do dołu	Pad	Pojazd jedzie do tyłu
Prawy joystick w lewo	Pad	Pojazd skręca w lewo
Prawy joystick w prawo	Pad	Pojazd skręca w prawo
Oba przyciski wciśnięte	Pad	Pojazd powtarza samoczynnie ruchy wykonane przez ostatnie 5 sekund
Lewy joystick do góry	7seg	Wyświetla się „d” (drive)
Lewy joystick do dołu	7seg	Wyświetla się „r” (reverse)
Oba przyciski wciśnięte	7seg	Wyświetla się „t” (track)
Brak czynności	7seg	Wyświetla się „n” (neutral)
Wykrycie obiektu przez jeden z czujników przednich	Czujniki IR	Brak możliwości jazdy do przodu tylko jazda do tyłu
Wykrycie obiektu przez jeden z czujników tylnych	Czujniki IR	Brak możliwości jazdy do tyłu tylko jazda do przodu
Lewy joystick do góry	Diody LD7 do LD0	Im bardziej wychylony joystick tym więcej diod się świeci zaczynając od LD7
Lewy joystick do dołu	Diody LD8 do LD15	Im bardziej wychylony joystick tym więcej diod się świeci zaczynając od LD8

## 4. Architektura

### 4.1. Moduł: top

Osoba odpowiedzialna: MW

SPI\_Steering\_top.sv

JSTK2\_SPI\_interface.sv

Clk\_Div.sv

SPI\_Ctrl.sv

JSTK\_SPI\_if\_X.sv

Clk\_Div\_X.sv

SPI\_Ctrl\_X.sv

Led\_Ctrl.sv

Led\_Comp.sv

Led\_PWM.sv

Backtrack.sv

Timer\_5s.sv

Clk\_Div\_5Hz.sv

Action\_Translator.sv

Val\_SR.sv

sseg\_ctrl.sv

anode\_ctrl.sv

char\_ctrl.sv

sseg\_out.sv

clk\_div\_1k.sv

Sensor\_Ctrl.sv

Steering\_X.sv

Counter\_X.sv

Left\_Ctrl.sv

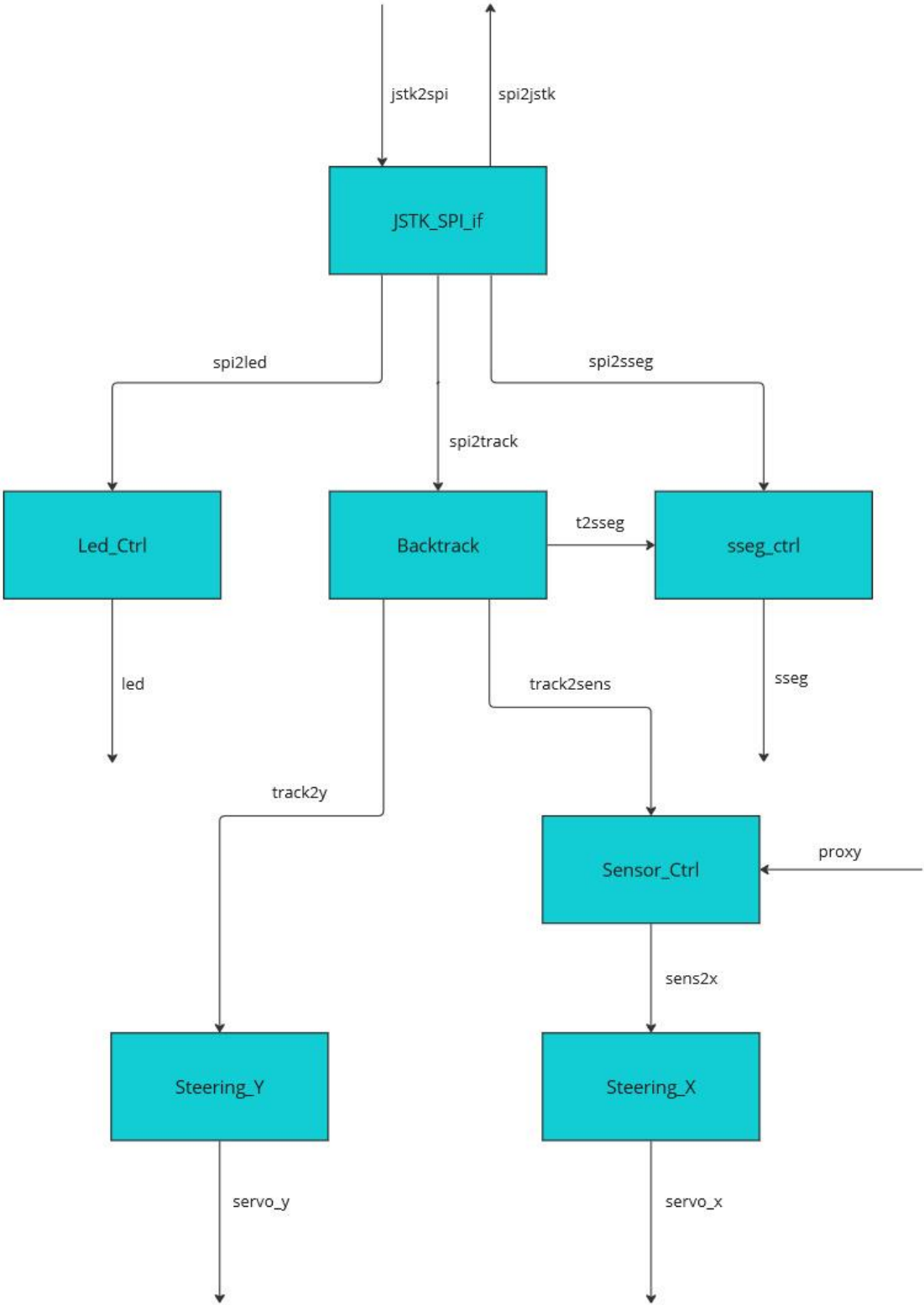
Steering\_Y.sv

Counter\_Y.sv

Turn\_Ctrl.sv

Right\_Ctrl.sv

4.1.1. Schemat blokowy



#### 4.1.2. Porty

##### a) *jstk2spi / spi2jstk – input/output*

nazwa portu	opis
MISO (y)	szeregowe wejście danych
MISO (x)	szeregowe wejście danych
SS (y)	Slave select
SS (x)	Slave_select
SCLK (y)	Zegar napędzający pmod JSTK
SCLK (x)	Zegar napędzający pmod JSTK

##### b) *proxy – input*

nazwa portu	opis
PROX_FL	Sygnał czujnika gdy wykryje przeszkodę (lewy przód)
PROX_FR	Sygnał czujnika gdy wykryje przeszkodę (prawy przód)
PROX_RL	Sygnał czujnika gdy wykryje przeszkodę (lewy tył)
PROX_RR	Sygnał czujnika gdy wykryje przeszkodę (prawy tył)

##### c) *led – output*

nazwa portu	opis
led[15:0]	Sygnał kiedy czujnik wykryje przeszkodę (lewy przód)

##### d) *sseg – output*

nazwa portu	opis
an[3:0]	Wybór 1 z 4 miejsc na wyświetlaczu
seg[6:0]	Wybór podświetlonych segmentów

##### e) *servo\_y – output*

nazwa portu	opis
PWM_y	Sygnał PWM sterujący osią skrętną

##### f) *servo\_x – output*

nazwa portu	opis
PWM_x	Sygnał PWM sterujący napędem

#### 4.1.3.

#### 4.1.4. Interfejsy

##### a) *spi2track – JSTK\_SPI\_if to Backtrack*

nazwa sygnału	opis
x_val[10:0]	Wartość wychylenia drążka do przód/tył
y_val[10:0]	Wartość wychylenia drążka do lewo/prawo
Bumper_x	Informacja o wciśniętym przycisku od JSTK x
Bumper_y	Informacja o wciśniętym przycisku od JSTK y

##### b) *spi2led – JSTK\_SPI\_if to Led\_Ctrl*

nazwa sygnału	opis
velocity [10:0]	Wartość prędkości bazującej (wychylenia drążka w osi x)

##### c) *spi2sseg – JSTK\_SPI\_if to sseg\_ctrl*

nazwa sygnału	opis
x_val [10:0]	Wartość wychylenia drążka do przód/tył

##### d) *t2sseg – Backtrack to sseg\_ctrl*

nazwa sygnału	opis
backtrack_active	Informacja czy włączony jest tryb Backtrack

##### e) *track2sens – Backtrack to Sensor\_ctrl*

nazwa sygnału	opis
x_val[10:0]	Wartość wychylenia drążka do przód/tył

##### f) *sens2x – Sensor\_ctrl to Steering\_X*

nazwa sygnału	opis
x_val[10:0]	Wartość wychylenia drążka do przód/tył

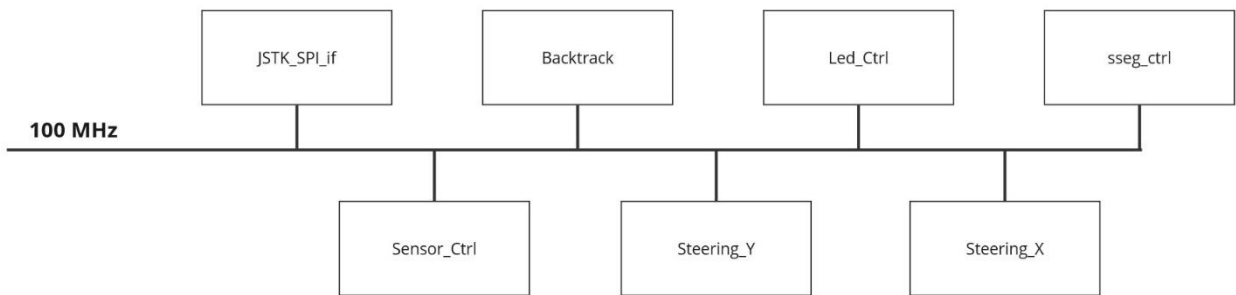
##### g) *track2y – Backtrack to Steering\_Y*

nazwa sygnału	opis
y_val[10:0]	Wartość wychylenia drążka do lewo/prawo

4.2. Rozprowadzenie sygnału zegara

Osoba odpowiedzialna: MW

W projekcie używany jest domyślny sygnał zegarowy o częstotliwości 100 MHz, który doprowadzony jest do wszystkich modułów. Niektóre moduły zostały spowolnione poprzez dodanie instrukcji warunkującej wykonanie programu, tylko wtedy, gdy sygnał Clock Enable (generowany w osobnych modułach) jest wysoki.



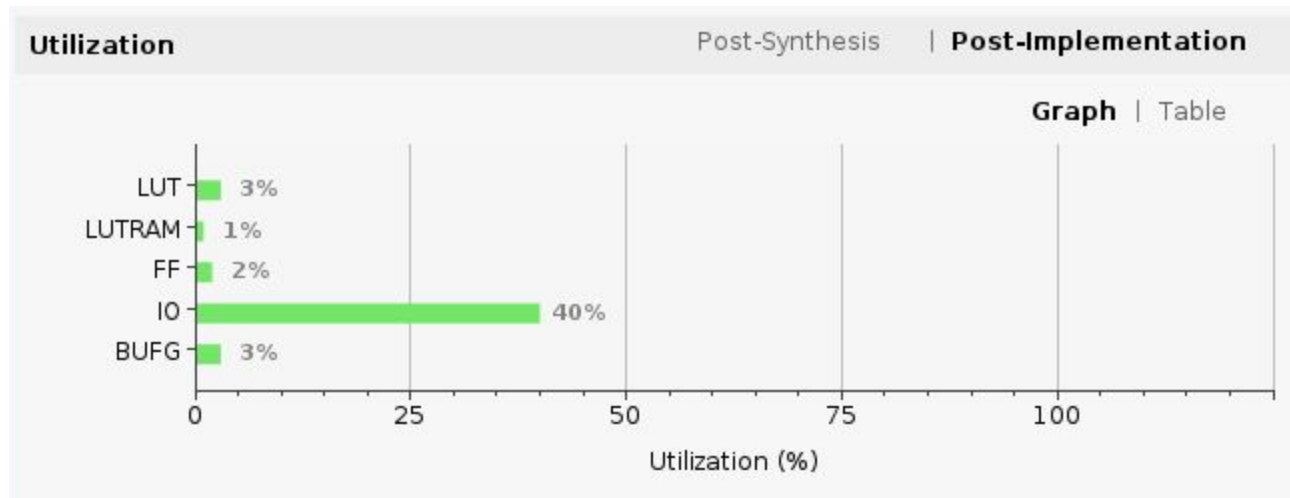
5. Implementacja

5.1. Lista zignorowanych ostrzeżeń Vivado.

Identyfikator ostrzeżenia	Liczba wystąpień	Uzasadnienie
[Synth 8-7080]	1	Informacja, że program nie jest w stanie zastosować do syntezy algorytmu równoległego. Oznacza to, że kod nie wykorzystuje tylu zasobów, by optymalizacja syntezy była możliwa.



## 5.2. Wykorzystanie zasobów



## 5.3. Marginesy czasowe

Timing		Setup	Hold	Pulse Width
Worst Negative Slack (WNS):	0.799 ns			
Total Negative Slack (TNS):	0 ns			
Number of Failing Endpoints:	0			
Total Number of Endpoints:	1779			
<a href="#">Implemented Timing Report</a>				

Timing		Setup	Hold	Pulse Width
Worst Hold Slack (WHS):	0.069 ns			
Total Hold Slack (THS):	0 ns			
Number of Failing Endpoints:	0			
Total Number of Endpoints:	1779			
<a href="#">Implemented Timing Report</a>				

## 6. Film.

Linki do ściągnięcia filmu:

<https://drive.google.com/drive/folders/1tnN-MqjzqUMWts2bJnrmXe84KneLpM4W?usp=sharing>

[https://mega.nz/folder/sqAGST7L#zthQUY\\_Fb7pHxMveTy\\_qHQ](https://mega.nz/folder/sqAGST7L#zthQUY_Fb7pHxMveTy_qHQ)

## 7. Informacje dodatkowe

Osoba odpowiedzialna za sprzęt: Gabriel Ptasiński