

robota

Oliwier Bogdański, Kacper Szponar

Github: <https://github.com/GacusPL/SWIM-Projekt>

1. Opis robota

Robot to dwukołowy autonomiczny pojazd mobilny z jednym dodatkowym kołem obrotowym dla stabilizacji, oparty na mikrokontrolerze STM32F303VCT6 Discovery. Napęd realizowany jest przez dwa silniki elektryczne z przekładniami sterowane za pomocą sygnałów PWM oraz sterownika silników L293D. Skręt wykonywany jest różnicowo – poprzez zmianę prędkości obrotowej poszczególnych silników. Układ zasilany jest autonomicznie przez trzy ogniwa Li-Ion typu 18650 każde o napięciu 3,6 V. Do zasilania mikrokontrolera, który pracuje z maksymalnym napięciem 5 V, zastosowano przetwornicę napięcia LM2596HVS, umożliwiającą obniżenie napięcia do bezpiecznego poziomu. Szacowany rozmiar robota to około 200 x 140 x 65 mm. Planowana dodatkowa funkcjonalność to dodanie głośnika.

2. Elementy wybrane do budowy robota

Podwozie: Chassis Rectangle 2WD 2-kołowe podwozie robota, oraz obrotowe koło podporowe



Napęd: 2x Koło + silnik 65x26mm 5V z przekładnią.

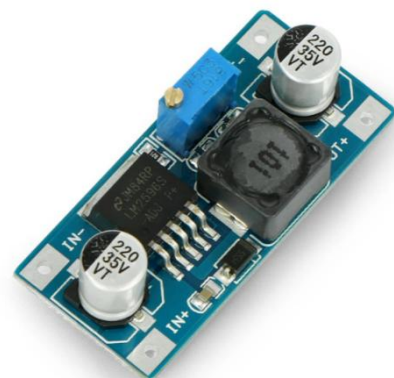




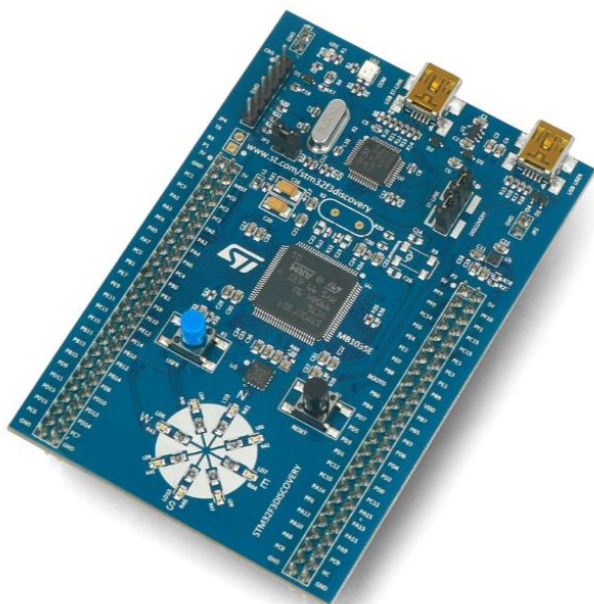
robota

Oliwier Bogdański, Kacper Szponar

Zasilanie: 3x Ogniwo 18650 Li-Ion INR18650-F1HR 3350mAh, koszyk na 3 akumulatory typu 18650 oraz przetwornica napięcia LM2596HVS



Sterowanie: Mikrokontroler STM32F303VCT6 (Discovery), sterownik silnika krokowego 2 DC L293D mini mostek H



robota

Oliwier Bogdański, Kacper Szponar

Elementy montażowe: Śrubki o rozmiarze m3, przewody połączeniowe.



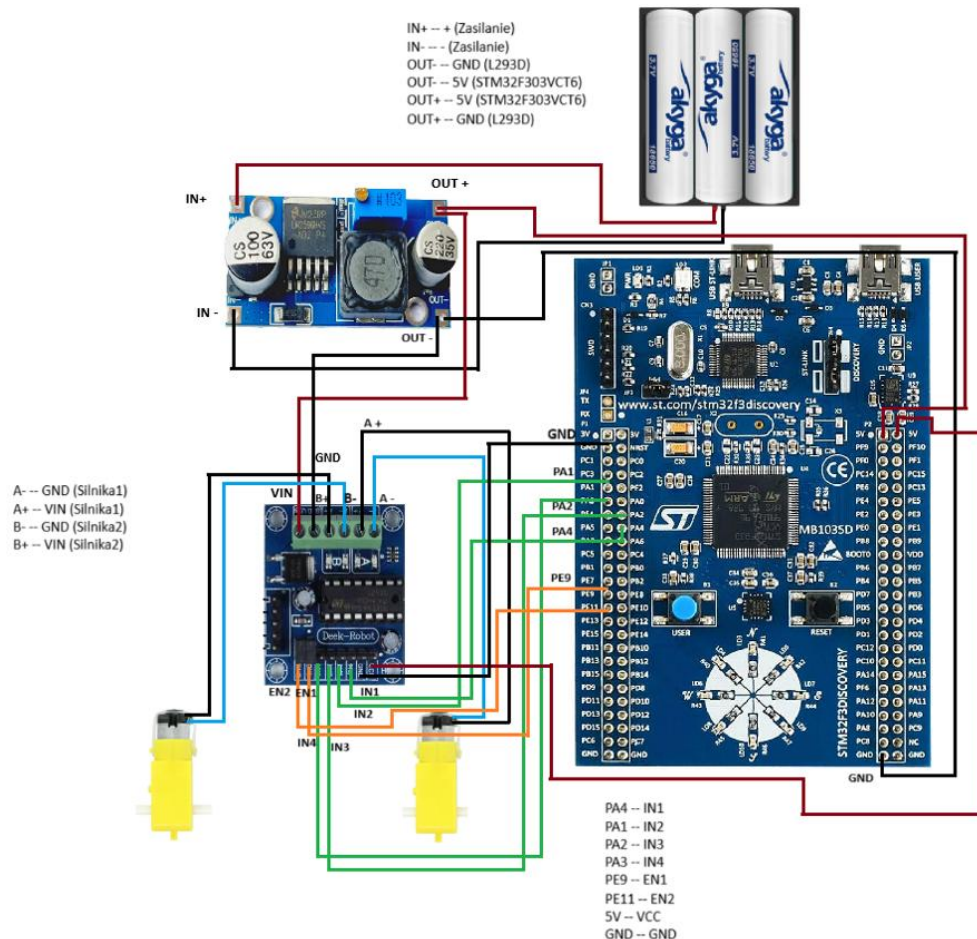
3. Mechanika robota

Mechanika robota oparta jest na podwoziu typu Chassis Rectangle 2WD, z dwukołowym napędem oraz z jednym dodatkowym kołem obrotowym. Napęd realizują 2 silniki DC 5V z wbudowaną przekładnią z kołami o wymiarach 65 x 26 mm. Elementy mechaniczne takie jak: koszyk na ogniwa, koło obrotowe, przetwornica, sterownik silników oraz same silniki zostały zamontowane za pomocą śrubek montażowych o rozmiarze m3, podzespoły zostały zamontowane w taki sposób, aby rozkład masy był jak najbardziej równomierny oraz aby zachować stabilność robota. Do połączenia wszystkich elementów zastosowaliśmy przewody połączeniowe typu jumper oraz przewody które zostały przylutowane, które zostały tak poprowadzone, aby nie kolidowały z innymi elementami.

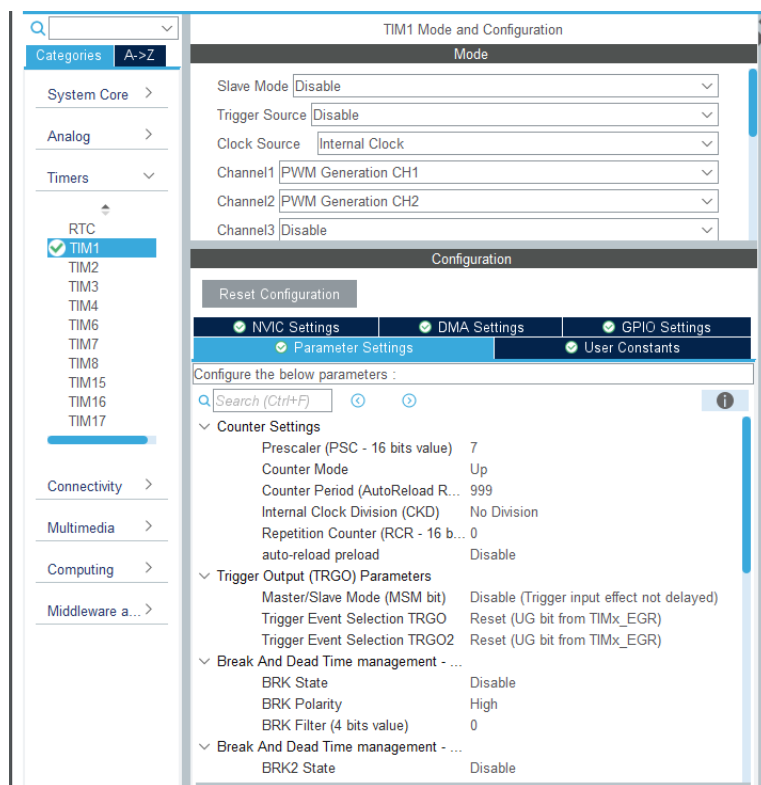
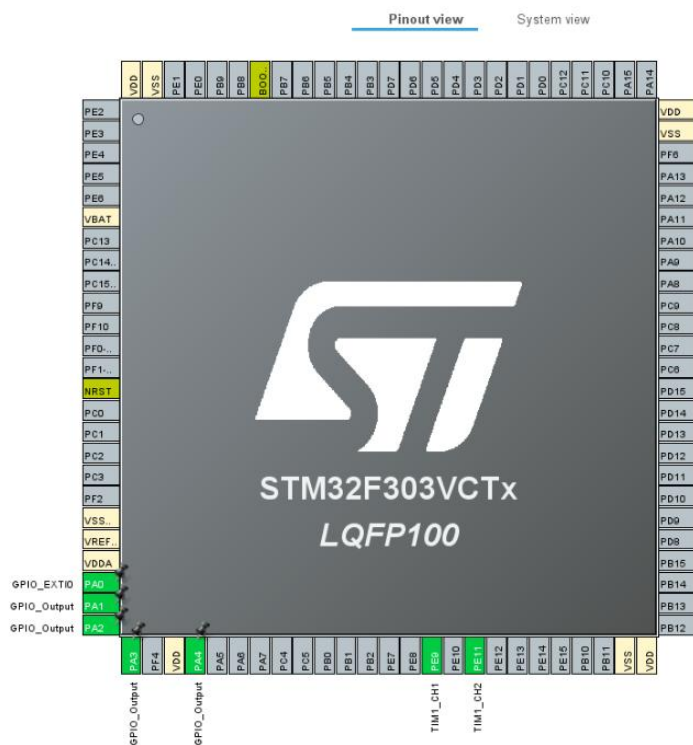
robota

Oliwier Bogdański, Kacper Szponar

4. Schemat elektroniczny robota



5. Oprogramowanie sterujące



Dodatkowo zaznaczyć należy EXTI line0 interrupt.



robota

Oliwier Bogdański, Kacper Szponar

```
328 /* Private define -----*/
330 /* USER CODE BEGIN PD */
340 volatile uint8_t robot_running = 0; // Zmienna globalna ustawiana w przerwaniu po naciśnięciu przycisku użytkownika (USER button).
350 /* USER CODE END PD */
360
370 /* Private macro -----*/
380
60 //Funkcja do uruchamiania robota w żądanym kierunku i prędkością za pomocą parametrów
61
62 void robot_drive(uint16_t pwm_right, uint8_t dir_right,
63                 uint16_t pwm_left, uint8_t dir_left)
64 {
65     __HAL_TIM_SET_COMPARE(&htim1, TIM_CHANNEL_1, pwm_right); // Ustawia wypełnienie PWM na kanale 1 (prawy silnik) - kontrola prędkości
66     __HAL_TIM_SET_COMPARE(&htim1, TIM_CHANNEL_2, pwm_left); // Ustawia wypełnienie PWM na kanale 2 (lewy silnik) - kontrola prędkości
67     // ustawianie odpowiednich stanów logicznych na wyjściach GPIO
68     // prawy silnik
69     if (dir_right) {
70         HAL_GPIO_WritePin(GPIOA, GPIO_PIN_4, GPIO_PIN_SET); // IN1
71         HAL_GPIO_WritePin(GPIOA, GPIO_PIN_1, GPIO_PIN_RESET); // IN2
72     } else {
73         HAL_GPIO_WritePin(GPIOA, GPIO_PIN_4, GPIO_PIN_RESET); // IN1
74         HAL_GPIO_WritePin(GPIOA, GPIO_PIN_1, GPIO_PIN_SET); // IN2
75     }
76
77     // lewy silnik
78     if (dir_left) {
79         HAL_GPIO_WritePin(GPIOA, GPIO_PIN_2, GPIO_PIN_SET); // IN3
80         HAL_GPIO_WritePin(GPIOA, GPIO_PIN_3, GPIO_PIN_RESET); // IN4
81     } else {
82         HAL_GPIO_WritePin(GPIOA, GPIO_PIN_2, GPIO_PIN_RESET); // IN3
83         HAL_GPIO_WritePin(GPIOA, GPIO_PIN_3, GPIO_PIN_SET); // IN4
84     }
85 }
86
87 // Funkcja zatrzymująca ruch robota. Ustawia wszystkie wyjścia GPIO NA 0 oraz wypełnienie PWM na 0 na obu silnikach
88 void robot_stop()
89 {
90     HAL_GPIO_WritePin(GPIOA, GPIO_PIN_4, GPIO_PIN_RESET);
91     HAL_GPIO_WritePin(GPIOA, GPIO_PIN_1, GPIO_PIN_RESET);
92     HAL_GPIO_WritePin(GPIOA, GPIO_PIN_2, GPIO_PIN_RESET);
93     HAL_GPIO_WritePin(GPIOA, GPIO_PIN_3, GPIO_PIN_RESET);
94     __HAL_TIM_SET_COMPARE(&htim1, TIM_CHANNEL_1, 0);
95     __HAL_TIM_SET_COMPARE(&htim1, TIM_CHANNEL_2, 0);
96 }
97
98
```



robota

Oliwier Bogdański, Kacper Szponar

```
100 void robot_sekwencja()  
101 {  
102     // Zmienne z wartościami wypełnienia pwm.  
103     const uint16_t pwm_r = 989; // Minimalnie niższe wypełnienie PWM dla prawego silnika, by skorygować tor jazdy  
104     const uint16_t pwm_l = 999; // Maksymalne wypełnienie PWM (pełna prędkość) dla lewego silnika  
105  
106     // 1. Prosto  
107     robot_drive(pwm_r, 1, pwm_l, 1);  
108     HAL_Delay(1000);  
109     robot_stop();  
110     HAL_Delay(2000);  
111  
112     // 2. Skręt w prawo (tylko lewy silnik)  
113     robot_drive(0, 1, pwm_l, 1);  
114     HAL_Delay(600);  
115     robot_stop();  
116     HAL_Delay(3000);  
117  
118     // 3. Prosto  
119     robot_drive(pwm_r, 1, pwm_l, 1);  
120     HAL_Delay(1000);  
121     robot_stop();  
122     HAL_Delay(2000);  
123  
124     // 4. Skręt w lewo (tylko prawy silnik)  
125     robot_drive(pwm_r, 1, 0, 1);  
126     HAL_Delay(600);  
127     robot_stop();  
128     HAL_Delay(3000);  
129  
130     // 5. Cofanie prosto  
131     robot_drive(pwm_r, 0, pwm_l, 0);  
132     HAL_Delay(1100);  
133     robot_stop();  
134     HAL_Delay(3000);  
135  
136     // 6. Obrót w miejscu (prawo przód, lewo tył)  
137     robot_drive(pwm_r, 1, pwm_l, 0);  
138     HAL_Delay(650);  
139     robot_stop();  
140     HAL_Delay(3000);  
141  
142     // 7. Krótki ruch do przodu  
143     robot_drive(pwm_r, 1, pwm_l, 1);  
144     HAL_Delay(600);  
145     robot_stop();  
146     HAL_Delay(2000);
```

```
147  
148     // 8. Obrót w miejscu (prawo tył, lewo przód)  
149     robot_drive(pwm_r, 1, pwm_l, 0);  
150     HAL_Delay(560);  
151     robot_stop();  
152     HAL_Delay(3000);  
153  
154     // 9. Skręt w miejscu w prawo (lewe koło do przodu)  
155     robot_drive(0, 0, pwm_l, 1);  
156     HAL_Delay(570);  
157     robot_stop();  
158     HAL_Delay(2000);  
159  
160     // 10. Cofanie z lekkim skretem  
161     robot_drive(pwm_r, 0, pwm_l - 200, 0);  
162     HAL_Delay(1620);  
163     robot_stop();  
164 }  
165
```



robota

Oliwier Bogdański, Kacper Szponar

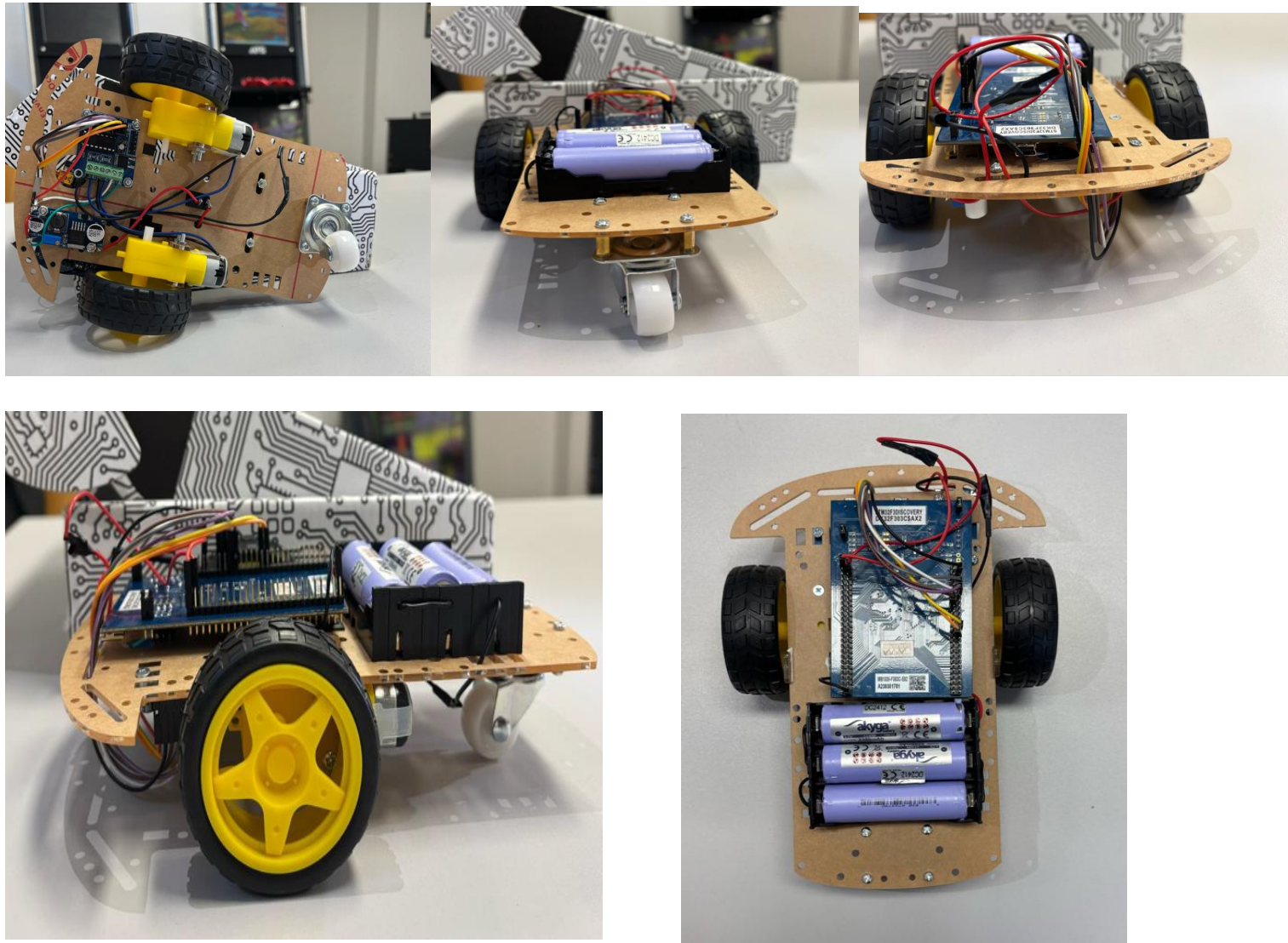
```
172 int main(void)
173 {
174
175     /* USER CODE BEGIN 1 */
176
177     /* USER CODE END 1 */
178
179     /* MCU Configuration-----*/
180
181     /* Reset of all peripherals, Initializes the Flash interface and the Systick. */
182     HAL_Init();
183
184     /* USER CODE BEGIN Init */
185
186     /* USER CODE END Init */
187
188     /* Configure the system clock */
189     SystemClock_Config();
190
191     /* USER CODE BEGIN SysInit */
192
193     /* USER CODE END SysInit */
194
195     /* Initialize all configured peripherals */
196     MX_GPIO_Init();
197     MX_TIM1_Init(); // Inicjalizacja timera 1
198     /* USER CODE BEGIN 2 */
199     HAL_TIM_PWM_Start(&htim1, TIM_CHANNEL_1); // Start generowania sygnału PWM na kanale 1 (prawy silnik)
200     HAL_TIM_PWM_Start(&htim1, TIM_CHANNEL_2); // Start generowania sygnału PWM na kanale 2 (lewy silnik)
201     /* USER CODE END 2 */
202     robot_stop();
203     /* Infinite loop */
204     /* USER CODE BEGIN WHILE */
205     while (1)
206     {
207         /* USER CODE END WHILE */
208
209         /* USER CODE BEGIN 3 */
210         if (robot_running) // Jeśli zostanie wciśnięty user button to sekwencja zostaje uruchomiona
211         {
212             HAL_Delay(1000);
213             robot_sekwencja();
214             robot_running = !robot_running;
215         }
216         else
217         {
218             robot_stop();
219         }
220     }
221     /* USER CODE END 3 */
222 }
391 /* USER CODE BEGIN 4 */
392 // Funkcja wywoływana przez funkcję obsługującą przerwanie (w pliku stm32fxx_it.c)
393 //
394 void HAL_GPIO_EXTI_Callback(uint16_t GPIO_Pin)
395 {
396     // Przełączenie trybu pracy robota po naciśnięciu przycisku (start/stop sekwencji).
397     if (GPIO_Pin == GPIO_PIN_0)
398     {
399         robot_running = !robot_running; // Przełączenie stanu zmiennej
400     }
401 }
402 /* USER CODE END 4 */
```




robota

Oliwier Bogdański, Kacper Szponar

6. Zdjęcia opracowanego robota



Film Youtube: <https://youtu.be/G137KO6tqbQ?si=3yl4mLEg99nJD2n>



AKADEMIA
NAUK STOSOWANYCH
w ELBLĄGU

Systemy
Wbudowane i Mikroprocesory
2024/2025

Raport z budowy

robota

Oliwier Bogdański, Kacper Szponar