

Mateusz GÓRNY¹

Opiekun naukowy: Jarosław JANUSZ²

PROJEKT I BUDOWA PROSTEGO POJAZDU ZDALNIE STEROWANEGO Z SYSTEMEM WIZYJNYM

Streszczenie: W pracy przedstawiono zakres prac niezbędnych do zbudowania prostego pojazdu sterowanego aparaturą radiową. Pojazd wyposażono w kamerę, która umożliwia przekazywanie obrazu rejestrowanego przez zdalnie sterowany pojazd.

Słowa kluczowe: sterowanie radiem, widok z pierwszej osoby, projekt, budowa

DESIGN AND CONSTRUCTION OF A REMOTE CONTROLLED VEHICLE WITH A VISION SYSTEM

Summary: The work presents the steps necessary to build a simple vehicle controlled by radio with a camera that allows a first-person view. The vehicle is equipped with a camera that allows you to upload images taken by a remotely operated vehicle.

Keywords: radio control, first person view, design, construction

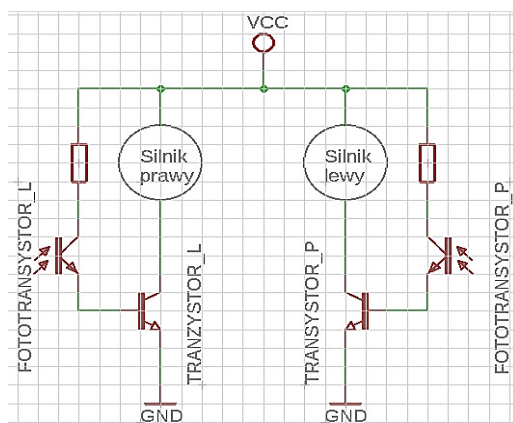
1. Wstęp

Pojazdy zdalnie sterowane mają szerokie zastosowanie w przemyśle do systemów kontroli elementów w obszarach trudno dostępnych dla ludzi. Pojazd zdalnie sterowany to jednostka, która ma możliwość przemieszczania się przy pomocy napędzanych kół, gąsienic lub wszelkiego rodzaju odnóży bardziej lub mniej zbliżonych do ludzkich nóg [3]. Sterowany jest przez użytkownika, który kontroluje pojazd dzięki odpowiedniej aparaturze (nadajnikowi). Nadajnik ten, zwykle posiada dedykowany odbiornik, który umieszcza się w pojeździe i podpina pod jego krytyczne części np.: sterownik, silnik. Oczywiście nie jest to jedyna możliwość. Poza profesjonalnymi rozwiązaniami istnieją też inne drogi. Dla przykładu robot

¹ student, Akademia Techniczno-Humanistyczna w Bielsku-Białej, Wydział Budowy Maszyn i Informatyki, Specjalność: Mechatronika i Robotyka, email: m-1971@tlen.pl

² dr inż. ATH, Akademia Techniczno-Humanistyczna w Bielsku-Białej, Wydział Budowy Maszyn i Informatyki, Katedra Podstaw Budowy Maszyn, Zakład Mechatroniki, email: jjanusz@ath.bielsko.pl

„Światłolub” jest prostym robotem, który dzięki fototranzystorom jest w stanie reagować na otoczenie i kierować się w stronę największego natężenia światła. W tym wypadku sterowanie odbywa się poprzez kontrolowanie natężenia światła a układ logiczny zbudowany z kilku elementów elektronicznych reaguje na te zmiany (Rys.1). Światło padające na fototranzystor „zwiększa” jego przepustowość co skutkuje podaniem większego sygnału na tranzystor. To natomiast powoduje, że tranzystor „otwiera” się mocniej czyli większy prąd przepływa przez niego oraz przez silnik co skutkuje większą mocą na wale silnika. Robot ten pomimo posiadania tylko dwóch silników czyli dwóch kół napędzanych potrafi skręcać. Jest tak, dlatego że skręt odbywa się dzięki różnej mocy na wale lewego i prawego silnika. „Światłolub” jest przykładem robota, który nie wymaga mikrokontrolera ani żadnej aparatury kontrolnej. Dzięki temu jest prosty do wykonania ale jego możliwości są bardzo ograniczone.

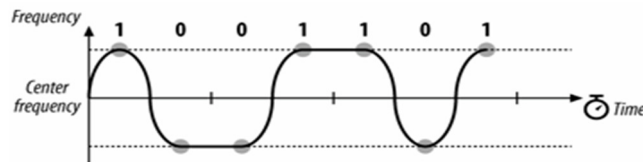


Rysunek 1. Schemat elektryczny robota podążającego za światłem

1.1. Aparatura RC

Sam skrót RC oznacza „Radio Control” czyli sterowanie w którym nośnikiem informacji jest fala elektromagnetyczna [4]. Układ składa się z nadajnika, którym jest pilot sterujący oraz odbiornika, który umieszczamy w pojeździe. Jest wiele czynników, które mogą powodować zakłócenia w takiej komunikacji. Niejednokrotnie jeden nadajnik może powodować zakłócenia dla drugiego nadajnika. By odbiornik mógł „zrozumieć” nadajnik wymagany jest pewien standard komunikacji. Istnieje wiele rodzajów np.: ASK „Amplitude Shift Keying” kluczkowanie amplitudy, QAM „Quadrature Amplitude Modulation”, FSK częstotliwościowe „Frequency-Shift Keying” i w tym następujące odmiany modulacji, które bazują na „istocie” FSK: MFSK “Multiple Frequency-Shift Keying” kluczkowanie wieloczęstotliwościowe, GFSK “Gaussian Frequency Shift Keying”, MSK „Minimum Shift Keying” kluczkowanie minimalnofazowe, GMSK “Gaussian Minimum Shift Keying” oraz PSK “phase-shift keying”. W dalszej części artykułu opisano tylko modulacją GFSK oraz FSK, gdyż to na niej oparta jest komunikacja w opisanym projekcie. GFSK jest to standard, który wykorzystuje fale o kształcie krzywej Gaussa. Logiczna „1” reprezentowana jest jako dodatnie odchylenie od

bazowej częstotliwości, „0” to natomiast ujemne odchylenie od częstotliwości bazowej (Rys. 2).



Rysunek 2. Idealne przedstawienie modulacji sygnału w metodzie GFSK [5]

W rzeczywistości dużo trudniej jest zaobserwować „zera” i „jedyne” gdyż modulacja częstotliwości to tak naprawdę zmiana długości bądź amplitudy fali. Modulację opartą na „odchyleniu” od wartości bazowej przedstawiono na rysunku 3. Można opisać ją wzorem:

$$s(t) = \sqrt{\frac{2E_s}{T}} \cos(2\pi f_c + \theta(t) + \theta_0) \quad (1)$$

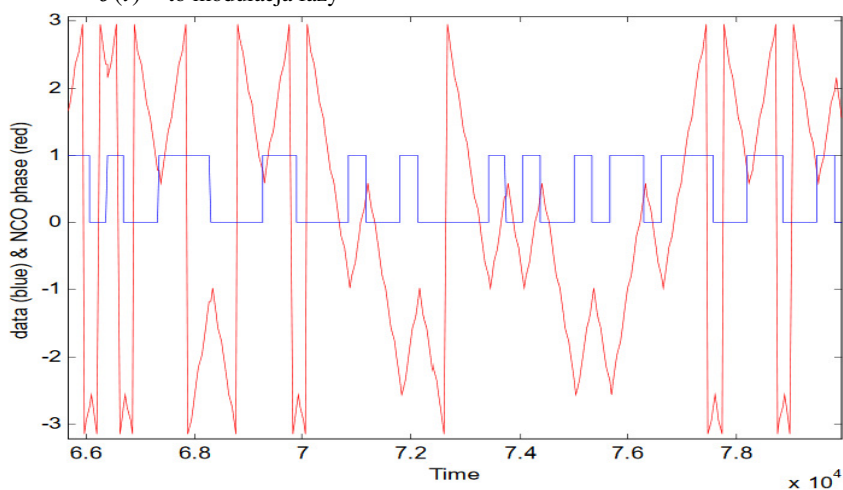
gdzie:

E_s - to niezbędna moc do wygenerowania jednego bitu danych

T -okres

f_c - bazowa częstotliwość

$\theta(t)$ - to modulacja fazy



Rysunek 3. Modulacja sygnału rzeczywistego (czerwony) wraz z nałożonym rysunkiem przedstawiający dane jakie „niesie” ten przebieg (niebieski) przy $f_c = 0$ [4]

Modulacja to nie wszystko. Dzięki niej nadajnik i odbiornik mają pewien „sposób” komunikacji. Dla przykładu: niech nadajnik i odbiornik są ludźmi. Dzięki modulacji oboje wiedzą, że teraz będą się komunikować mówiąc, niemniej jednak nadal pozostaje problem z samym przekazem gdyż istnieje wiele języków. By rozwiązać

ten problem należy stworzyć kolejny standard czyli odpowiednio zbudować ramkę danych jaka jest przekazywana.

1.2. FPV

FPV „First Person View” . Skrót ten oznacza widok z pierwszej osoby, czyli operator pojazdu RC ma możliwość podglądu tego co „widzi” jego pojazd. Jest to możliwe dzięki kamerze zabudowanej na pojeździe oraz nadajnika/odbiornika i urządzenie na którym zostanie wyświetlony obraz. Kamery analogowe mogą nagrywać filmy w różnych standardach. Główne standardy to NTSC, PAL oraz SECAM [6]. W dzisiejszych czasach analogowa telewizja praktycznie nie istnieje, a wymienione standardy zostają zastąpione przez standard telewizji cyfrowej np.: DVB-T, niemniej jednak nadal można zakupić kamery dla modeli RC, które pracują jednym z wymienionych standardów [6, 8].

2. Projekt i budowa pojazdu RC

Do budowy mobilnego robota zdalnie sterowanego drogą radiową i wyposażonego w FPV (First Person View) przyjęto podstawowe założenia:

- pojazd ma się poruszać po zróżnicowanym terenie (beton, piasek, ziemia),
- konstrukcja pojazdu ma zapewniać stabilność pojazdu, oraz możliwość manewrowania między przeszkodami,
- pojazd musi posiadać własne zasilanie (baterie) z możliwością szybkiej i łatwej wymiany,
- jednostką centralną układu sterowania ma być mikrokontroler,
- komunikacja między pojazdem a operatorem ma się odbywać drogą radiową z wykorzystaniem modelarskiej aparatury RC,
- pojazd ma być wyposażony w kamerę z komunikacją radiową.

Do zasilania pojazdu zastosowano akumulator litowo-polimerowy o pojemności 2200 mAh, 4S1P, 14.8V, 25C. Dobór źródła zasilania zależy przede wszystkim od przewidywanego poboru prądu. Najbardziej prądożernym elementem w robocie są oczywiście silniki. Zastosowane w pojeździe silniki prądu stałego 12VDC pobierają pod nominalnym obciążeniem 580mA każdy, przy czym maksymalny, bezpieczny prąd jaki może pobrać silnik wynosi 1A. Prąd zwarcia wynosi 4.8A.

Zastosowany akumulator ma wydajność prądową 55A co znacznie przewyższa zapotrzebowanie układu napędowego. Należy jednak pamiętać, że na pojeździe musi być zabudowany układ sterujący napędem, aparatura RC do komunikacji i kamera. Akumulatory li-pol są bardzo niebezpiecznymi akumulatorami. Przeładowanie akumulatora (powyżej 4,2V) skutkuje jego rozszczelnieniem i powoduje powstanie reakcji chemicznej, która kończy się pożarem. Przekroczenie dolnej granicy napięcia (3V) skutkuje uszkodzeniem akumulatora co objawia się zmniejszeniem jego pojemności.

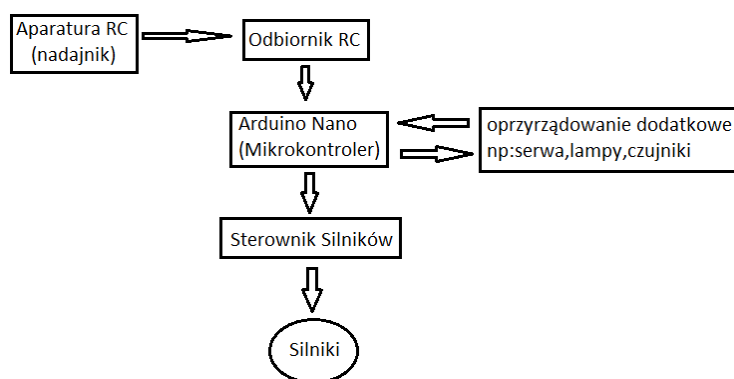
Jako konstrukcję nośną pojazdu wraz z silnikami napędzającymi zastosowano handlową wersję gąsienicowego podwozia mobilnego. Gąsienice są rozwiązaniem dla robotów, które mają poruszać się w trudnym terenie czyli na wszelkiego rodzaju gruncie który nie jest utwardzony. Duża powierzchnia zapewnia odpowiedni rozkład masy co utrudnia zapadnięcie się w podłożu.

Do projektu zastosowano kamerę FPV 1200TVL 150 Degree Super Small, nadajnik AV 5,8GHz Mini 600mW oraz odbiornik AV 5,8GHz BOSCAM RC805. Zarówno nadajnik jak i odbiornik pracują na paśmie 5,8GHz. Urządzenia te pracują w górnej granicy pasma, na które nie jest wymagane zezwolenie czyli 5725MHz. W Polsce jak i w innych krajach pasma radiowe podzielone są na pewnego rodzaju segmenty. Niektóre z nich dostępne są dla wszystkich bez konieczności posiadania zezwoleń od UKE (Urząd Komunikacji Elektronicznej) [7].

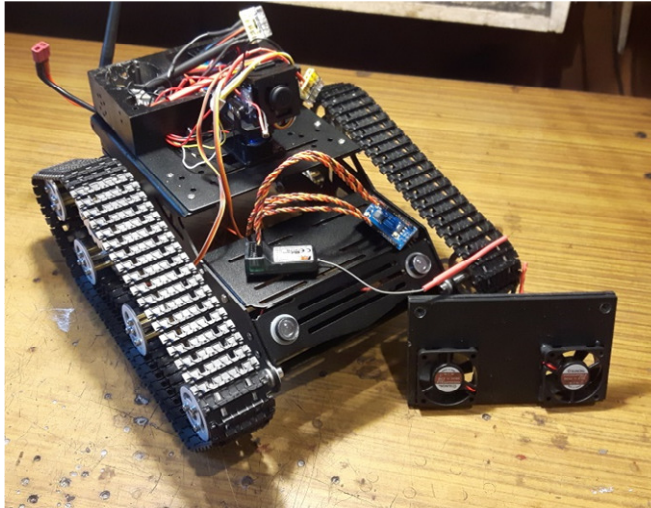
Dobierając aparaturę należy zwrócić uwagę na ile kanałów potrzebnych do komunikacji z pojazdem. Dla przykładu samolot, który posiada lotki, klapy, ster wysokości, ster kierunku oraz przepustnicę nie może być obsługiwany przez aparaturę, która posiada trzy kanały. Minimum to cztery kanały, przy czym nie uwzględniono dodatkowych elementów jak na przykład sterownie. Kolejny parametr to czy dana aparatura obsługuje telemetrię czyli czy jest w stanie odczytać prędkość, wysokość lub inne parametry z pojazdu. Następna przydatna informacja to czy dana aparatura może być podpięta do komputera co umożliwi nam użycie jej jako pilota w odpowiednich programach symulacyjnych oraz aktualizowanie oprogramowania. Do prezentowanego projektu wykorzystano aparaturę FlySky FS-T6.

Dobierając sterownik silników należy zwrócić uwagę na maksymalny prąd jaki będą pobierały zastosowane w projekcie silniki. To jest główne kryterium, poza nim możemy poszukać sterownika, który ma wbudowany stabilizator dla układów logicznych (na 5V). Dodatkowo można przyjrzeć się w jaki sposób odbywa się sterowanie tym regulatorem gdyż to nim będziemy decydować o kierunku obrotów silników za pomocą mikrokontrolera.

Jednostka centralna ma za zadanie scentralizowanie sterowania sygnałami w pojeździe. Do obsługi przez mikrokontroler jest: 6 kanałów sterowania od odbiornika RC, pomiar napięcia zasilania, sterowanie dwoma serwami, które obracają kamerkę, sterowanie oświetleniem. Podsumowując projekcie przewidziane jest użycie dziesięciu sygnałów, choć ta liczba wzrosła w późniejszych etapach budowy. Zagospodarowanie sześciu kanałów od odbiornika RC nie byłoby możliwe gdyż odbiornik ten jak i cała aparatura RC przystosowana jest do współpracy z serwami, dlatego też w projekcie mikrokontroler jest niezbędny (rys. 4). Wykorzystano układ Arduino Nano.



Rysunek 4. Schemat układu sterowania pojazdu



Rysunek 5. Widok zbudowanego pojazdu zdalnie sterowanego z kamerą

3. Program sterujący

Poniżej przedstawiono wybrane fragmenty kodu programu napisanego w języku C dla Arduino Nano, który umożliwia sterowanie zbudowanym przeze mnie robotem. Należy pamiętać, że kod ten w chwili pisania artykułu nie był finalnym kodem co skutkuje tym że w jego strukturze pojawiają się zbędne komendy.

```
//Informacje na temat przeznaczenia kanałów
//ch1=MA5 LEWO/PRAWO
//ch2=MA4 PRZÓD/TYŁ
//ch3=MA3 KAMERA GÓRA/DÓŁ
//ch4=MA2 KAMERA LEWO/PRAWO
//ch5=MA1 HAMULEC tzw. Rynczny!
//ch6=MA0 ŚWIATŁA ON/OFF
//A7=Pomiar napięcia na celi (gdy <3,6V to znaczy że bateria
pusta!)
//D4=Dioda sygnalizująca niskie napięcie
//Sterowanie silnikami
/*ENA-Kanał sterowania prędkością silnika 1(out1 i out2)
   ENB-Kanał sterowania prędkością silnika 2(out3 i out4)
   IN1/IN2-Sterowanie kierunkiem obrotów silnika 1(out1 i
out2)
   IN3/IN4-Sterowanie kierunkiem obrotów silnika 2(out3 i
out4)
   IN1/IN3 IN2/IN4   OPIS
   1     0   Silnik kręci się zgodnie z zegarem
   0     1   Silnik kręci się przeciwnie do zegara
   0     0   Szybkie hamowanie(gdy PWM=MAX)
   1     1   Szybkie hamowanie(gdy PWM=MAX)
   1     1   Swobodne hamowanie (gdy PWM=MIN)
```

```
        Czerwona kropka pod + (przy silniku) PRAWA GAŚNIENICA
MA SILNIK nr.2
*/
```

W języku stosowanym przez arduino komentarze rozpoczyna się poprzez „/” lub jeżeli chcemy za komentować więcej niż jedną linię to piszemy „/*...*/”[1, 2]. Powyższe komentarze są bardzo ważne, jest tak dlatego że program musi być dostosowany do sprzętu. Sprzęt w tym wypadku to robot. W powyższych komentarzach zapisano pod jakie piny arduino podpięto elementy wykonawcze takie jak między innymi wejścia sterownika silników oraz informacje o tym jak nim sterować.

```
#include <Servo.h> //Biblioteka odpowiedzialna za serwa
Servo  serwomechanizm_LP,serwomechanizm_GD; //Tworzymy
obiekt, dzięki któremu możemy odwołać się do serwa
//Deklaracje zmiennych dla sterownika silnika
int ENA = 5;//ENA-Enable A oraz PWM dla silnika A (D5)
int ENB = 6;//ENA-Enable B oraz PWM dla silnika B (D6)
int IN1 = 12;//Silnik A
int IN2 = 11;//Silnik A
int IN3 = 10;//Silnik B
int IN4 = 9;// Silnik B
//Zmienne dla PWM
int PWM_Lewo,PWM_Prawo,PWM_Przod,PWM_Tyl,PWM_Hamuj;//PWM dla
LewoPrawo,PrzódTył,Hamuj
//Deklaracja zmiennych
int ch1;
int ch2;
int ch3;
int ch4;
int ch5;
int ch6;
//ADC zmienne
int pomiar;
float napiecie;
//Zmienne dla kamerki
int pozycja_L,pozycja_P,pozycja_G,pozycja_D;
//Funkcje
//Kierunki jazdy
void lewo(int PWM_L)
{
    digitalWrite(IN1, 1);
    digitalWrite(IN2, 0);
    digitalWrite(IN3, 0);
    digitalWrite(IN4, 1);
    analogWrite(ENA, PWM_L);
    analogWrite(ENB, PWM_L);
}
void prawo(int PWM_P)
{
    digitalWrite(IN1, 0);
    digitalWrite(IN2, 1);
    digitalWrite(IN3, 1);
    digitalWrite(IN4, 0);
}
```

```
    analogWrite(ENA, PWM_P);
    analogWrite(ENB, PWM_P);
}
void przod(int PWM_Przod)
{
    digitalWrite(IN1, 0);
    digitalWrite(IN2, 1);
    digitalWrite(IN3, 0);
    digitalWrite(IN4, 1);
    analogWrite(ENA, PWM_Przod);
    analogWrite(ENB, PWM_Przod);
}
void tyl(int PWM_T)
{
    digitalWrite(IN1, 1);
    digitalWrite(IN2, 0);
    digitalWrite(IN3, 1);
    digitalWrite(IN4, 0);
    analogWrite(ENA, PWM_T);
    analogWrite(ENB, PWM_T);
}
void hamuj(int PWM_H)
{
    digitalWrite(IN1, 1);
    digitalWrite(IN2, 1);
    digitalWrite(IN3, 1);
    digitalWrite(IN4, 1);
    analogWrite(ENA, PWM_H);
    analogWrite(ENB, PWM_H);
}
void kamerkaL(int pozycja_L)
{
    serwomechanizm_LP.write(pozycja_L);
}
void kamerkaP(int pozycja_P)
{
    serwomechanizm_LP.write(pozycja_P);
}
void kamerkaG(int pozycja_G)
{
    serwomechanizm_GD.write(pozycja_G);
}
void kamerkaD(int pozycja_D)
{
    serwomechanizm_GD.write(pozycja_D);
}
```

Powyżej zapisano funkcje, które odpowiadają za sterowanie ruchami robota i kamerki. Zapis `digitalWrite(IN1, 1);` oznacza że wysterylujemy pin D12 (IN1=12) na HIGH czyli 1. Funkcje te zostały zbudowane na bazie diagramu dostarczonego wraz z sterownikiem silników.

```
//Konfiguracja I/O
void setup() {
```



```
//Do testów zainicjalizowanie komunikacji
Serial.begin(9600); //Ustawienie prędkości transmisji
//Wysterowanie pinów
pinMode(PD4, OUTPUT); //Dioda sygnalizująca niskie napięcie
pinMode(ENA, OUTPUT);
pinMode(ENB, OUTPUT);
pinMode(IN1, OUTPUT);
pinMode(IN2, OUTPUT);
pinMode(IN3, OUTPUT);
pinMode(IN4, OUTPUT);
pinMode(A0, INPUT);
pinMode(A1, INPUT);
pinMode(A2, INPUT);
pinMode(A3, INPUT);
pinMode(A4, INPUT);
pinMode(A5, INPUT);
// Serial.begin(9600);
serwomechanizm_LP.attach(PD2); //Serwomechanizm lewo/prawo
podłączony do pinu 2
serwomechanizm_GD.attach(PD3); //Serwomechanizm góra/dół
podłączony do pinu 3
}
```

W funkcji void setup() umieszczono kod którego zadaniem jest odpowiednie przygotowanie arduino. Dodatkowo uruchomiono uart (Serial.begin(9600);) by móc go potem wykorzystać w trakcie debugowania programu.

```
void loop()
{
  //Pobór danych od odbiornika
  ch1 = pulseIn(A5, HIGH, 35000);
  ch2 = pulseIn(A4, HIGH, 35000);
  ch3 = pulseIn(A3, HIGH, 35000);
  ch4 = pulseIn(A2, HIGH, 35000);
  ch5 = pulseIn(A1, HIGH, 35000);
  ch6 = pulseIn(A0, HIGH, 35000);
  //Skalowanie danych od odbiornika
  ch1 = map(ch1, 1000, 2000, -100, 100);
  ch2 = map(ch2, 1000, 2000, -100, 100);
  ch3 = map(ch3, 1000, 2000, -100, 100);
  ch4 = map(ch4, 1000, 2000, -100, 100);
  ch5 = map(ch5, 1000, 2000, -100, 100);
  ch6 = map(ch6, 1000, 2000, -100, 100);
}
```

Przedstawiona komenda PulseIn służy do pomiaru długości impulsu. Serwomechanizmami steruje się w taki sposób, że podaje się im sygnały o odpowiedniej długości. Długość sygnału decyduje o położeniu serwonapędu. Dzięki tej komendzie jestem w stanie przechwycić informacje jakie wysłałby odbiornik RC do serwonapędu gdyby był z nim połączony. Od teraz mogę wykorzystać te sygnały w jaki tylko chcę sposób, oto prawdziwa zaleta posiadania mikrokontrolera gdyż sygnał który jest modulowany w taki sposób by zrozumiał go serwonapęd zostanie przetworzony na prosty sygnał on/off, który jest zrozumiały dla sterownika silników. Komenda map służy do przemapowania zakresu sygnału wejściowego na odpowiedni

zakres wyjściowy. Wartości jakie wpisano w funkcji map zostały dobrane eksperymentalnie dzięki zainicjalizowaniu uart. Dzięki temu, że odczytywane są informacje z arduino można w prosty sposób zobaczyć w jakich granicach mieszczą się sygnały wysyłane przez odbiornik RC i przekształcić je na takie jakie są potrzebne.

```
//Obsługa pwm
//Kierunki Lewo/Prawo
PWM_Lewo = map(ch1, -121, 0, 255, 0);
if (PWM_Lewo<=15) PWM_Lewo=0;
if (PWM_Lewo>=250) PWM_Lewo=255;
PWM_Prawo = map(ch1, -5, 114, 0, 255);
if (PWM_Prawo<=15) PWM_Prawo=0;
if (PWM_Prawo>=250) PWM_Prawo=255;
//Kierunki Przód/Tył
PWM_Przod = map(ch2, -5, 114, 0, 255);
if (PWM_Przod<=15) PWM_Przod=0;
if (PWM_Przod>=250) PWM_Przod=255;
PWM_Tyl = map(ch2, -121, -5, 255, 0);
if (PWM_Tyl<=15) PWM_Tyl=0;
if (PWM_Tyl>=250) PWM_Tyl=255;
//HAMUJ!!!
PWM_HamuJ = map(ch5, 114, -102, 0, 255);
if (PWM_HamuJ<5) PWM_HamuJ=0; if (PWM_HamuJ>250) PWM_HamuJ=255;
//KAMERKA!!!
//LEWO PRAWO
pozycja_L = map(ch4, -121, 0, 90, 0);
pozycja_P = map(ch4, -5, 114, 90, 180);
//Test Akumulatora Lit-Pol
pomiar=analogRead(A7);
napiecie=pomiar*(5.0/1024.0)-0.25;
//Do testów wypis danych ze zmiennych
//Serial.println("Start testu:"); //Jednorazowe wysłanie
tekstu
//Serial.println(ch1);
//Serial.println(ch2);
Serial.println(ch3);
Serial.println("xxx");
Serial.println(ch4);
//Serial.println(ch5);
//Serial.println(ch6);
//Serial.println("PWM");
//Serial.println(PWM_Lewo);
//Serial.println(PWM_Prawo);
//Serial.println(PWM_Przod);
//Serial.println(PWM_Tyl);
//Serial.println(PWM_HamuJ);
//Serial.println(pomiar);
//Serial.println(napiecie);
//Odpowiedź na dane z odbiornika

if (ch3<-10)
{
  kameraL(pozycja_L);
}
if (ch3>10)
```

```
{
  kameraP (pozycja_P);
}

if (napiecie < 3.6)
{
  PWM_Hamuj = 255;
  hamuj (PWM_Hamuj);
  digitalWrite (PD4, HIGH);
}
else
{ //else start
  digitalWrite (PD4, LOW);
  //Obsługa Kamerki!
  if (ch3 >= -10 | ch3 <= 10)
  {
    serwomechanizm_GD.write (68); //Pozycja normalna kamery
  }
  if (ch4 >= -10 | ch4 <= 10)
  {
    serwomechanizm_LP.write (105); //Pozycja normalna kamery
  }
  //Koniec obsługi kamerki!
  if (ch5 < -290)
  {
    PWM_Hamuj = 255;
    hamuj (PWM_Hamuj);
  }
  else
  {
    //Hamowanie
    if (ch1 >= -10 | ch1 <= 10)
    {
      hamuj (PWM_Hamuj);
    }
    if (ch2 >= -10 | ch2 <= 10)
    {
      hamuj (PWM_Hamuj);
    }
    //Jazda Lewo/Prawo
    if (ch1 < -10)
    {
      lewo (PWM_Lewo);
    }
    if (ch1 > 10)
    {
      prawo (PWM_Prawo);
    }
    //Jazda Przód/Tył
    if (ch2 < -10)
    {
      tyl (PWM_Tyl);
    }
    if (ch2 > 10)
    {
```

```
        przod(PWM_Przod);  
    }  
}  
} //else stop(dla pomiaru ADC)  
delay(200);  
}
```

4. Podsumowanie

Przedstawiony w pracy tok postępowania pozwolił na zbudowanie prototypu pojazdu zdalnie sterowanego, wyposażonego w kamerę zapewniającą widok z pojazdu. Sterowanie pojazdem jest realizowane drogą radiową, a jednostką centralną jest mikrokontroler. Zbudowany pojazd jest wersją prototypową pozwalającą na dalszą rozbudowę i możliwość realizacji dodatkowych zadań specjalnych takich jak pomiar temperatury, wilgotności czy nawet skażenia kontrolowanego obszaru.

LITERATURA

1. MIROSŁAW K.: Mikrokontrolery AVR Język C podstawy programowania. Atmel, Szczecin 2013.
2. MIROSŁAW K.: Język C Pasja programowania mikrokontrolerów 8-bitowych. Atmel, Szczecin 2014.
3. DAVID C.: Budowa robotów dla średnio zaawansowanych. Helion, Gliwice 2013.
4. Serwis internetowy COMBLOCK – modulator FSK/MSK/GFSK/GMSK:
<https://www.comblock.com/download/com1028.pdf>, 9.9.2009
5. Serwis internetowy flylib.com - Gaussian Frequency Shift Keying (GFSK):
https://flylib.com/books/en/2.519.1/gaussian_frequency_shift_keying_gfsk_.html, 11.1.2018
6. Serwis internetowy sklep.delta.poznan.pl - CVBS - standard przesyłu sygnału wizyjnego: https://sklep.delta.poznan.pl/cvbs-standard-przesylu-sygnalu-wizyjnego_11_aid771.html, 13.4.2018
7. Serwis internetowy COMPUTERWORLD - Częstotliwości z licencją i bez:
<https://www.computerworld.pl/news/Czestotliwosci-z-licencja-i-bez,379225.html>, 17.5.2017
8. Serwis internetowy mvteamcctv.com - What is NTSC, PAL for Analog signal cameras?: <https://www.mvteamcctv.com/news/What-is-NTSC-PAL-for-Analog-signal-cameras.html>, 18.1.2018