

AGH, WIET	<b>Technika Mikroprocesorowa 2</b>	Kierunek: <b>EiT</b>
Temat: Projekt pojazdu sterowanego na pilot i unikającego kolizji		
Data wykonania: 22.01.2020	Imię i nazwisko: <ol style="list-style-type: none"> <li>1. Monika Lis</li> <li>2. Wojciech Przybyło</li> <li>3. Artur Nowak</li> </ol>	

## ***Cel projektu***

---

Głównym założeniem projektu było stworzenie pojazdu poruszającego się w dwóch trybach:

a) tryb sterowania za pomocą pilota przewodowego

W tym trybie pojazd porusza się do w zależności od tego który guzik na pilocie został wciśnięty. Pojazd może poruszać się do przodu, do tyłu, w prawo i w lewo. Dodany został piąty przycisk służący jako klakson.

b) tryb poruszania się do przodu do momentu wykrycia możliwości kolizji

W tym trybie pojazd się samodzielnie do przodu do momentu w którym sonar zasygnalizuje zdefiniowaną w oprogramowaniu przez użytkownika odległość. W naszym przykładzie ta odległość będzie wynosić 50 cm.

Pierwotnym projektem był pojazd podążający za dźwiękiem (np. klaskaniem) realizowany za pomocą macierzy kierunkowej Matrix Creator, jednakże ze względu na ograniczone możliwości czasowe i trudny kontakt z producentem wyżej wymienionej macierzy nie byliśmy w stanie uporać się z trudnościami z nią związanymi na czas oddania projektu.

## ***Użyte narzędzia projektowe***

---

Do wykonaniu powyższego projektu użyte zostały następujące narzędzia oprogramowania i sprzętu:

a) oprogramowania:

- Keil uVision 5
- STM32CubeMX
- Eclipse
- STM32 ST-LINK Utility

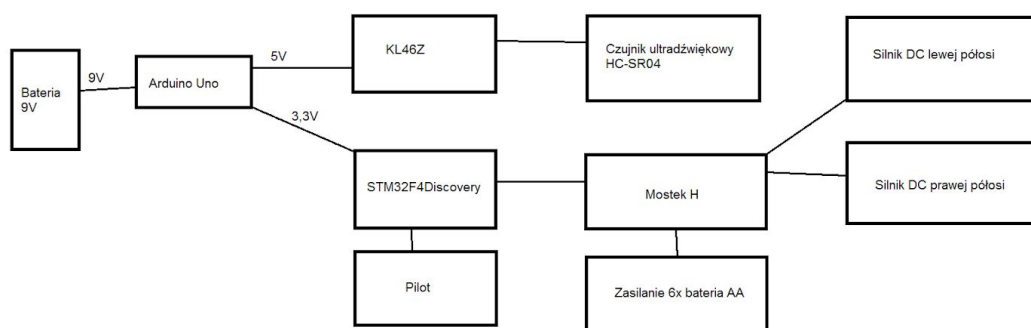
b) sprzętu:

- Dwa silniki DC
- Jedna bateria 9V
- Sześć baterii AA
- Mostek H - Układ L293D
- Sonar HC-SR04
- Mikrokontroler KL46Z
- Mikrokontroler STM32F4Discovery
- Mikrokontroler Arduino Uno - wykorzystane jedynie jako źródło zasilania dwóch powyższych mikrokontrolerów

## Wykonanie

---

Do przedstawienia działania i wykonania projektu posłużymy się niżej zamieszczonym schematem blokowym:



Rys. 1 Schemat blokowy projektu

### 1. Zasilanie logiki

Zasilanie logiki układu odbywa się za pomocą baterii 9V podłączonej do jednego z pinów Arduino Uno. Następnie z pinów Arduino 5V i 3,3V poprowadzono zasilanie do pinów zasilających logikę mikrokontrolerów KL46Z i STM32F4Discovery. Takie rozwiązanie pozwoliło nam uniknąć budowania układów stabilizujących napięcie i prąd. Co więcej przy potencjalnym spaleniu płytki ryzykowaliśmy utratę stosunkowo niedrogiego Arduino zamiast dużo droższych mikrokontrolerów KL46Z i STM32F4Discovery.

## 2. Działanie sonaru

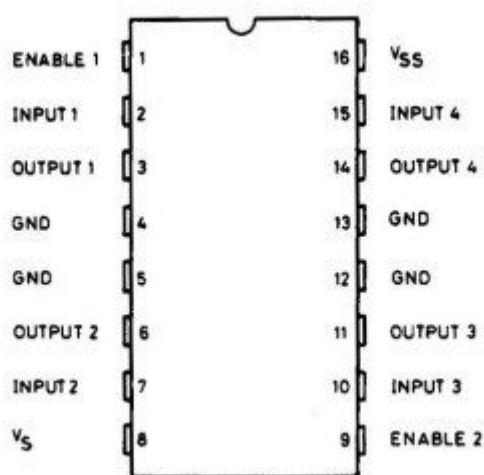
Działanie czujnika ultradźwiękowego jako sonaru zostało zrealizowane za pomocą mikrokontrolera KL46Z zgodnie z zamieszczoną instrukcją do laboratorium czwartego przedmiotu "Technika Mikroprocesorowa 2". Jedyną różnicą jest to, że sonar przy odczytaniu wykryciu odległości krótszej niż 50cm ustawia jeden z pinów GPIO, połączonego do mikrokontrolera STM, w stan wysoki. O tym czy sterowanie odbywa się za pomocą pilota czy sonaru decyduje użytkownik wciskając odpowiednio przełącznik na pilocie.

## 3. Działanie pilota

Działanie pilota zostało zrealizowane za pomocą przycisków dołączonych do płytki prototypowej, znajduje się na niej 5 przycisków odpowiadających za jazdę w przód, skręt w prawo, skręt w lewo, jazdę w tył oraz przycisk klaksonu. Są one połączone przewodowo z płytką STM na której odpowiednie piny GPIO ustawiane są w stan wysoki lub niski. O tym czy sterowanie odbywa się za pomocą pilota czy sonaru decyduje użytkownik wciskając odpowiednio przełącznik na pilocie.

## 4. Zasilanie i działanie silników

Za działanie silników odpowiadają dwa główne komponenty tj. zasilanie i sterowanie. Zasilanie realizowane jest za pomocą sześciu baterii AA podpiętych do mostka H. Schemat mostka zamieszczamy poniżej:



Rys. 2 Schemat wyprowadzeń układu L293D

Wejścia 1 i 9 decydują za ilość mocy doprowadzanej do każdego silników z osobna. W tym miejscu powinna zostać użyta przez nas technika modulacji szerokości impulsu PWM jednakże, jeśli stosowane są wartości inne niż maksymalne na tych pinach ze względu na mechanikę pojazdu, pojazd nie rusza z miejsca dlatego porzuciliśmy pomysł stosowania PWM na tych pinach i podawane są cały czas napięcia maksymalne.

Pary pinów 2 i 7 oraz 10 i 15 służą do sterowania kierunkiem obrotów silnika. Podanie wysokiego napięcia na pinie np. drugim i niskiego na pinie siódmym powoduje obrót silnika w jedną stronę a przy odwrotnym zasileniu pinów, silnik kręci się w stronę przeciwną.

Pary pinów 3 i 6 oraz 11 i 14 to piny z których podawane jest napięcie z pochodzące z pinu 8 do którego podpięte jest sześć baterii AA.

Piny 4 i 5 oraz 12 i 13 to piny masy, są ze sobą zwarte.

Pin 16 służy do zasilania logiki mostka.

## ***Problemy związane z projektem***

---

W trakcie realizacji projektu natknęliśmy się na kilka problemów, wymienione zostały one poniżej:

### **1. Manewrowość pojazdu i budowa stelażu**

Zdecydowanie największym problem okazała się być manewrowość pojazdu. Często zdarzało się, że bez względu na konfiguracją elektryczną i elektroniczną silników pojazd nie ruszał z miejsca albo nie był w stanie wykonać skrętu. Problem udało nam się rozwiązać za pomocą przebudowy stelażu i wyczyszczenia układu mechanizmu silników.

### **2. Problemy z zasilaniem**

Innym, znacznie mniejszym już, problemem okazało się być odpowiednie zasilanie płytek jak i pojazdu. Rozważaliśmy użycie stabilizatorów napięcia tj. LM7805, LM78T05 jednakże postanowiliśmy skorzystać z pinów zasilania i stabilizatorów scalonych zamontowanych w Arduino Uno.

### **3. Problem połączeń**

Kolejnym problemem okazała się duża ilość połączeń, które przy gwałtownych ruchach pojazdu miały tendencję do wypadania z pinów. Problem ten rozwiązaliśmy dużą ilością taśmy izolacyjnej