

Politechnika Wrocławska  
Wydział Elektroniki  
Automatyka i Robotyka  
Wizualizacja danych sensorycznych - projekt

# WIZUALIZACJA DANYCH Z CZUJNIKÓW LINE FOLLOWERA

*Autorzy:*  
*Beata Berajter 218629*  
*Ada Weiss 218641*

*Prowadzący:*  
*dr inż. Bogdan Kreczmer*

19 czerwca 2017

# Spis treści

<b>1</b>	<b>Cel projektu</b>	<b>2</b>
<b>2</b>	<b>Założenie projektowe</b>	<b>2</b>
<b>3</b>	<b>Harmonogram</b>	<b>2</b>
3.1	Dekompozycja projektu na zadania . . . . .	2
3.2	Kamienie milowe . . . . .	2
<b>4</b>	<b>Komunikacja z urządzeniem</b>	<b>3</b>
4.1	Odczyt danych pomiarowych . . . . .	3
4.2	Ramka danych . . . . .	4
<b>5</b>	<b>Opis działania programu</b>	<b>4</b>
5.1	Rozmieszczenie obiektów . . . . .	4
5.2	Diagram klas . . . . .	4
5.3	Diagram czynności . . . . .	4
<b>6</b>	<b>Opis poszczególnych elementów interfejsu graficznego</b>	<b>7</b>
6.1	Dane z czujników optycznych . . . . .	7
6.2	Dane z enkoderów . . . . .	7
6.3	Rysowanie trasy . . . . .	8
6.4	Informacja o danych sensorycznych w wybranym punkcie . . . . .	8
<b>7</b>	<b>Opis urządzenia</b>	<b>8</b>
7.1	Schematy elektroniczne robota . . . . .	9
7.1.1	Czujniki . . . . .	9
7.1.2	Stabilizator . . . . .	9
7.1.3	Mostek H . . . . .	9
7.1.4	Mikrokontroler . . . . .	9
7.1.5	Płytki PCB . . . . .	9
<b>8</b>	<b>Integracja, testowanie i poprawki projektu</b>	<b>13</b>
<b>9</b>	<b>Dokumentacja</b>	<b>13</b>
<b>10</b>	<b>Podsumowanie</b>	<b>13</b>

# 1 Cel projektu

Celem projektu jest zapoznanie się z metodami wizualizacji danych pochodzących z sensorów robota.

## 2 Założenie projektowe

Założeniem projektu jest zebranie danych pobranych z czujników line followera. Czujniki, z których należy pobrać informacje to czujniki optyczne oraz enkodery. Dane zbierane są w celu ich wizualizacji. W projekcie zostanie umieszczona animacja, która pokaże rozmieszczenie czujników, oraz w przypadku transoptorów odbiciowych pokaże, który z nich aktualnie wykrył linię. Dane z czujników będą przetwarzane na wykresy oraz ilustracje "słupkowe". Narysowana została również droga/ścieżka, którą podąża robot.

## 3 Harmonogram

### 3.1 Dekompozycja projektu na zadania

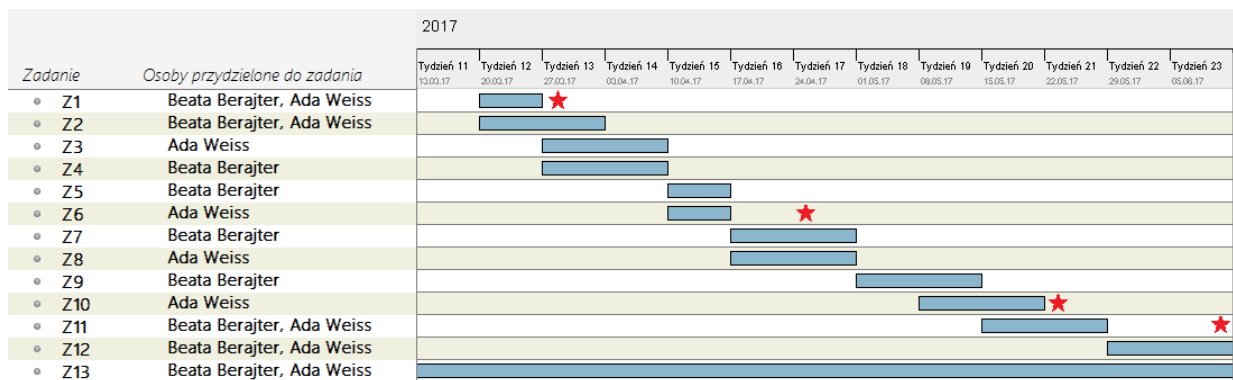
Wydzielono następujące zadania do wykonania w czasie realizacji projektu:

- Z1 - Określenie harmonogramu i podziału zadań
- Z2 - Zapoznanie ze środowiskiem QT
- Z3 - Projekt okienek programu
- Z4 - Opracowanie metody wczytywania danych w celu zastosowania jej do momentu przygotowania hardware'u
- Z5 - Opracowanie przedstawienia danych z czujników optycznych
- Z6 - Opracowanie przedstawiania danych słupkowych z enkoderów
- Z7 - Projekt wykresów do przedstawiania danych historycznych
- Z8 - Stworzenie metody obliczającej na podstawie danych z enkoderów położenie aktualne robota
- Z9 - Zaprojektowanie wizualizacji trasy robota
- Z10 - Zaprojektowanie dodatkowego okna do przedstawiania danych historycznych w wybranym punkcie
- Z11 - Integracja z działającym robotem
- Z12 - Testowanie programu
- Z13 - Tworzenie dokumentacji projektu.

### 3.2 Kamienie milowe

Kolejne kamienie milowe zostały oznaczone na rys. 1. czerwonymi gwiazdkami.

- Wybór tematu projektu : Data 19.03.2017 r.  
Do niedzieli 19 marca należało wybrać temat projektu oraz zrobić wstępny opis, w którym ogólnie trzeba było określić cel zadania.



Rysunek 1: Diagram Gantta wraz z przydzieleniem osób do zadań

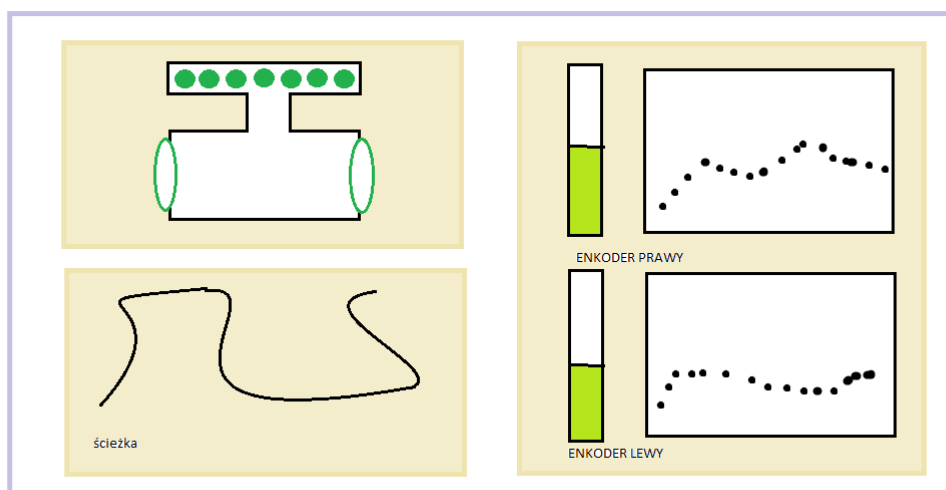
- Opis zadania i harmonogram : Data 26.03.2017 r.  
W tym terminie należało szczegółowo opisać założenia projektu, podział zadań oraz terminy, w których poszczególne elementy powinny być wykonane.
- Wstępne rezultaty : Data 23.04.2017 r.  
W raporcie przedstawiono wyniki dotychczasowych prac nad projektem wraz z opisem co zostało zrobione bądź nie i dlaczego.
- Rezultaty prawie końcowe : Data 14.05.2017 r.  
Sprawozdanie należało uzupełnić o wykonane zadania oraz przedstawić efekty prac.
- Oddanie projektu : 19.06.2017 r.  
Jest to końcowy termin oddania projektu do oceny.

## 4 Komunikacja z urządzeniem

Line follower jest realizowany w ramach projektu z robotów mobilnych w tym semestrze. Do momentu gdy robot będzie nieukończony posługujemy się zastępczym urządzeniem symulującym, które będzie wysyłało dane do komputera. Ponieważ komunikacja przez bluetooth jest podobna jak w przypadku interfejsu szeregowego do symulacji wykorzystamy płytkę STM32L476G Discovery podłączoną do portu USB komputera. Wizualizowane dane pochodzą z 7 czujników optycznych i 2 enkoderów. Czujniki optyczne informują o wykryciu zmiany koloru - linii. Na ten sygnał robot reaguje zmianą kierunku jazdy. Sygnał z czujników jest przekazywany wprost do pinów ADC mikrokontrolera. W mikrokontrolerze informacje zostają przetworzone na formę 0-1. W takiej postaci zostaną przesyłane przez moduł bluetooth do komputera (docelowo, w tym momencie przez USB)

### 4.1 Odczyt danych pomiarowych

Odczyt informacji z przesłanych z mikrokontrolera do programu realizowany jest poprzez czytanie danych z portu szeregowego komputera. W tym celu należało dodać



Rysunek 2: Rozmieszczenie widgetów na oknie głównym

użytkownika do grupy dialout. Pojawiające się urządzenie jest widoczne w `/dev` jako `ttyACMx`, gdzie `x` jest liczbą większą równą 0.

## 4.2 Ramka danych

Dane wysyłane będą poprzez moduł bluetooth HC-05 jako string w postaci:

X	7*cz. optyczne(0 lub 1)	2*enkoderyw	suma kontrolna	\r	\n
---	-------------------------	-------------	----------------	----	----

Dane z enkoderów będą reprezentowane przez dwie wartości w systemie szesnastkowym. Suma kontrolna będzie obliczana jako suma odczytów czujników modulo  $2^8$ .

# 5 Opis działania programu

## 5.1 Rozmieszczenie obiektów

Obiekty zostały rozmieszczone w sposób przedstawiony na rys. 2, by można było łatwo odczytywać dane. Wizualizacje dotyczące enkoderów pogrupowano w jednym miejscu. Dzięki temu nie trzeba szukać powiązanych ze sobą informacji po całym ekranie.

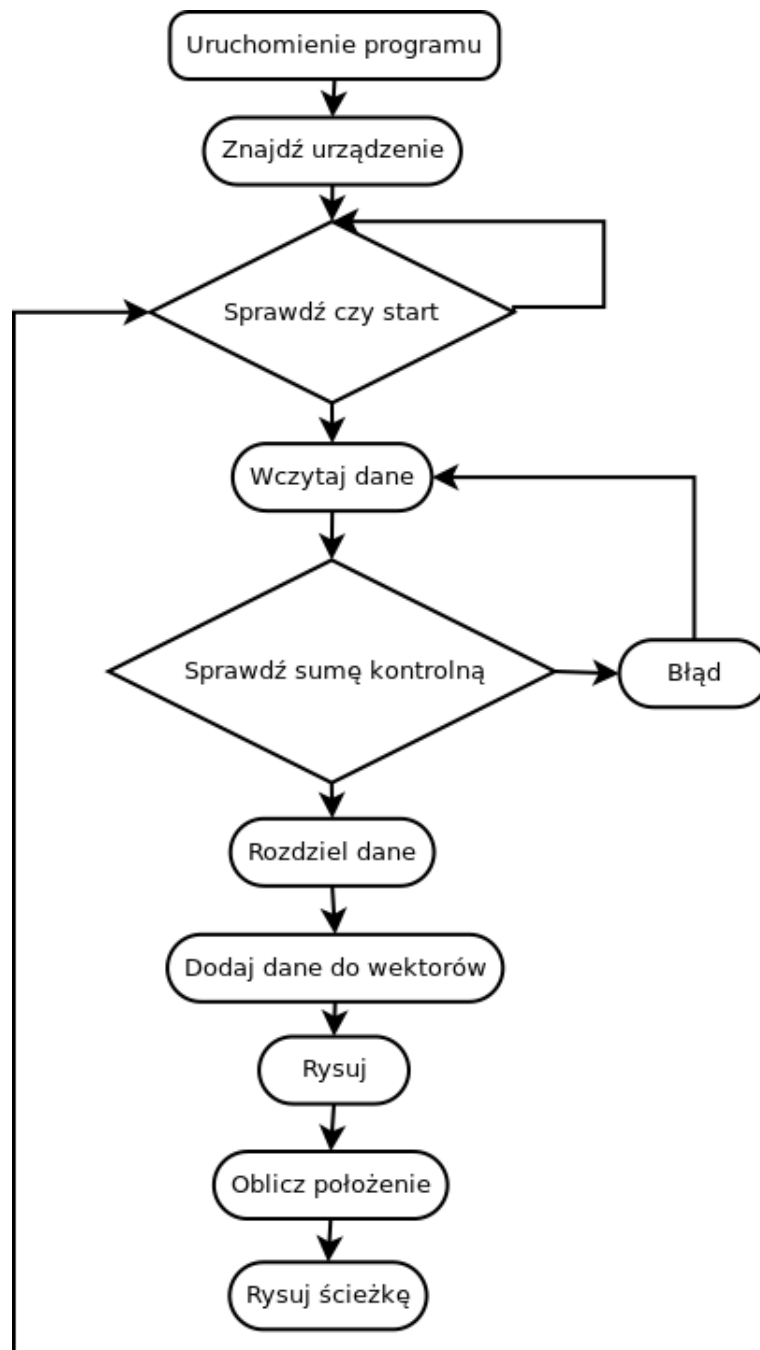
## 5.2 Diagram klas

Diagram klas w obecnej fazie projektu prezentuje się jak pokazano na rys. 3

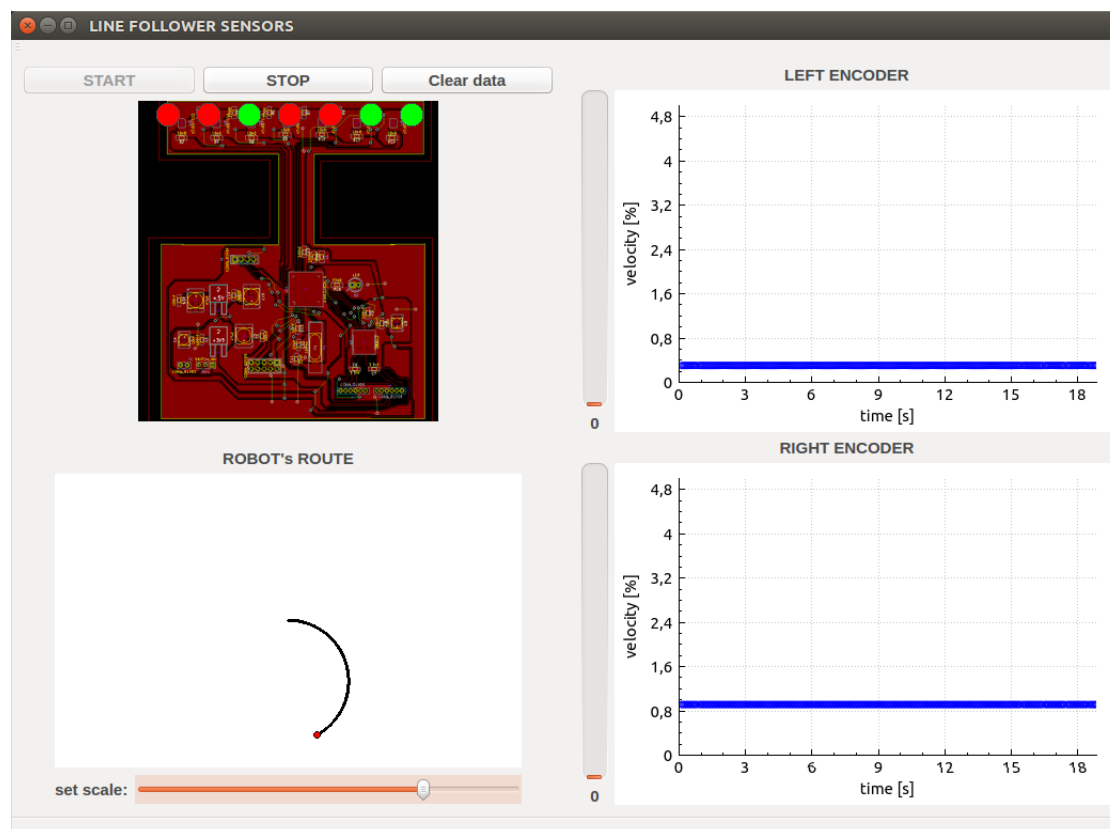
## 5.3 Diagram czynności

Uproszczony model działania programu realizujący założenia projektowe przedstawia 4. Ma on jedynie w sposób przybliżony opisać funkcjonowanie aplikacji.





Rysunek 4: Schemat działania programu



Rysunek 5: Okno główne programu

## 6 Opis poszczególnych elementów interfejsu graficznego

Interfejs graficzny programu (okno główne) został wykonany przy użyciu bibliotek Qt oraz QtDesignera i jest przedstawiony na rys. 5. Dalszy opis poszczególnych elementów odnosić się będzie do tego rys.

### 6.1 Dane z czujników optycznych

Czujniki optyczne zostaną przedstawione w postaci kółek zielonych (1 - aktywne) lub czerwonych (0 - nieaktywne) znajdujących się w odpowiednich miejscach na rys. płytki PCB.

### 6.2 Dane z enkoderów

Dane z enkoderów przedstawiane są w postaci słupków (aktualne) oraz wykresów (historyczne) znajdujących się w prawej części okna. Dane historyczne są gromadzone od wciśnięcia przycisku START do wciśnięcia przycisku Clear data, który je usuwa.

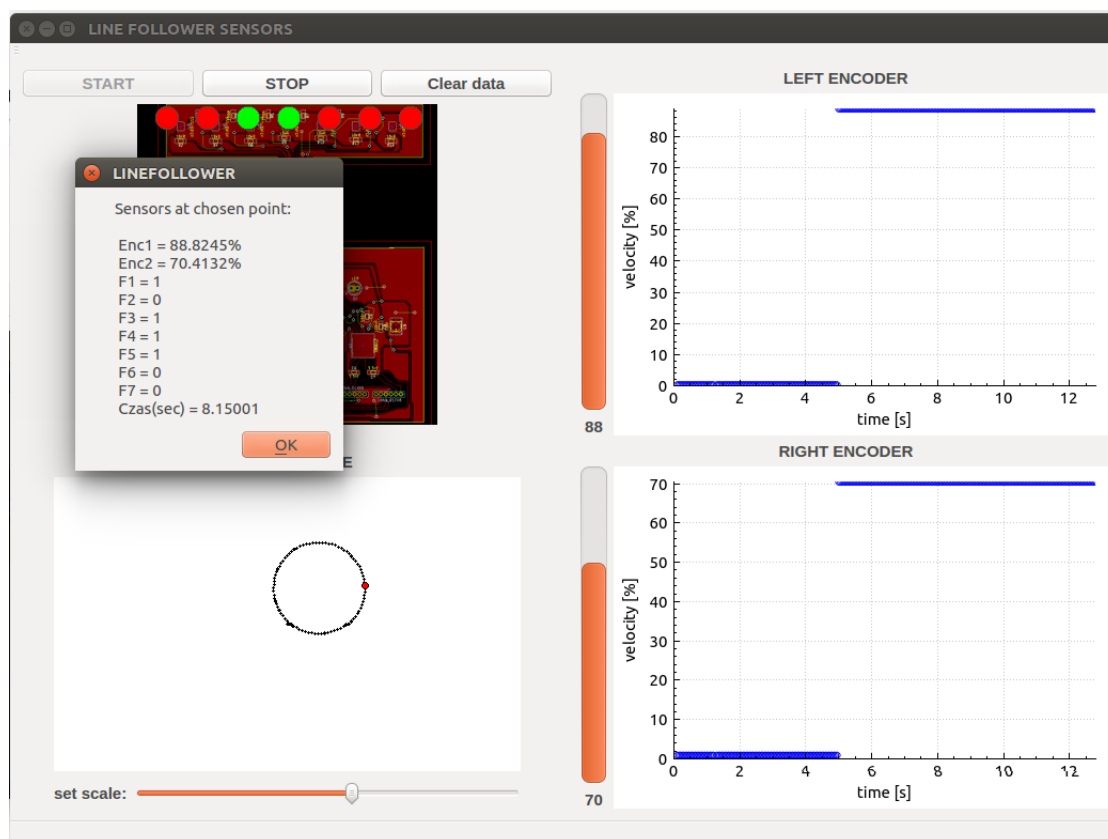


### 6.3 Rysowanie trasy

Dane z enkoderów zostały także użyte w funkcji obliczającej położenie robota ( $x$ ,  $y$  oraz jego orientację), dzięki czemu będzie można narysować ścieżkę, którą robot podąża. Możliwe jest również skalowanie ścieżki. Dane o położeniu są gromadzone, aby możliwe było wyświetlenie danych sensorycznych w wybranym punkcie trasy.

### 6.4 Informacja o danych sensorycznych w wybranym punkcie

Kliknięcie na wybrany punkt trasy robota na polu ROBOT'S ROUTE umożliwia wyświetlenie informacji o danych sensorycznych w danym momencie na trasie co przedstawia rys. 6



Rysunek 6: Okno główne programu z wyświetlonym dodatkowym okienkiem o informacji na temat danych w konkretnym punkcie

## 7 Opis urządzenia

Robot jest wyposażony w:

- 7 transoptorów odbiciowych, z których sygnał będzie przetwarzany przez przetwornik analogowo-cyfrowy mikrokontrolera,
- 2 miniaturowe enkodery dla silników Pololu 3.3V

- łącze bluetooth, które umożliwi wysyłanie danych z powyższych czujników do komputera,
- pozostałe elementy interfejsu sprzętowego: mikroprocesor ATmega32, układ zasilający, stabilizator, mostek h, programator.

## 7.1 Schematy elektroniczne robota

### 7.1.1 Czujniki

Jak zostało pokazane na schemacie (rys. 7.) czujniki optyczne zostały bezpośrednio dołączone do pinów ADC mikrokontrolera.

### 7.1.2 Stabilizator

W trakcie projektowania płytki wynikła konieczność użycia dwóch stabilizatorów na 5V i 3V3. Stało się tak, ponieważ silniki zasilane będą nie bezpośrednio z akumulatora, jak zakładano, ale poprzez enkodery. Zostaną one podłączone jak na rys. 8.

### 7.1.3 Mostek H

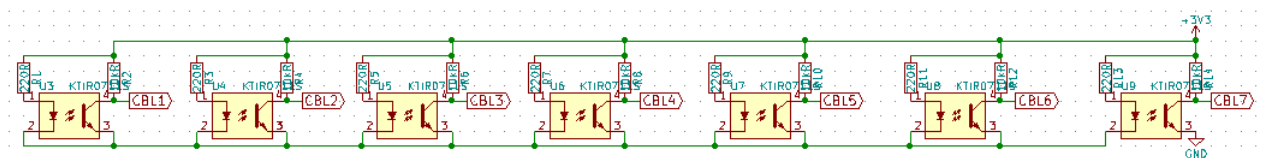
Na schemacie (rys. 9.) zostało pokazane podłączenie obu silników poprzez mostek H i enkodery.

### 7.1.4 Mikrokontroler

Jeden z najważniejszych elementów - Atmega32. Do niej dochodzą sygnały odebrane z czujników (CBL1-7) oraz enkoderów (E1-4). Połączona jest także z modulem bluetooth (J5), z niej wychodzą sygnały PWM sterujące silnikiem. Schemat został przedstawiony na rys. 10.

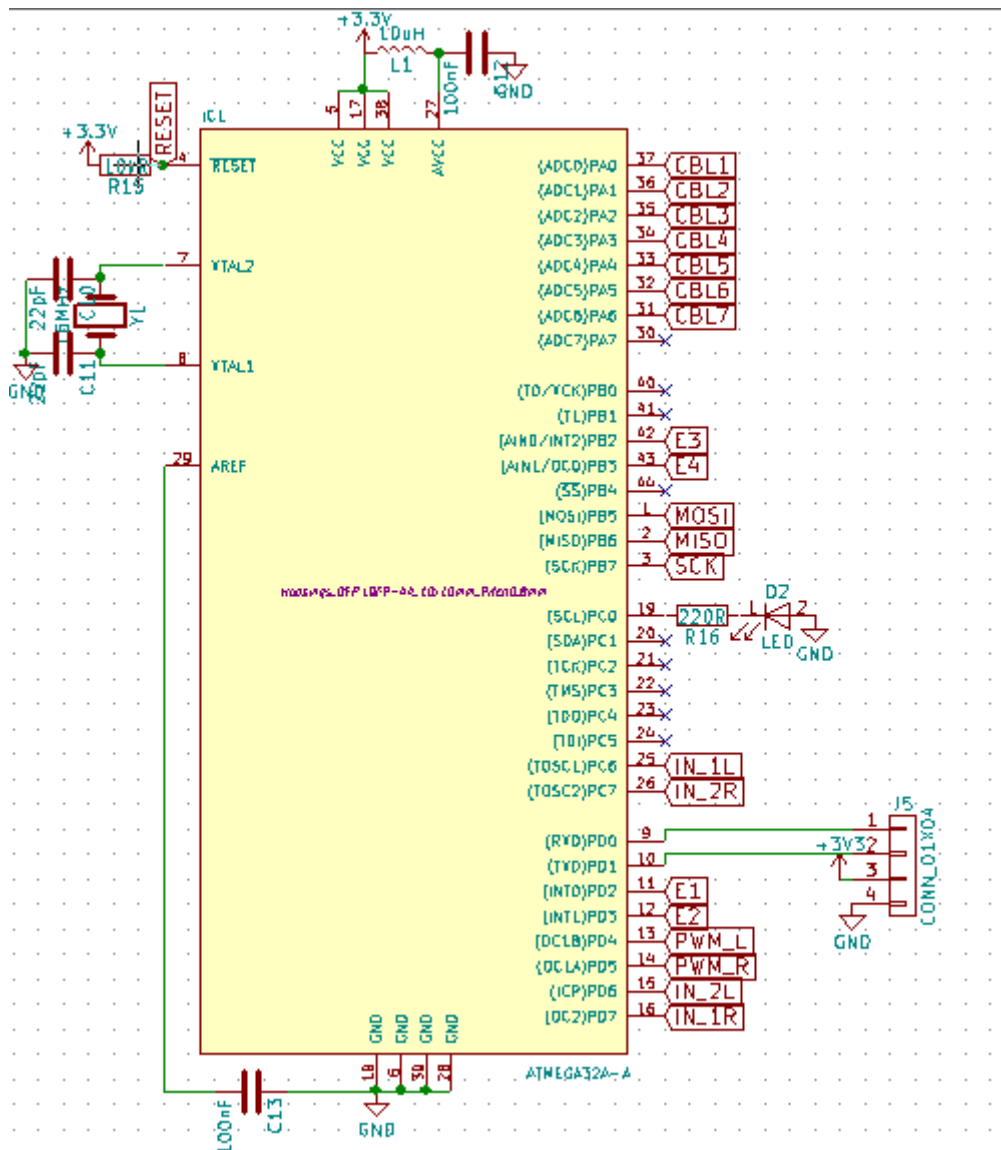
### 7.1.5 Płytką PCB

Zaprojektowana została płytką PCB (rys. 11.). Ma ona wymiary 10cm x 16cm. Przy jej rysowaniu nieocenioną pomocą okazał się tutorial na youtube.com.



Rysunek 7: Schemat połączenia czujników





Rysunek 10: Schemat podłączenia mikroprocesora



## 8 Integracja, testowanie i poprawki projektu

Ze względu na opóźnienie w budowie robota nie udało się jeszcze zintegrować robota z oprogramowaniem. Wymagałoby to drobnych modyfikacji, np. w rysowaniu trasy (modyfikacja skali). Do celów testowania mikrokontroler został zaprogramowany tak, że generuje losowe dane z czujników optycznych oraz po naciśnięciu przycisku zmieniana jest prędkość na inną losowo wygenerowaną dla każdego z kół.

## 9 Dokumentacja

Dokumentację projektu stanowi dokument html wygenerowany za pomocą Doxygena, który został dostarczony wraz ze sprawozdaniem oraz kodem źródłowym programu.

## 10 Podsumowanie

Udało się wykonać wszystkie zaplanowane zadania z wyjątkiem integracji z robotem. Wybrana metoda symulacji robota powinna jednak być na tyle skuteczna, że integracja robota z oprogramowaniem powinna przebiec pomyślnie.