Podstawy teorii mikroprocesorowej	Piątek TN 7:30-9:00	Prowadzący: Mgr. Inż. Maciej Filiński
Wykonał: Leonard Feszczuk	Data oddania: 01.06.2020	

Projekt:

Autonomiczny robot do walk „minisumo”

1. ## Wprowadzenie

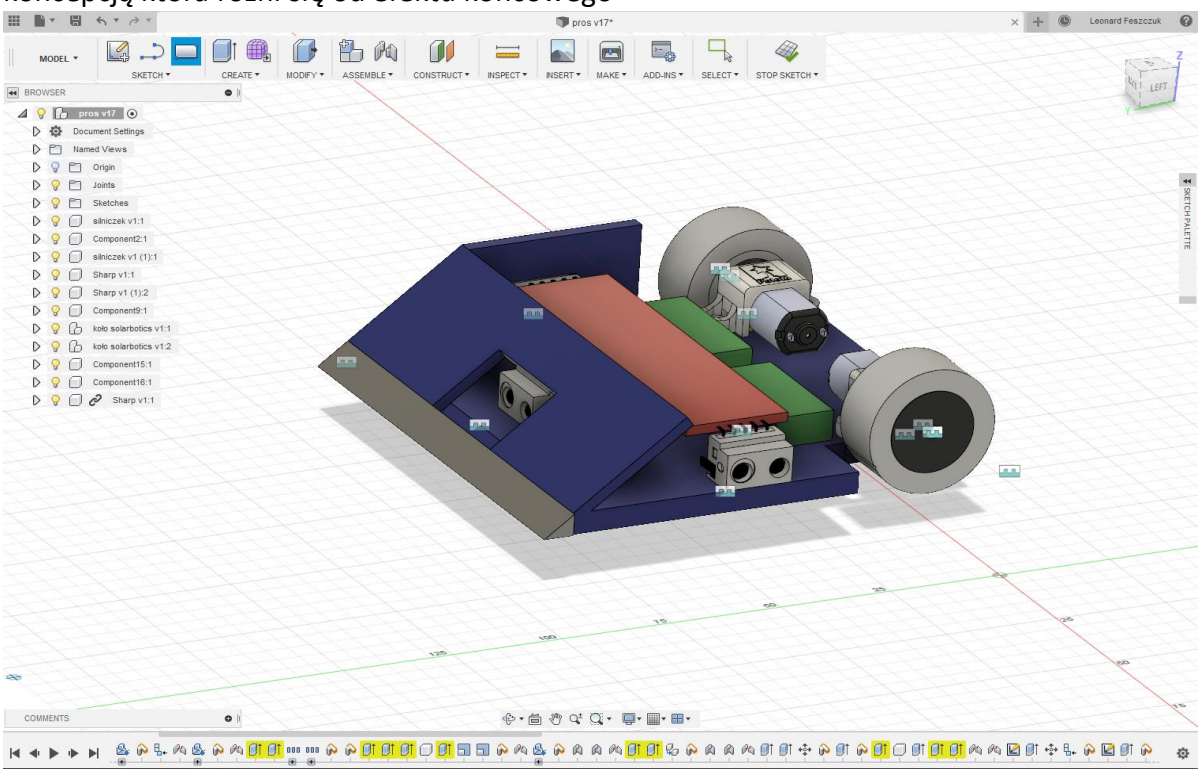
   Walki minisumo rozgrywają się na kolistej arenie o średnicy 150 cm. Robot musi być w pełni autonomiczny oprócz momentu uruchomienia oraz zatrzymania. Waga nie może przekraczać 500g a jego obrys musi się mieścić w kwadracie 10x10 cm.

   Na Politechnice wrocławskiej co roku organizowane są zawody robotów w tej i innych konkurencjach,robotic arena organizowana jest przez koło naukowe KONAR.

2. ## Wykonanie

   a. ### Model 3D

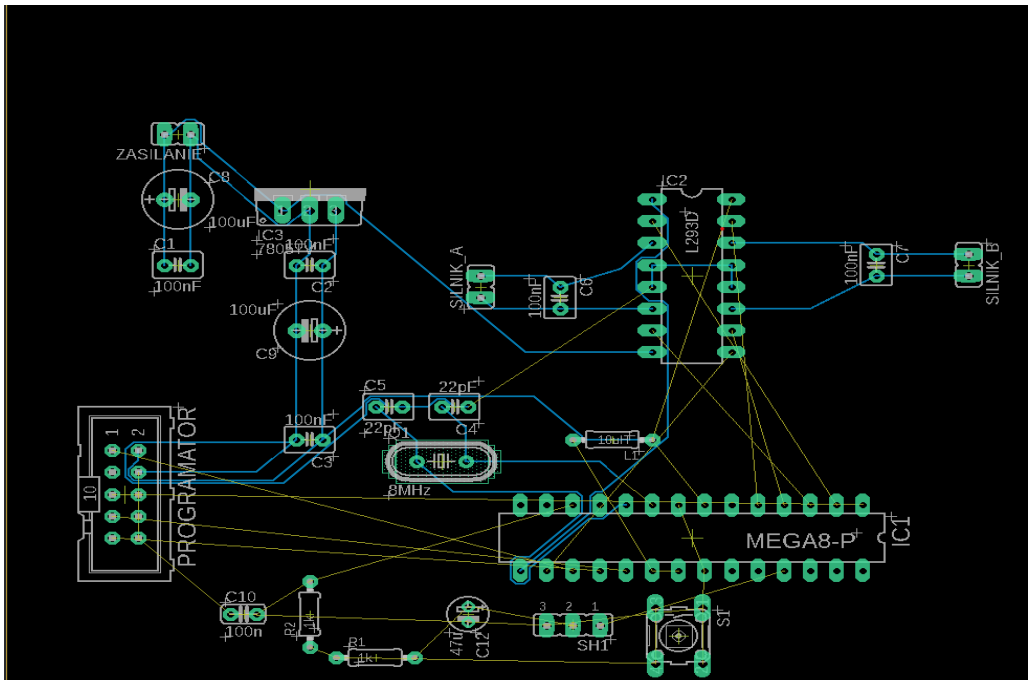
   Pracę nad robotem zacząłem, rok temu podczas warsztatów koła KONAR na których prezentowana była praca z programem Autodesk Fusion 360. Wykonany model był wstępną koncepcją która różni się od efektu końcowego



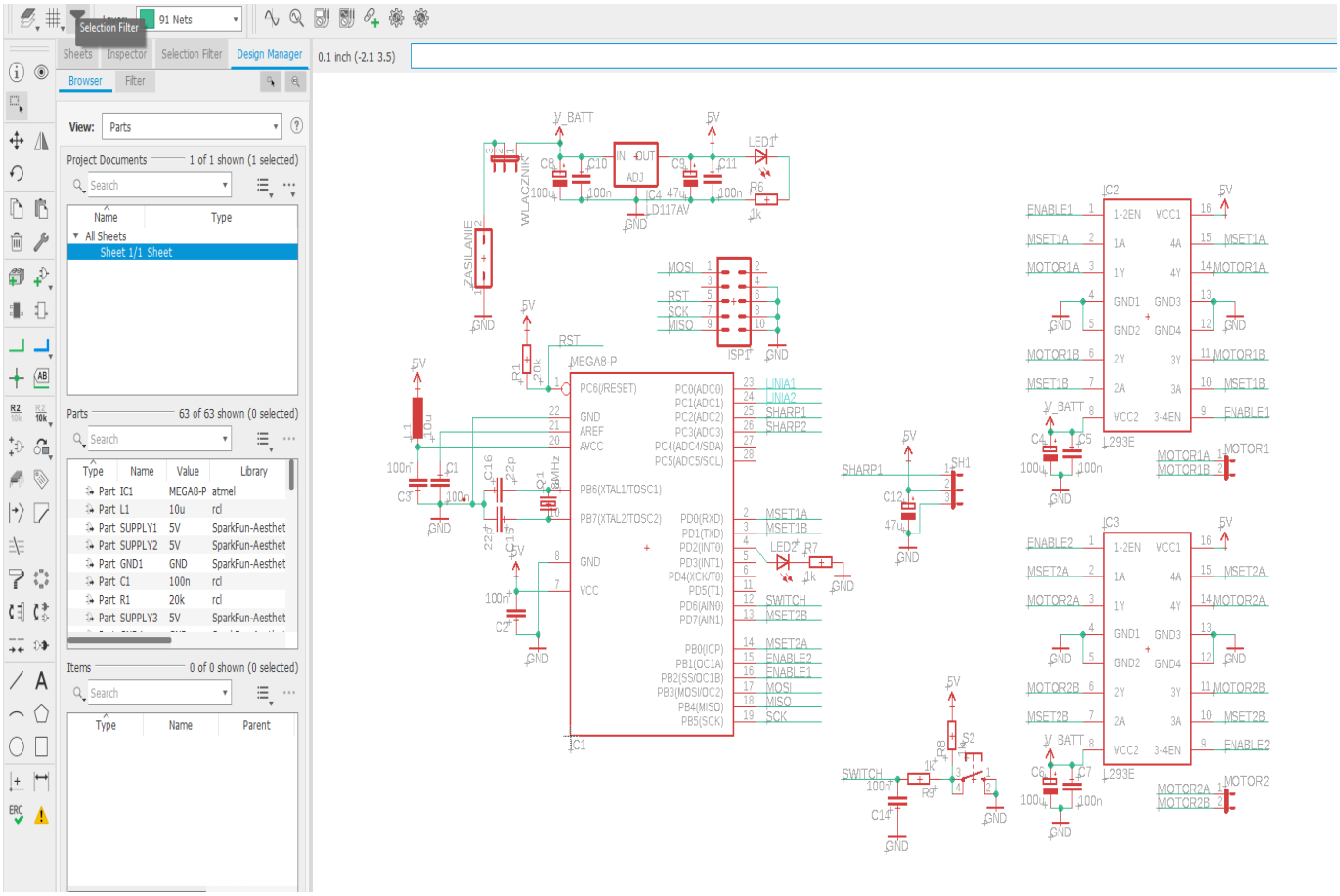
Rys. 2.a.1

b. Projekt płytki PCB

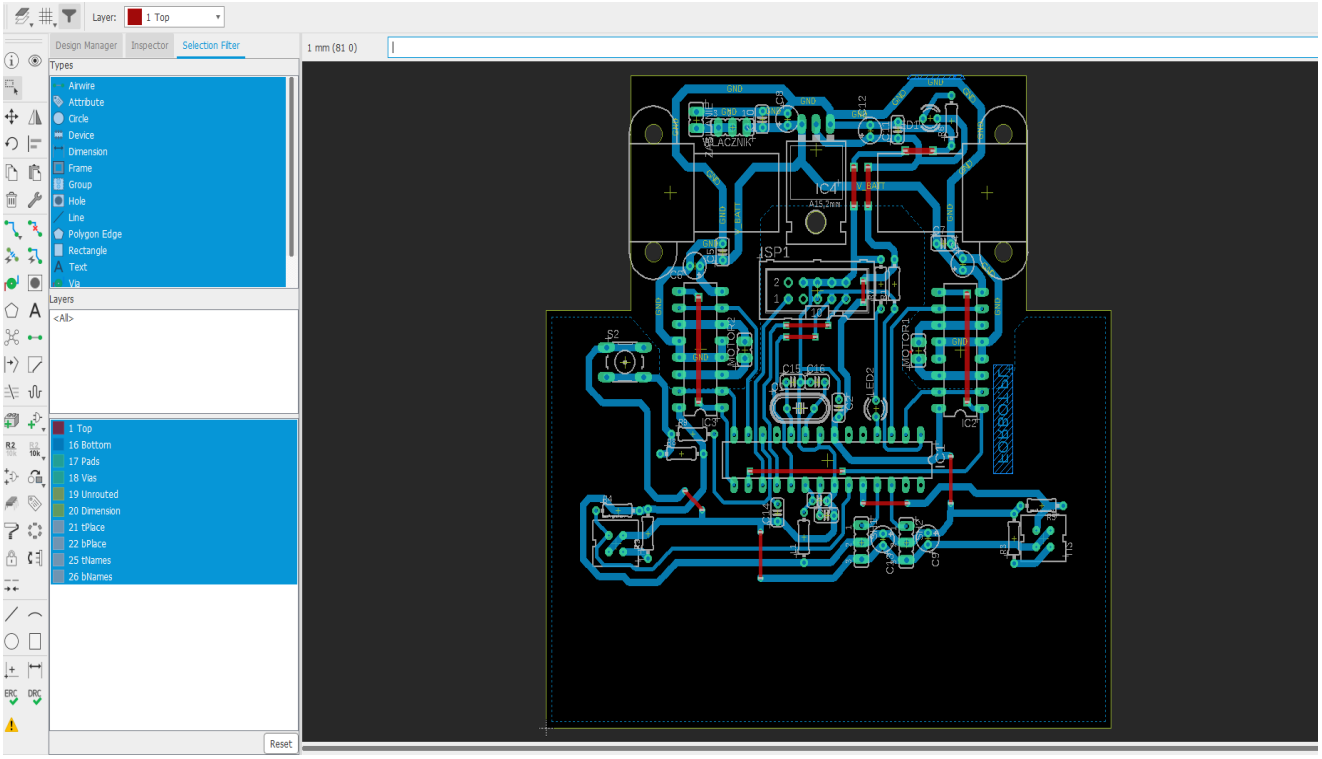
Początkowo próbowałem stworzyć projekt płytki samodzielnie w programie Eagle Autodesk lecz na etapie łączenia całości napotykałem trudność nie do przejścia w postaci łączenia pinów i tworzenia ścieżek(Rys.2.b.1). Poradniki w internecie niewiele dawały ani też komenda ratsnest nie rozplątywała „gniazda". Postanowiłem użyć gotowego schematu (Rys2.b.2 ,Rys.2b.3)różniącego się od moich początkowych koncepcji ale wciąż jednak wystarczająco podobny. Schemat pochodzi ze strony www.forbot.pl.



Rys.2.b.1



Rys.2.b.2



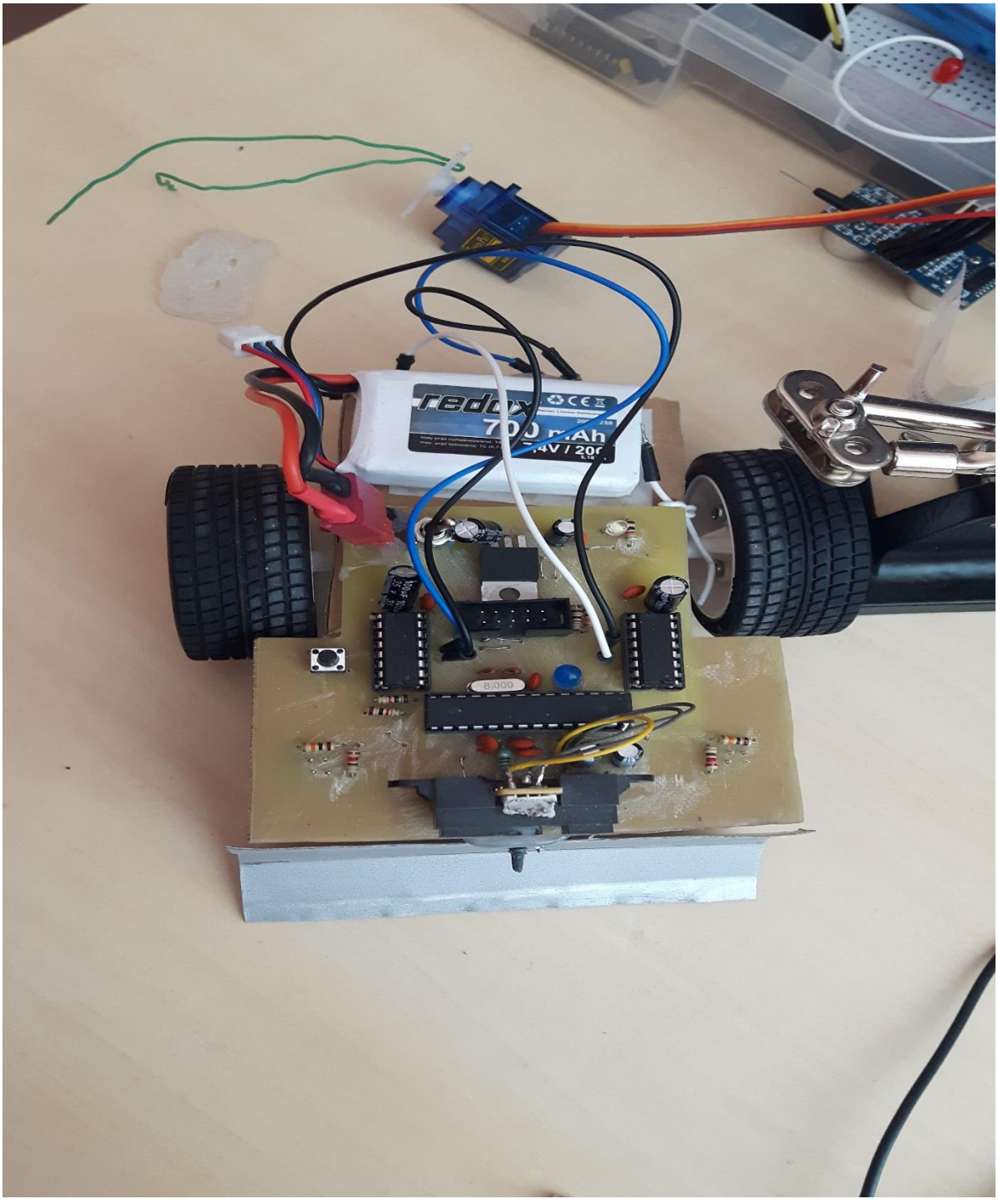
Rys.2.b.3

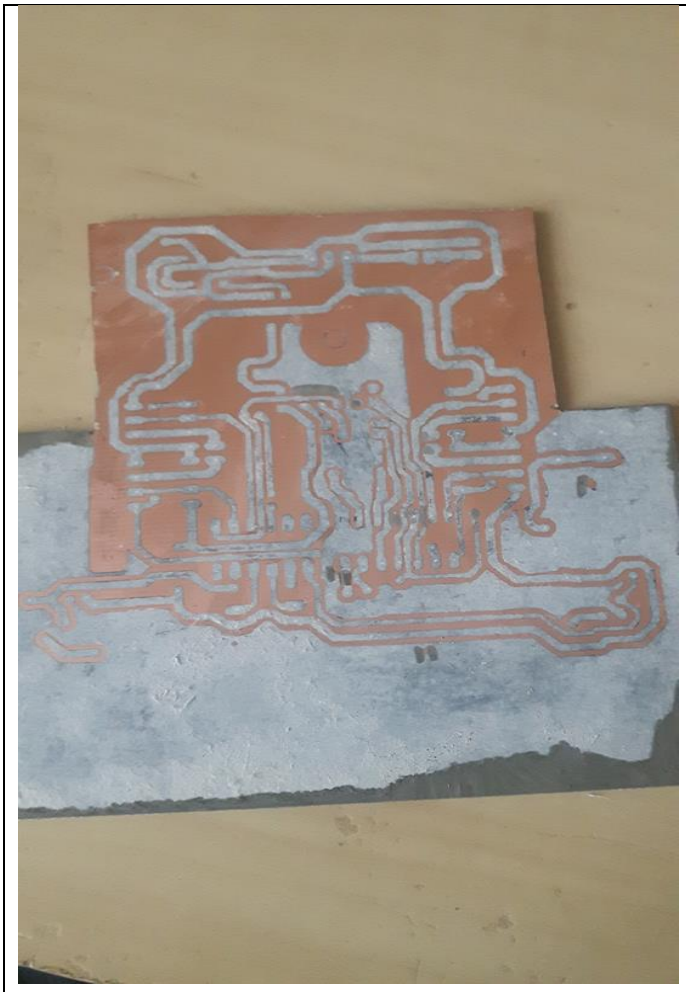
- Opis układu
  Użyte elementy:
- - laminat jednostronny
- -wytrawiacz
- - topnik w płynie, np. RF800
- - mikrokontroler Atmega8 + podstawka
- - 2x podwójne mostki H L293DNE + podstawki
- - 2x transoptory odbiciowe CNY70 do wykrywania linii
- - 1x dalmierz optyczny SHARP GP2Y0A41SK0F
- - stabilizator LDO LM1117 na 5V
- - 3x kondensatory elektrolityczne 100uF
- - 3x kondensator elektrolityczny 47uF
- - 2x rezystor 1k,10k,230
- - 2x diody LED 3mm w dowolnych kolorach
- - 1x dławik 10uH
- Akumulator Li-pol (2S-dwa ogniwa szeregowo) 20C 700mAh 7,4V

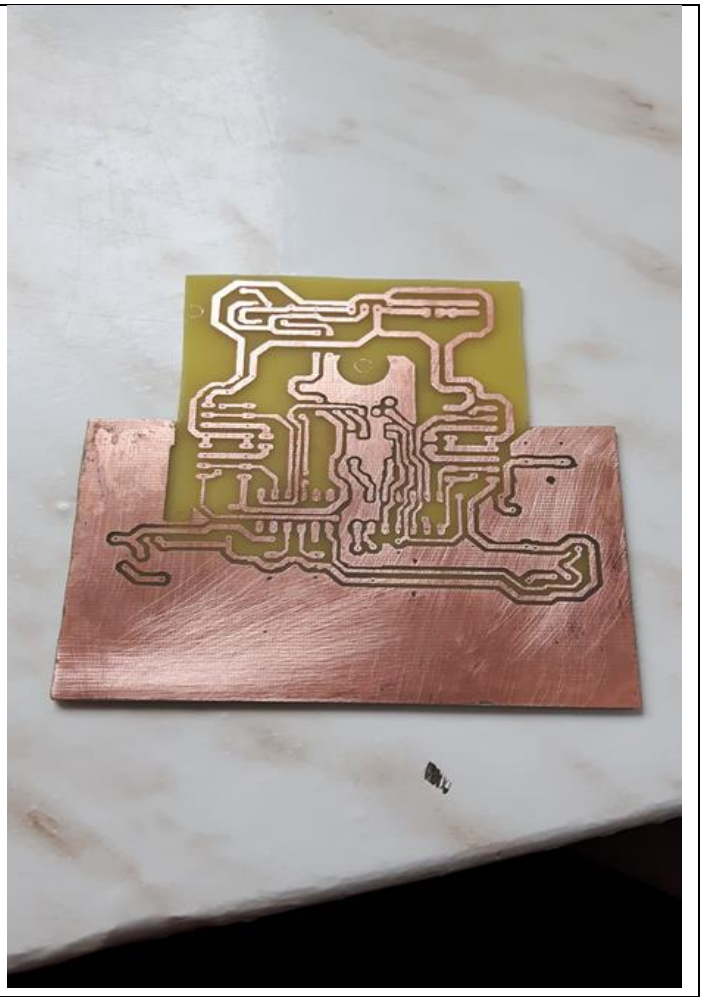
Prąd dostarczany z baterii dociera do switcha z którego filtrowany przez kondesatory dochodzi do stabilizatora napięcia, napięcie 5 V dostarczane jest do Atmegi, czujnika odległości, wejścia na taśme IPS oraz do mostków sterowniczych, napięcie 7,4 V również dostarczane jest do mostków w celu zasilania silników.

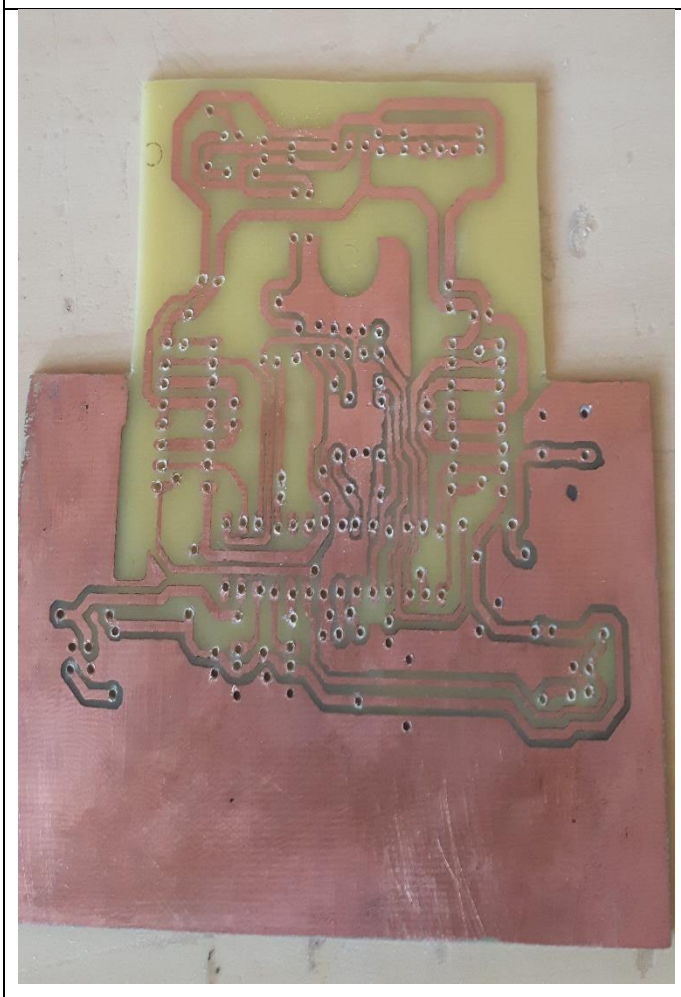
c. Wytrawianie i lutowanie
   Projekt płytki drukujemy(drukarka laserowa) na papierze kredowym o gramaturze około 150, następnie przygrzewamy papier do laminatu żelazkiem do czasu przylgnięcia atramentu do płytki, atrament tworzy warstwę ochronną która nie pozwoli wytrawiaczowi dostać się pod spód tym samym zostawiając wybrane ścieżki. Usuwamy resztki papieru z płytki sprawdzamy pod lupą ścieżki wszelkie przerwy uzupełniamy markerem olejowym(bo tylko

przez taki nie przedostanie się wytrawiacz)(Rys1). Gotujemy wodę i mieszamy z wytrawiaczem w misce(plastykowej lub szklanej) w proporcjach podanych na opakowaniu, wkładamy płytkę i czekamy około 15 minut na wytrawienie warstwy miedzi. Efekt końcowy wytrawiania (Rys2).Następnie wiercimy dziury na piny, ja użyłem do tego drymera z wiertłem 1mm(Rys3). Kolejny etap to lutowanie ścieżek, zajęcie to okazało się w miarę łatwe, co dziwne to nie lutowanie cienkich ścieżek było problemem to duże połacie masy sprawiały największe trudności co widać w efekcie końcowym gdzie cała dolna część to bąble. Lutowanie pinów było na pewno trudniejsze od lutowania ścieżek ale również po paru próbach nabiera się wprawy i wszystko idzie gładko, najciężej było przylutować czujniki linii ponieważ były montowane z drugiej strony laminatu, nie obeszło się bez przypalonego plastyku obudowy.









3. Oprogramowanie

   Instalacja potrzebnego oprogramowania:

   Ta część projektu wydawała mi się jednocześnie najtrudniejsza i najmniej satysfakcjonująca. Instalacja eclipsa na Windowsie okazała się niemożliwa z powodu wymagania środowiska java; sama java IDE mimo wielu prób wciąż napotykała problemy instalacyjne z tego powodu przeniosłem się na system równolegle zainstalowany na moim komputerze czyli Ubuntu 18.1, który uważam za wiele bardziej godny obdarzenia zaufaniem. Instalacja na systemie Ubuntu skróciła się do wpisania paru sudo apt getów w terminalu oraz późniejszego doinstalowania pakietów eclipsa związanych z mikroprocesorami(zwykła nakładka AVR nie zawierała takich pakietów). Należało jeszcze załączyć plik Global.h i środowisko było gotowe do pracy.

4. Programowanie

   Zaczynamy od ustawienia wejść i wyjść na odpowiednich portach:

DDRD=0xBF; // 1011|1111 same wyjścia oprócz PIN6 czyli guzika

DDRC=0x00; //0000|0000 trzy wejścia analogowe ADC0(czujnik lini1) ADC1(czujnik linii 2) ADC2 (czujnik odległości)

DDRB=0xFF; //1111|1111 również same wyjścia

Następnie przygotowujemy mostek H do obsługi silników:

Tabelka mówiąca nam o kierunku obrotów silnika

ENABLE	IN 1	IN 2	SILNIK
PWM	1	0	LEWO
PWM	0	1	PRAWO
PWM	0	0	STOP
PWM	1	1	STOP

W tym wypadku odpowiedzialne będą porty PD0,PD1 Silnik 1 oraz PD7,PB0 Silnik 2
Przykładowa funkcja sterująca:

```
void jazda_przod()

{

sbi(PORTD, PD2); //Dioda

PORTD|=0x01; //ustawia na 1 SIL1 0000|0001

PORTD&=0xFD; //ustawia na 0 SIL1 1111|1101

PORTD&=0x7F; //ustawia na 0 SIL2 0111|1111

PORTB|=0x01; // ustawia na 1 SIL2 0000|0001

OCR1A=100; // Wypełnienie sygnału PWM Silnik 2 Lewy patrząc od tyłu

OCR1B=100; // Silnik 1 Prawy

}
```

Sygnał PWM

To sygnał kwadratowy o odpowiednim stosunku stanu wysokiego do niskiego np. gdy ustawimy wartość PWM na 50% to stan wysoki będzie trwał tyle samo co niski przez co silniki będą dostawały prąd przez połowę czasu (będą pracować w 50% mocy).

Aby korzystać z tego sygnału potrzebujemy dostępnego w ATmedze timera 16-bit ustawionego w tryb fast PWM 8-bit. Szukamy w nocie katalogowej Atmegi 8 rozdziału dotyczącego np. timera1 i znajdujemy tabelkę poświęconą modulacji fali(*Waveform Generation Mode Bit Descriptio)*

Table 16-5. Waveform Generation Mode Bit Description

Mode	WGM13	WGM12 (CTC1)	WGM11 (PWM11)	WGM10 (PWM10)	Timer/Counter Mode of Operation[1]	TOP	Update of OCR1x	TOV1 Flag Set on
0	0	0	0	0	Normal	0xFFFF	Immediate	MAX
1	0	0	0	1	PWM, Phase Correct, 8-bit	0x00FF	TOP	BOTTOM
2	0	0	1	0	PWM, Phase Correct, 9-bit	0x01FF	TOP	BOTTOM
3	0	0	1	1	PWM, Phase Correct, 10-bit	0x03FF	TOP	BOTTOM
4	0	1	0	0	CTC	OCR1A	Immediate	MAX
5	0	1	0	1	Fast PWM, 8-bit	0x00FF	BOTTOM	TOP
6	0	1	1	0	Fast PWM, 9-bit	0x01FF	BOTTOM	TOP
7	0	1	1	1	Fast PWM, 10-bit	0x03FF	BOTTOM	TOP
8	1	0	0	0	PWM, Phase and Frequency Correct	ICR1	BOTTOM	BOTTOM
9	1	0	0	1	PWM, Phase and Frequency Correct	OCR1A	BOTTOM	BOTTOM
10	1	0	1	0	PWM, Phase Correct	ICR1	TOP	BOTTOM
11	1	0	1	1	PWM, Phase Correct	OCR1A	TOP	BOTTOM
12	1	1	0	0	CTC	ICR1	Immediate	MAX
13	1	1	0	1	(Reserved)	–	–	–
14	1	1	1	0	Fast PWM	ICR1	BOTTOM	TOP
15	1	1	1	1	Fast PWM	OCR1A	BOTTOM	TOP

Note: 1. The CTC1 and PWM11:0 bit definition names are obsolete. Use the WGM12:0 definitions. However, the functionality and location of these bits are compatible with previous versions of the timer.

Z niej jasno wynika że musimy ustawić bity WGM12 oraz WGM10 w stan wzbudzony. Znajduja się one w dwóch różnych rejestrach TCCR1A TCCR1B ustawianie ich wygląda tak:

```
TCCR1A |= (1<<WGM10);

TCCR1B |= (1<<WGM12);
```

Kolejnym krokiem jest podłączenie Timera do odpowiedniego pinu mikrokontrolera oraz określenie sposobu sterowania tym wyjściem. W tym celu szukamy tabelki dotyczącej Fast PWM.

Table 16-3. Compare Output Mode, Fast PWM

COM1A1/ COM1B1	COM1A0/ COM1B0	Description
0	0	Normal port operation, OC1A/OC1B disconnected.
0	1	WGM13:0 = 15: Toggle OC1A on Compare Match, OC1B disconnected (normal port operation). For all other WGM1 settings, normal port operation, OC1A/OC1B disconnected.
1	0	Clear OC1A/OC1B on Compare Match, set OC1A/OC1B at BOTTOM, (non-inverting mode)
1	1	Set OC1A/OC1B on Compare Match, clear OC1A/OC1B at BOTTOM, (inverting mode)

Note: 1. A special case occurs when OCR1A/OCR1B equals TOP and COM1A1/COM1B1 is set. In this case the Compare Match is ignored, but the set or clear is done at BOTTOM. See "Fast PWM Mode" on page 91. for more details.

Interesuje nas opcja trzecia, oznacza ona że im większą wartość wpiszemy do OCR tym dłużej na wyjściu będzie panował stan wysoki aby mieć dwa kanały potrzebujemy wpisać jedynkę do COM1A1 i COM1B1:

TCCR1A |= (1<<COM1A1)|(1<<COM1B1);
Ostatnią czynnością jest wybór prescalera mówi on z jaka częstotliwością będzie pracował nasz PWM
Służą do tego bity CSx

Table 16-6. Clock Select Bit Description

CS12	CS11	CS10	Description
0	0	0	No clock source. (Timer/Counter stopped)
0	0	1	$clk_{I/O}$/1 (No prescaling)
0	1	0	$clk_{I/O}$/8 (From prescaler)
0	1	1	$clk_{I/O}$/64 (From prescaler)
1	0	0	$clk_{I/O}$/256 (From prescaler)
1	0	1	$clk_{I/O}$/1024 (From prescaler)
1	1	0	External clock source on T1 pin. Clock on falling edge.
1	1	1	External clock source on T1 pin. Clock on rising edge.

Wybieramy preferowaną wartość w tym wypadku 64
TCCR1B |= (1<<CS10)|(1<<CS11);

Analogowy czujnik odległości

Pierwszą rzeczą do skonfigurowania są wejścia analogowe a dokładniej ADC czyli analog to digital converter. Korzystamy z noty katalogowej Atmegi 8 Dokładniej ze stron od 199 do 201. Po kolei zezwalamy na konwersje, ustawiamy napięcie referencyjne itd. Funkcja inicjująca odczyt:

```
void ADC_init(int kanal)
{
ADMUX&=0xFC; //wyzerowanie mux0 i mux1
ADMUX|=_BV(REFS0);          // makro BV_ to nic innego jak 1<<
if(kanal==0){}                    //kanał ADC0 Czujnik linii 1
else if(kanal==1)
ADMUX|=_BV(MUX0);          //kanał ADC1 Czujnik linii 2
else if(kanal==2)
        ADMUX|=_BV(MUX1); //kanał ADC2 Czujnik odległości
ADCSRA |=_BV(ADEN);          // zezwolenie na konwersje
ADCSRA |=_BV(ADPS0);        // Prescaler
ADCSRA |=_BV(ADPS1);
ADCSRA |=_BV(ADPS2);
}
```

Funkcja odczytu wartości analogowej:

```
float pomiary(int wybor)
{
float pomiar;
ADC_init(wybor);
ADCSRA |=_BV(ADSC);
while(ADCSRA & _BV(ADSC)){}
 pomiar=(float)(ADCL | (ADCH<<8))/1024*5;
 return pomiar;
}
```

Zmienna pomiar wynika ze wzoru przedstawionego w nocie:

For single ended conversion, the result is:

$$ADC = \frac{V_{IN} \cdot 1024}{V_{REF}}$$

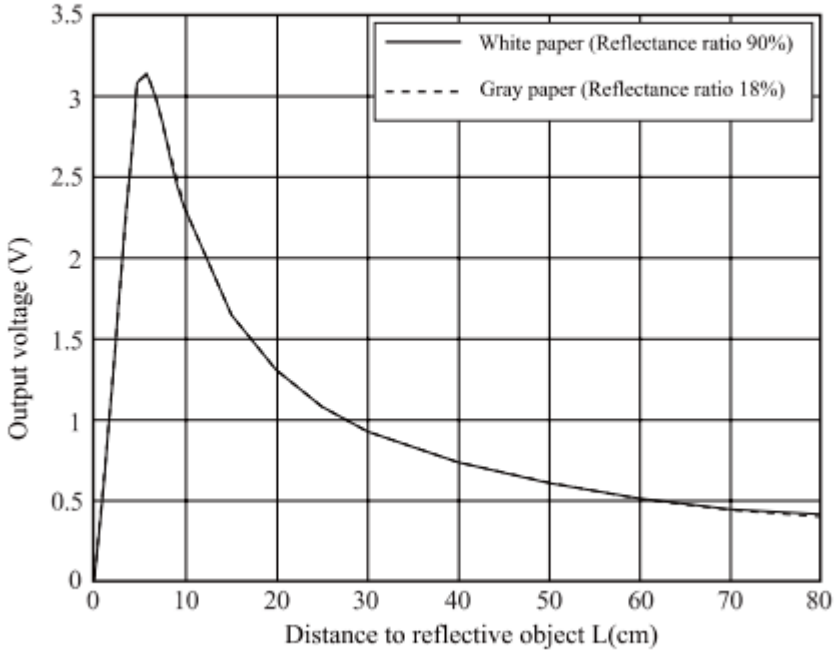
Gdzie pomiar to $V_{in\ napięcie}$ wejściowe
V_{ref} to 5V

ADC to odczyt przechowywany w dwóch rejestrach ADCL i ADCH

Odczyty z czujników linii działają w ten sam sposób jedyna różnica polega na wartościach pomiaru.

Oczekiwane wartości dla czujnika odległości możemy znaleźć w jego datasheeet:

Fig. 2 Example of distance measuring characteristics(output)



Widzimy że poniżej 10 cm odczyt jest niewłaściwy a im dalszy obiekt tym napięcie jest mniejsze. Wykorzystujemy ta zależność w następujący sposób:

else if(pomiary(2)>0.8)

jazda_przod();

dla wszystkich pomiarów z napięciem wyższym od 0.8V czyli około 40cm jedź naprzód.

W przypadku czujników linii nie znalazłem tak bezpośredniego wykresu za to znalazłem informacje że biały odpowiada odczytowi ADC z zakresu <50 a kolor czarny to odczyty 700-1023. Podstawiamy do wzoru wartość trochę mniejszą np.500 i otrzymujemy wartość napięcia 2.4 V. Niestety w czasie tworzenia konstrukcji i modyfikacji zapomniałem o fakcie że czujniki linii musza być około 3mm nad powierzchnią aby działać poprawnie u mnie są one 5 cm nad powierzchnią co uniemożliwia ich prawidłowy odczyt. Sprawdziłem jednak ich działanie i funkcjonują poprawnie.

Wnioski:

- W przyszłych projektach zastanowię się nad użyciem płytki stykowej innym sposobem jest dogłębne nauczenie się obsługi programu Eagle, projektowanie płytki było jednym z trudniejszych zadań którym musiałem sprostać w tym projekcie.
- Kolejną trudnością było oprogramowanie, choć teraz już po pierwszym razie poszłoby mi to na pewno szybciej wciąż uważam że instalacja mogłaby być prostsza.
- Tworzenie konstrukcji robota wybór części i lutowanie sprawiło mi na pewno największą przyjemność i satysfakcję jest to niewątpliwie najlepsze co do tej pory robiłem na uczelni.
- Programowanie zdecydowanie nauczyło mnie najwięcej, nie tylko musiałem poznać zasadę działania części i mikroprocesora ale też jak się między sobą komunikują i przetwarzają dane.

Źródła:

[1] Sygnał PWM

https://forbot.pl/blog/kurs-elektroniki-ii-sterowanie-sygnalem-pwm-id9527

[2] Noty katalogowe

https://www.alldatasheet.com/

[3] Płytka PCB

https://forbot.pl/forum/topic/5636-kurs-przepis-na-robota-minisumo-dla-kazdego-czesc-1/

[4] Atmega ADC

www.elektroda.p/rtvforum/miernikadc/

[5] Operacje bitowe DDRY PORTY itd.

http://hobby.abxyz.bplaced.net/index.php?pid=4&aid=2