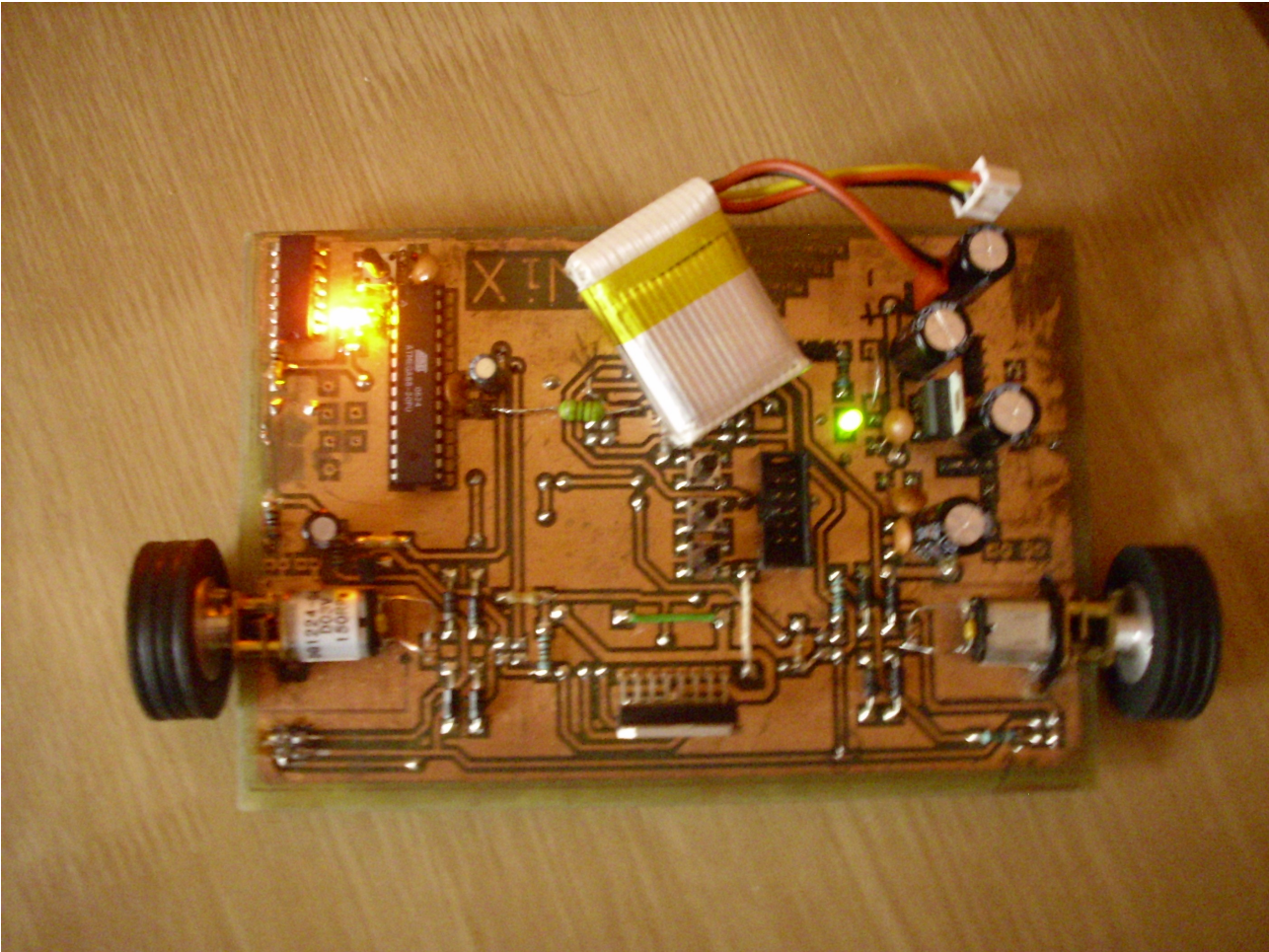


Mateusz Skiba
Jakub Szymaczek

LINEFOLLOWER „WiX”



Spis Treści

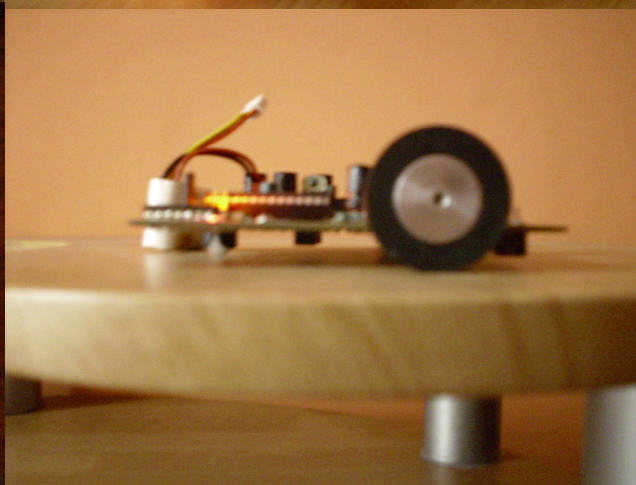
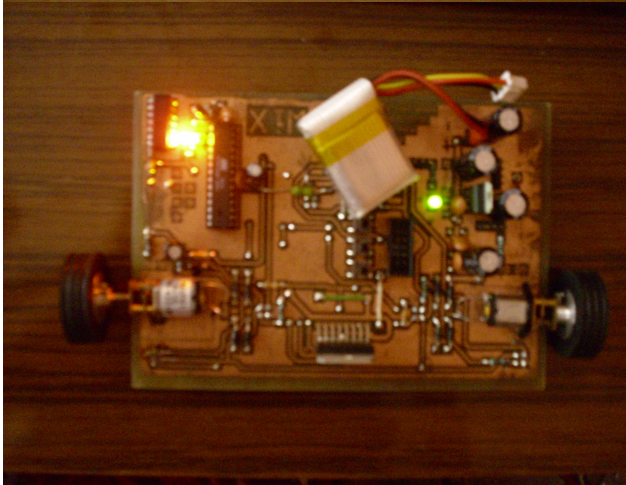
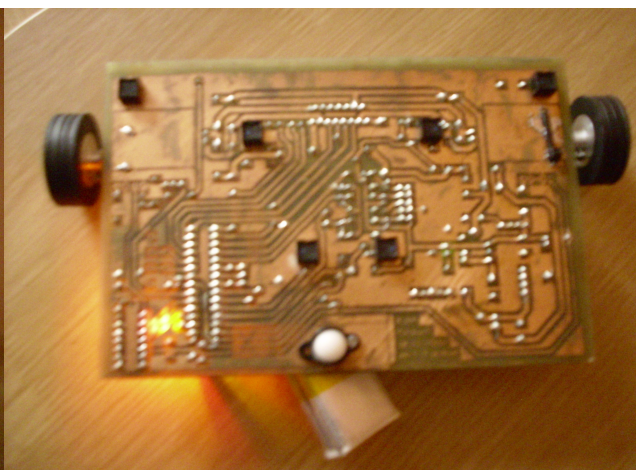
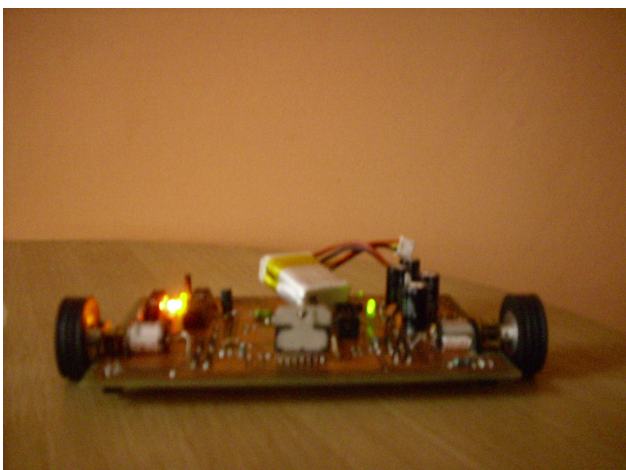
| | |
|--|--|
| 1. Wstęp..... | |
| 2. Konstrukcja nośna..... | |
| 3. Zasilanie..... | |
| 4. Mikrokontroler i towarzyszące elementy elektroniczne..... | |
| 5. Czujniki..... | |
| 6. Sterowanie napędami..... | |
| 7. Zaimplementowany program..... | |
| 8. Literatura..... | |
| 9. Zdjęcia..... | |

1. Wstęp

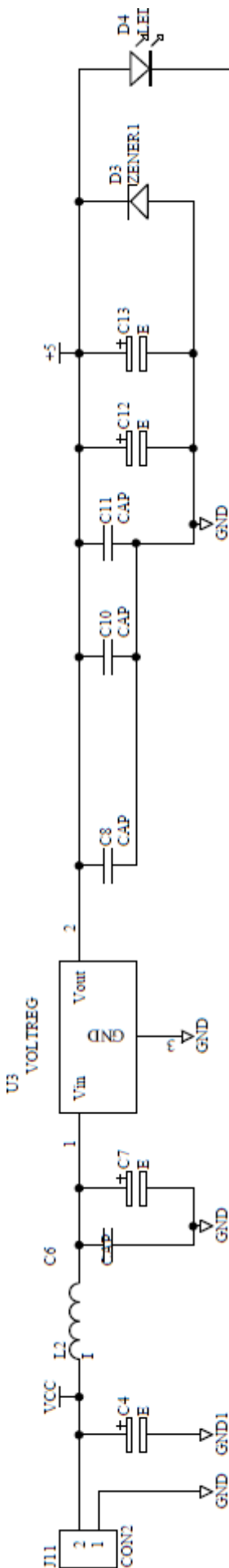
Przedstawiony raport prezentuje informacje dotyczące tworzenia robota typu linefollower o nazwie „WiX”. Robot powstał w związku z warsztatami rekrutacyjnymi do koła naukowego KoNaR.

2. Konstrukcja nośna

Konstrukcja nośna robota opiera się na płytce drukowanej wykonanej z laminatu dwustronnego. Do płytki po bokach przymocowane są silniki wraz z przekładniami oraz wyłożyszowana kulka spełniająca rolę ślizgu.



3. Zasilanie



Robot zasilany jest akumulatorem litowo-polimerowym, 2-ogniowym o napięciu 8,4V oraz pojemności 400 mAh (złącze J11). Akumulator świetnie radzi sobie ze znacznym obciążeniem pochodzącym z silników. Do układu zasilającego zalicza się również stabilizator napięcia 7805. Stabilizuje on napięcie akumulatora na 5V, które służy do zasilania mikroprocesora robota. Jest to stabilizator typu Low Drop Out – nawet przy krytycznie niskim poziomie akumulatora (6V) napięcie wyjściowe wynosić będzie 5V. Kondensatory ceramiczne (o pojemności 100 nF każdy) oraz elektrolitowe (o pojemności 2200uF każdy) służą do odfiltrowania zakłóceń w układzie. W układzie w celach filtracji przewidziana została również cewka L2. Przy budowie robota zastąpiono ją jednak zwarcieniem. Poprowadzono dwie ścieżki masy (GND oraz GND1) osobne dla mostka H z silnikami oraz reszty układu celem zmniejszenia zakłóceń pochodzących od silników.

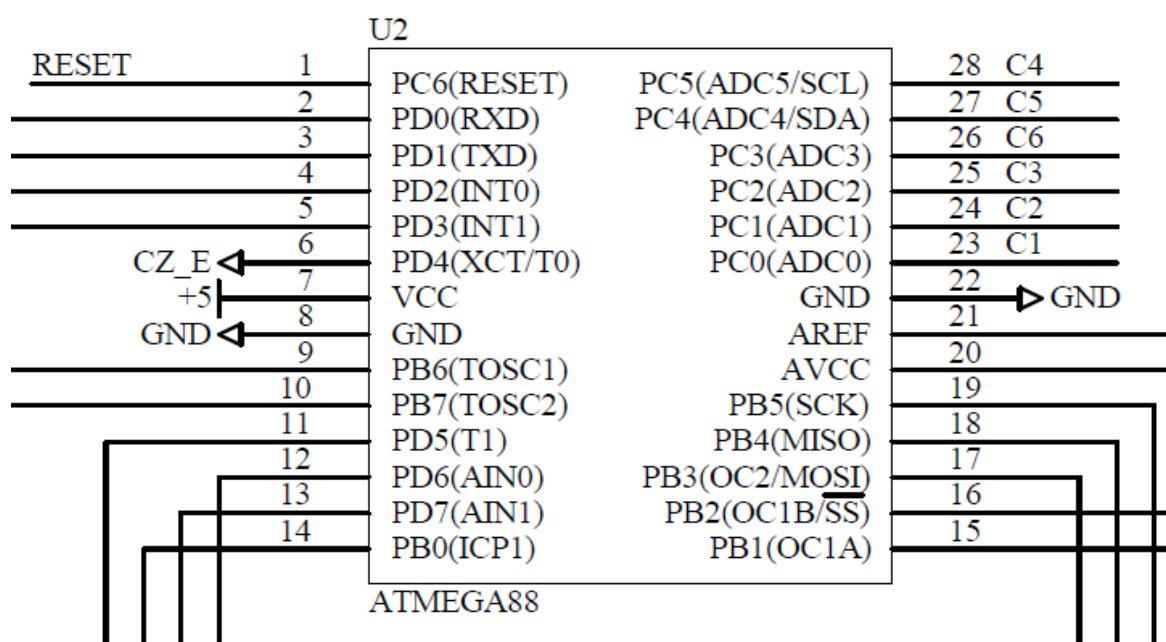
4. Czujniki

Wykorzystano czujniki odbiciowe CNY70 w ilości 6 sztuk. Zostały one rozmieszczone symetrycznie wzdłuż robota, po 3 z każdej strony na planie 1/4 okręgu w ten sposób że rozchodzą się od środka na zewnątrz. Dwa czujniki skrajne S1 oraz S2 (po jednym z każdej strony) znalazły się przed osią kół. Reszta znalazła się za osią. Odległość między środkami czujników S5 oraz S6 wynosi 22 mm. Czujnikami steruje tranzystor typu MOSFET odbierający sygnał z mikroprocesora. Mikroprocesor włącza i wyłącza tranzystor co powoduje naprzemienne pomiary natężenia odbitego promieniowania z włączoną i wyłączoną diodą (pomiar różnicowy). Rezystancje elementów R2 oraz R3 zostały dobrane eksperymentalnie, tak by brak natężenia odbitego światła dawał na wyjściu czujnika napięcie około 5V, a natężenie odbite od materiały białego napięcie około 150 - 200 mV.

5. Mikrokontroler i towarzyszące elementy elektroniczne

a) Mikrokontroler

Do sterowania robotem użyto mikroprocesora z rodziny AVR firmy Atmel – ATmega88. Układ zasilany może być napięciem 4,5 – 5,5 V. Taktowany jest wewnętrznym rezonatorem 8MHz. Posiada przetwornik analogowo-cyfrowy z 6 kanałami, obsługuje 2 przerwania zewnętrzne, posiada kanały PWM. Posiada 8kB pamięci flash, co jest wystarczające nawet na bardzo obszerny program na robota typu Linefollower. Od mikrokontrolera wyprowadzone są nóżki razem z filtry na zewnętrzny rezonator. Oprócz zasilania logiki podpięte i odfiltrowane za pomocą kondensatora (C3, 100 nF) oraz cewki (L1, 10uH) napięcie zasilające przetwornik analogowo-cyfrowy. Do wyprowadzenia złącza do programowania (JP1) podpięte są micoswitche. Dzięki temu wyprowadzenia mikroprocesora, które domyślnie są przeznaczone do programowania mogą być wykorzystywane w czasie pracy programu. Przyciski filtrowane są kondensatorami(C26, C27, C28, każdy po 100nF) celem pozbycia się efektu drgania styków przycisków.



b) Układ scalony 4093

Przewidziane zostały enkodery na koło w postaci diody, fotokomórki oraz kółka z naciętymi szczelinami pochodzące z myszki kulkowej. Układ scalony składa się z 4 bramek NAND z układem Schmitta, których celem jest zniwelowanie stanów wolnozmiennych generowanych przez enkodery.

c) dodatkowe wyprowadzenia

Przygotowane zostały dodatkowe złącza J12(5V), J16(zasilanie z akumulatora) oraz J17(masa), które umożliwiają dalszą rozbudowę robota.

6. Sterowanie napędami

Robot napędzany jest dwoma miniaturowymi silnikami prądu stałego DG1224-043 ze zintegrowaną przekładnią zębatą o przełożeniu 43:1(150 obrotów/min, napięcie zasilania 3V). Do ich sterowania użyty został mostek H – L298. 4 wyprowadzenia z mikrokontrolera (PD5,PD6,PD7,PB0) pozwalają sterować kierunkiem obrotu silnika. Wyprowadzenia PB1, PB2 odpowiadają za modulację PWM, co pozwala regulować moc obrotu silników. Układ L298 zasilany jest z napięcia stabilizowanego 5V, silniki natomiast, bezpośrednio z akumulatora. Silniki zabezpieczone przed prądami wstecznymi są diodami szybkimi 1N4007.

7. Zaimplementowany program:

Program napisany został w języku C. Część programu wzorowana lub skopiowana została z udostępnionych przez KoNaR materiałów. Zajmuje on w przybliżeniu 1kB (1/8 pamięci flash mikroprocesora) . Można go podzielić można na 4 części:
-sekcja nagłówkowa (zdefiniowane makra ułatwiające pisanie kolejnych części programu):

```
#define F_CPU 8000000L
#define SET_IN2 PORTD  |= _BV(7)  /* PB0 -> INPUT 1 L298 */
#define CLR_IN2 PORTD  &= ~_BV(7)
#define SET_IN2_OUT DDRD |= _BV(7)

#define SET_IN1 PORTB  |= _BV(0)  /* PD7 -> INPUT 2 L298 */
#define CLR_IN1 PORTB  &= ~_BV(0)
#define SET_IN1_OUT DDRB |= _BV(0)

#define SET_IN4 PORTD  |= _BV(5)  /* PD6 -> INPUT 3 L298 */
#define CLR_IN4 PORTD  &= ~_BV(5)
#define SET_IN4_OUT DDRD |= _BV(5)

#define SET_IN3 PORTD  |= _BV(6)  /* PD5 -> INPUT 4 L298 */
#define CLR_IN3 PORTD  &= ~_BV(6)
#define SET_IN3_OUT DDRD |= _BV(6)

/* UWAGA: Dwa ponizsze porty nie moga zostac zmienione na inne! */
#define SET_ENA_OUT DDRB |= _BV(1) /* PB1 (OC1A) */
#define SET_ENB_OUT DDRB |= _BV(2) /* PB2 (OC1B) */
#define ADC_ENABLE          (1<<ADEN)
#define ADC_CONVERT         (1<<ADSC)
#define ADC_VREF_TYPE       ((0<<REFS1) | (1<<REFS0))
#define ADC_PRESCALE_128   ((1<<ADPS2) | (1<<ADPS1) | (1<<ADPS0))

#define T 25
#define PRZOD 600
#define LEKKO 300
#define OSTRO 0

#define L1 500*0.6
#define L2 300*0.6
#define L3 0
#define WW 600

#define P1 530*0.6
#define P2 315*0.6
#define P3 0

#define WS 1023

#define PROG 400
#include <avr/io.h>
```

```
#include <util/delay.h>
```

```
/* pętla właściwa programu odpowiedzialna za sterowanie robotem na podstawie wiadomości z czujników */
```

```
while(1){  
  
    stan=czytaj_adc();  
    if(!stan)  
        jazdawprzod(L1,P1);  
    else if(stan&0b100000){  
        obrotwprawo(WS,WW);  
        _delay_ms(T);}  
    else if(stan&0b000001)  
    {  
        obrotwlewo(WW,WS);  
        _delay_ms(T);}  
    else if(stan&0b010000)  
        jazdawprzod(L3,P1);  
    else if(stan&0b000010)  
        jazdawprzod(L1,P3);  
    else if(stan&0b000100)  
        jazdawprzod(L2,P1+100);  
    else if(stan&0b001000)  
        jazdawprzod(L1+100,P2);  
    else if(stan&0b001100)  
        jazdawprzod(L1,P1);  
  
    else if(stan&0b000011)  
    {  
        obrotwlewo(WW,WS);  
        _delay_ms(T);}    else if(stan&0b110000)  
    {  
        obrotwprawo(WS,WW);  
        _delay_ms(T);}  
  
    else if(stan&0b111000)  
    {  
        obrotwprawo(WS,WW);  
        _delay_ms(1);}  
    else if(stan&0b000111)  
    {  
        obrotwlewo(WW,WS);  
        _delay_ms(1);}  
    else if(stan&0b110011)  
        jazdawprzod(L1,P1);  
    else if(stan&0b011110)  
        jazdawprzod(L1,P1);  
  
    }  
}
```

8. Wnioski

Podczas budowy robota oraz testowania go na torze wykryto kilka wad, których należałoby uniknąć podczas konstruowania tego typu robotów:

a) Założenie, że lepiej sterować robotem z osią obrotu z przodu okazało się błędne. W przypadku ostrzejszych zakrętów tylko przednie czujniki wykrywają czarną linię. Gdy robot jedzie z dużym wypełnieniem sygnału PWM po wykryciu czarnej linii silniki nie są w stanie wystarczająco szybko zareagować i siła bezwładności robota wyrzuca go poza linię i robot nie jest w stanie już wrócić.

b) Wymiary robota oraz rozmieszczenie czujników na planie okręgu celem „ścinania” zakrętów nie spełniają swoich założeń. Efekt jest wprost odwrotny – robot skręcając zatacza szeroki łuk. Najlepszym wyjściem wydaje się tu odwrotne, standardowe podejście: długi robot z wysuniętymi czujnikami.

c) Silniki zasilane bezpośrednio z w pełni naładowanego akumulatora powodują szybszą pracę kół jednak wraz ze spadkiem napięcia zasilającego robot zachowuje się inaczej, co bardzo komplikuje dobór odpowiednich nastaw algorytmu. Rozsądnym wyjściem wydaje się zastosowanie przetwornicy impulsowej, która zasili silnik stałym napięciem, bez znacznych strat energii.

d) Kondensatory podłączone równolegle do microswitchy celem zniwelowania efektu drgania styków musiano wylutować. Okazało się, że wygładzają sygnały programatora, co uniemożliwiało komunikację z mikroprocesorem.