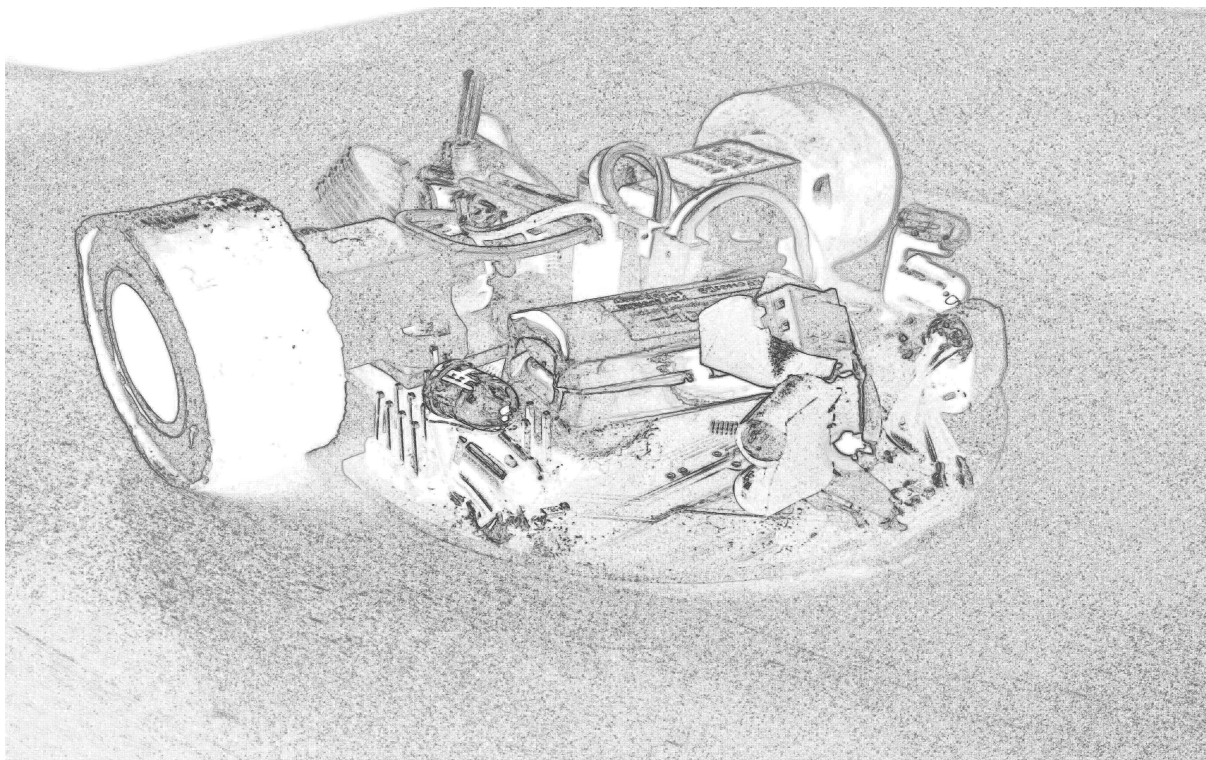

Raport

Robot mobilny klasy micromouse

Adrian Gałęziowski
Paweł Urbaniak



Wrocław, 12 marca 2013

Spis treści

1. Założenia projektu	2
2. Model w programie Autodeksk Inventor.	2
3. Konstrukcja mechaniczna.	4
4. Konstrukcja elektroniczna.	5
5. Algorytm.	9
6. Problemy.	9
7. Planowane modyfikacje.	10

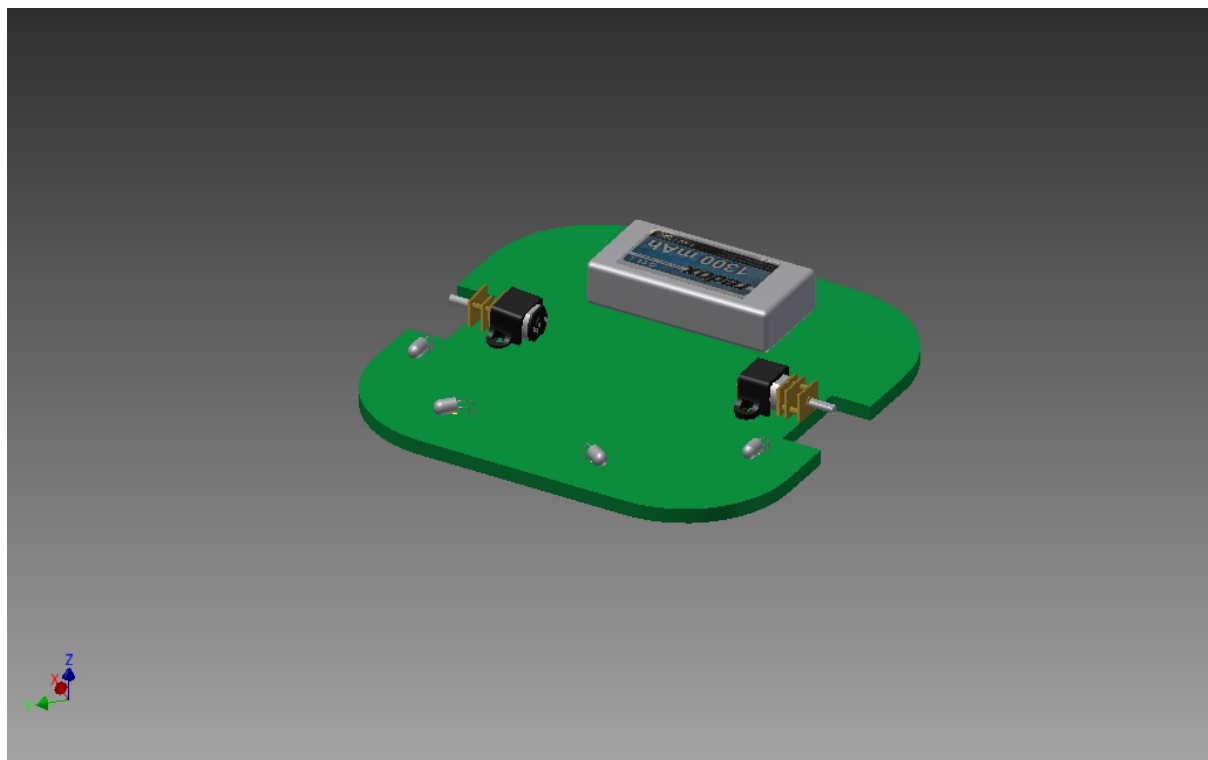
1. Założenia projektu

Celem projektu było zbudowanie robota mobilnego klasy 2.0, typu micromouse, który mógłby z powodzeniem rywalizować z podobnymi konstrukcjami na zawodach robotów. Robota udało mi się ukończyć przed zawodami Robotic Arena 2012, co pozwoliło na sprawdzenie konstrukcji podczas zawodów.

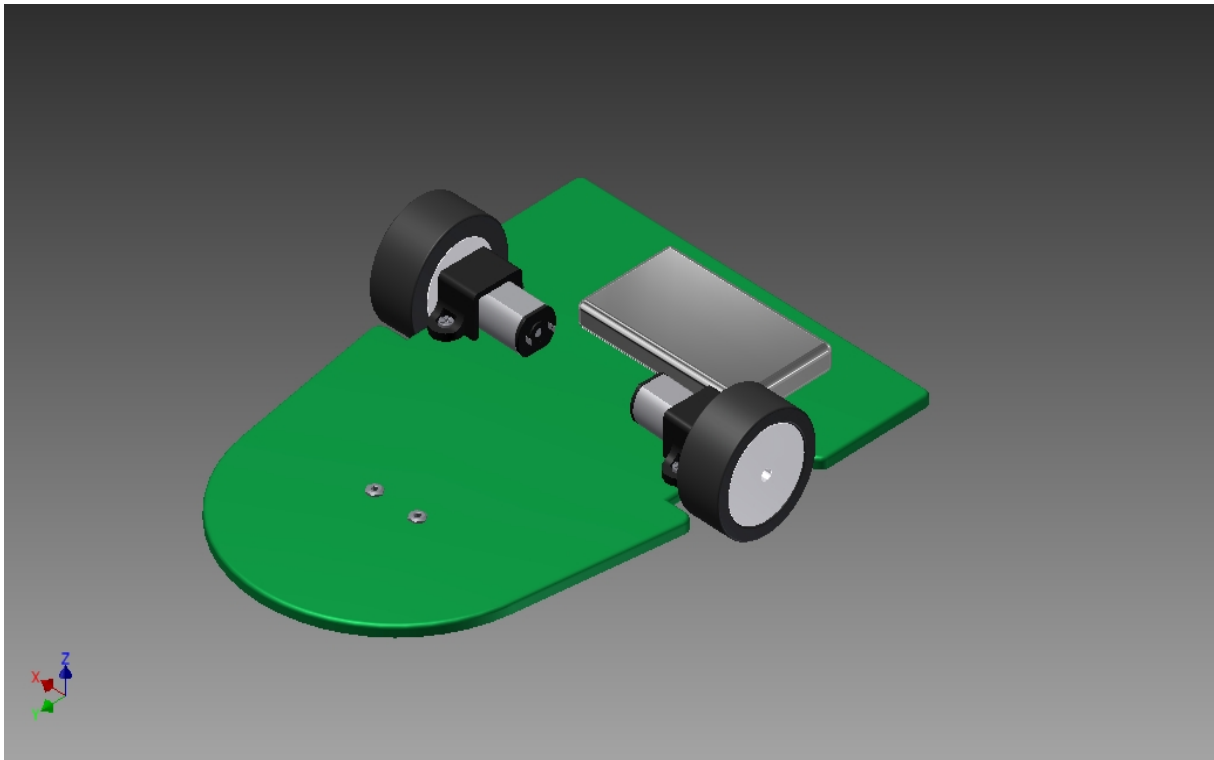
Niniejszy raport ma na celu przedstawienie kolejnych etapów budowy robota począwszy od modeli 3D w programie Autodesk Inventor, aż po finalną konstrukcję.

2. Model w programie Autodesk Inventor.

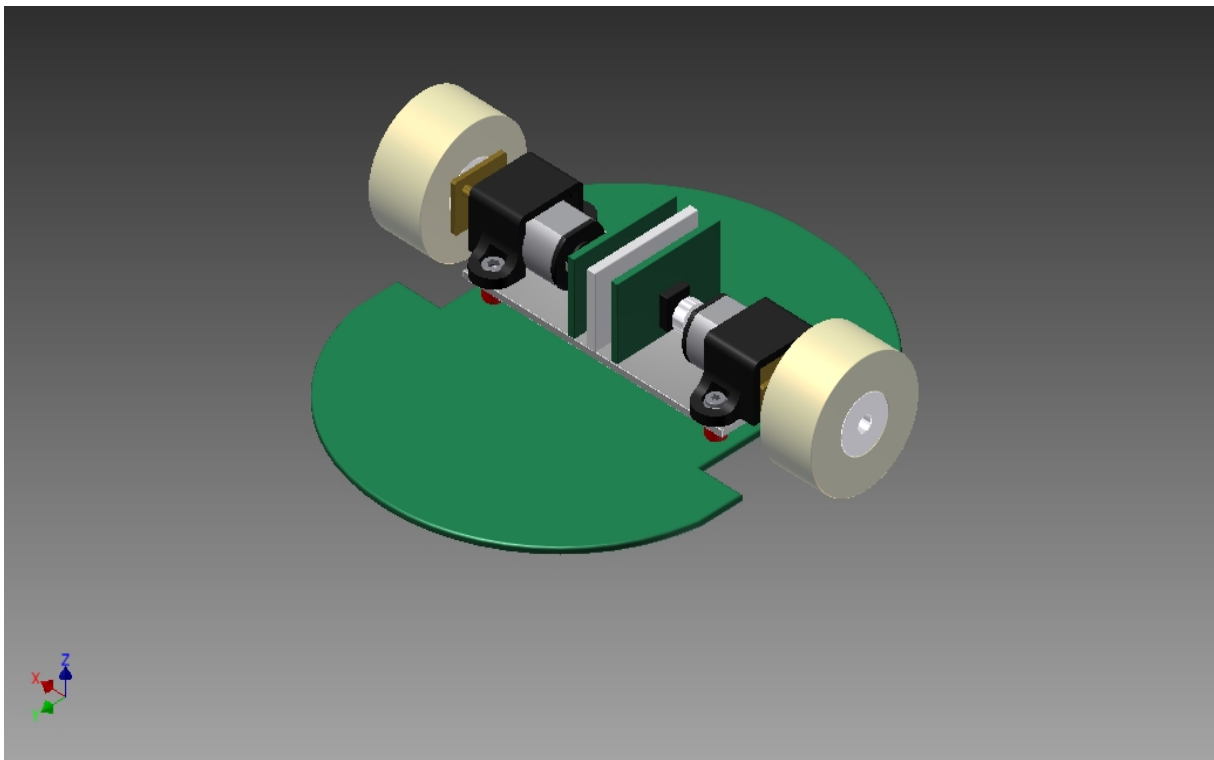
Pracę nad projektem rozpocząłem od wykonania kilku modeli 3D w programie Autodesk Inventor, w celu wybrania najlepszej koncepcji co do kształtu robota, a także żeby mieć pewność, że wszystkie zaplanowane elementy zmieszczą się w konstrukcji robota. Poniżej prezentuje kolejne stadia projektu.



Rysunek 1. Wczesne stadium - pierwsza koncepcja na wygląd robota.



Rysunek 2. Drugi projekt.

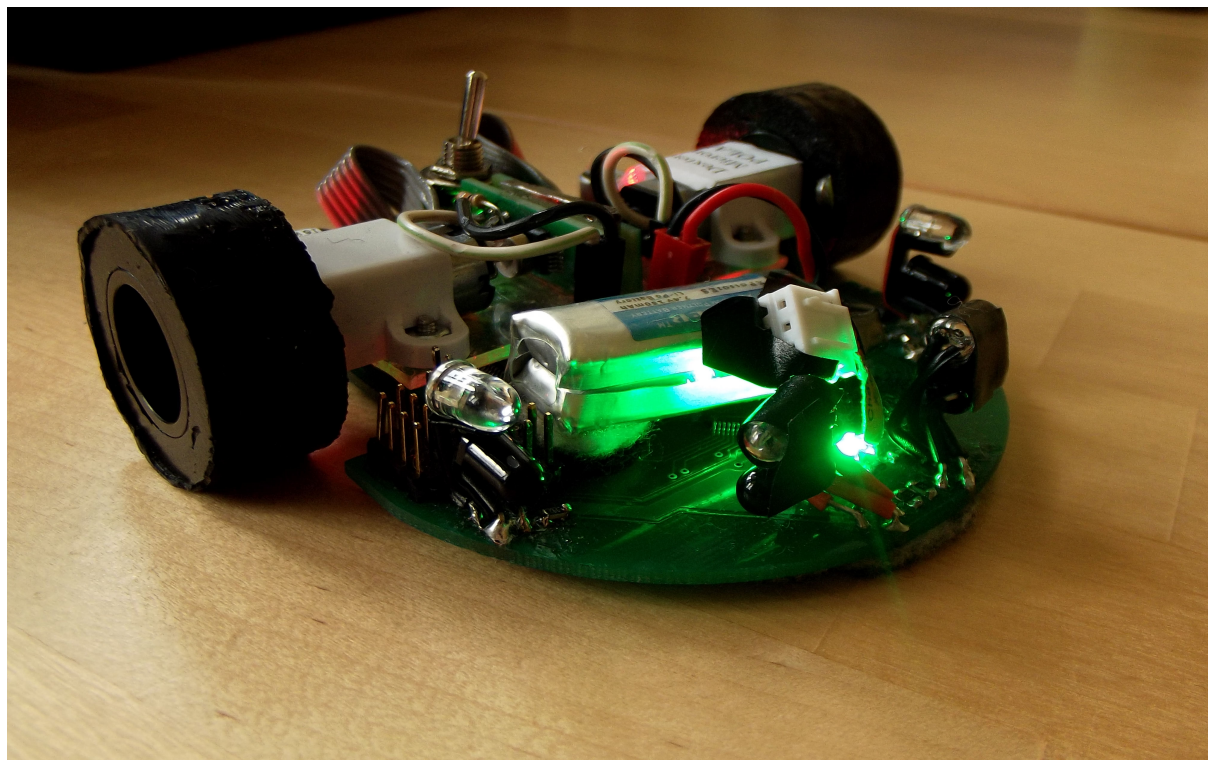


Rysunek 3. Ostateczna wersja.

W pierwszym założeniu robot był krótki, ale zbyt szeroki, co pozostawiałoby mu bardzo mały margines błędu podczas pokonywania zakrętów. W drugiej konstrukcji robot został znacząco wydłużony, jednakże całościowo był zbyt duży. Ostatecznie robot został skrócony i maksymalnie zwężony do minimalnej szerokości wyznaczonej przez silniki i enkodery, które zostały zaplanowane pomiędzy silnikami. Ostateczne wymiary robota są następujące: 10cm x 10 cm.

3. Konstrukcja mechaniczna.

Bazę dla całej konstrukcji robota stanowi jego płytkę z elektroniką. W robotach micro-mouse jednym z ważniejszych parametrów jest niska masa, stąd decyzja o zrezygnowaniu z płytki bazowej. Silniki zostały podniesione o ok. 3mm i zamocowane na dodatkowej płytce z pleksiglasu, w celu obniżenia prześwitu robota. Dzięki temu uzyskałem również miejsce na moduł bluetooth, który został umieszczony pod silnikami. Płytki z enkoderami zostały przyklejone do płytki z pleksiglasu. Pomiedzy nimi została umieszczona płytki antymagnetyczna w celu wykluczenia występujących zakłóceń pomiędzy enkoderami. Do napędu robota posłużyły silniki Pololu z przekładnią 10:1 (1000 obr/min), zapewniając mu bardziej niż wystarczające osiągi. Powszechnie dostępne koła modelarskie zapewniają stosunkowo słabą przyczepność, dlatego koła zostały wykonane samodzielnie, z użyciem silikonu sanitarnego, który spisywał się bardzo dobrze przy wysokich prędkościach zapewnianych przez silniki ($V_{max} = 3m/s$)



Rysunek 4. Dexter.

4. Konstrukcja elektroniczna.

1. Mikroprocesor.

Micromouse potrzebuje procesora o dużej mocy obliczeniowej, który byłby w stanie na bieżąco aktualizować mapę labiryntu, obliczać sterowanie dla silników i obsługiwać wszystkie peryferia, w szczególności enkodery. Wybór padł na procesor z rdzeniem Cortex-M3 - **STM32F103RBT6**. Kilka parametrów technicznych, które zdecydowały o wyborze:

- Maksymalna częstotliwość pracy: 72 MHz;
- 128 kB Flash;
- 20 kB RAM;
- Minimalny czas przetwarzania ADC: $1\mu\text{s}$;
- Sprzętowa obsługa enkoderów.

2. Enkodery.

Do pomiaru przemieszczenia robota użyłem enkoderów magnetycznych **AS5040**, które mierzą przemieszczenie z rozdzielczością 10 bitów zapewniając 1024 tiki na jeden obrót koła. Enkodery współpracują z magnesami AS5000-MD6H-2 zalecanymi przez producenta.

3. Żyroskop.

W celu dokładniejszego pomiaru kąta obrotu robot został wyposażony w żyroskop analogowy **LY3200ALH**, który potrafi zmierzyć maksymalną prędkość obrotu na poziomie 2000dps.

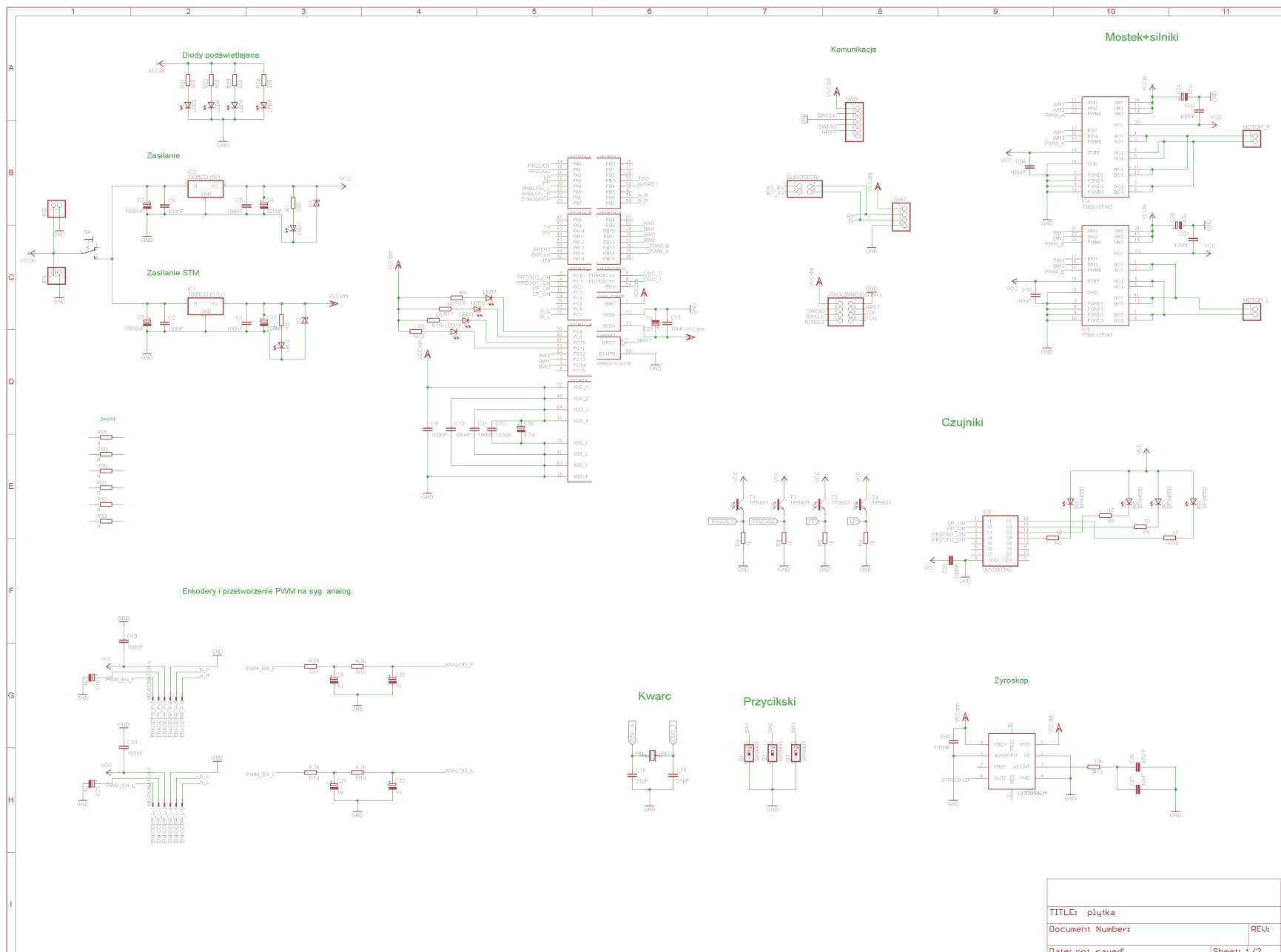
4. Czujniki.

Czujniki do wykrywania ścian zostały również wykonane samodzielnie na podstawie podczerwonej diody nadawczej **SFH4550** emitującej falę o długości 850nm oraz fototranzystora **SFH313**.

5. Komunikacja.

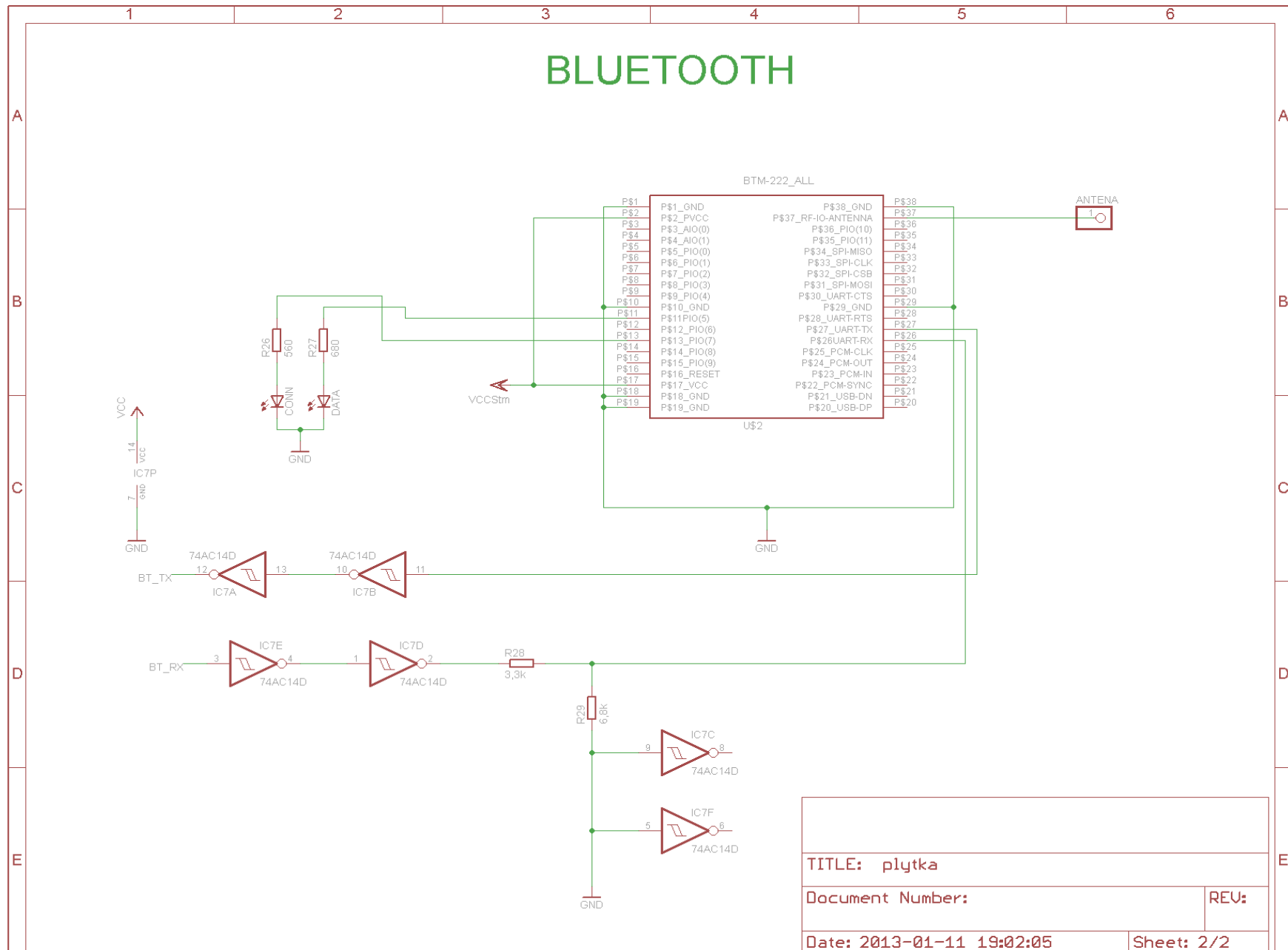
W robocie istnieje możliwość komunikacji trzema różnymi sposobami, za pomocą interfejsu SWD, JTAG oraz bezprzewodowo za pomocą modułu bluetooth BTM-222.

Poniżej zamieszczam schemat elektroniczny robota:



TITLE: płytka	
Document Numbers:	REV:
Date: not saved!	Sheet: 1/2

Rysunek 5. Schemat elektroniczny cz.1



Rysunek 6. Schemat elektroniczny cz.2

5. Algorytm.

Podstawowym elementem robota micromouse poza przemyślaną i zwartą konstrukcją mechaniczną jest optymalny pod względem szybkości i jakości działania algorytm rozwiązywania labiryntu. W robocie został zaimplementowany **algorytm zalewania wodą (Bellmana - Forda)**.

Główną idą tego algorytmu jest przypisanie każdemu z pól w labiryncie odpowiedniej wartości liczbowej. Wiąże się to z dużą złożonością pamięciową jak na możliwości mikroprocesorów, ponieważ potrzebujemy tablicy 256 bajtowej do przechowywania wag każdego z pól oraz kolejnej tablicy 256 bajtowej do przechowywania informacji o położeniu ścian w dane komórce oraz innych informacji, które uznamy za niezbędne. Jak wiadomo każdy bajt ma 8 bitów, więc taka tablica powinna nam w zupełności wystarczyć do przechowywania wszystkich niezbędnych informacji.

1. Zasada wypełniania tablicy

Algorytm przypisuje wartość 0 środkowi labiryntu. Następnie każdemu sąsiadowi danej komórki, który nie jest oddzielony ścianą przypisujemy wartość o 1 większą. W ten sposób przechodzimy po wszystkich polach tablicy.

Procedura ta musi być powtórzona w momencie gdy robot wykryje kolejną ścianę w labiryncie podczas przejazdu rozpoznawczego.

W ten sposób poruszając się od kwadratu o największej wartości do najmniejszej otrzymujemy najkrótszą drogę do celu.

2. Droga do celu

W momencie, gdy robot zbada cały labirynt otrzymujemy najkrótszą, ale nie niekonięcznie najszybszą drogę do celu. Robot w każdym ruchu powinien wykonać następujące czynności:

- 1) Zaktualizuj mapę ścian.
- 2) Zaktualizuj tablicę wag poszczególnych pól.
- 3) Zdecyduj, który z sąsiadów danego pola ma najmniejszą wartość odległości.
- 4) Przejdź **do** pola, o najmniejszej wartości odległości.

6. Problemy.

Podczas tworzenia konstrukcji jak i podczas zawodów spotkałem się z następującymi problemami. Poniżej prezentuję krótki opis problemu oraz moje własne pomysły na jego rozwiązanie.

1. Czujniki.

Najwięcej problemów podczas zawodów sprawiły czujniki, które wykazują brak odporności na zmienne oświetlenie. Z tego powodu robot nie był również w stanie dojechać do środka labiryntu podczas zawodów.

Planuję poprawić obecnie zaimplementowany pomiar różnicowy, który w obecnej postaci nie spełnia swojego zadania. Dodatkowo chcę wprowadzić dynamiczne ustawianie wszystkich progów, a nie tylko niektórych jak to ma miejsce obecnie, przed starterem robota.

2. Resetowanie mikroprocesora.

Znacznie większym problemem jest resetowanie się mikroprocesora.

Podejrzewam, że obecnie zaimplementowany algorytm obliczania ścieżki powinien zostać zoptymalizowany, przede wszystkim operacje na tablicach powinny zostać zastąpione operacjami na stosie. Ponadto resetowanie następuje w momencie zerowania zmiennych odpowiadających za pomiar przemieszczania, należy więc zmienić moment resetowania tych zmiennych.

7. Planowane modyfikacje.

1. Uruchomienie komunikacji bluetooth z laptopem, ponieważ z powodu problemów sprzętowym z laptopem możliwa jest jedynie komunikacja z telefonem komórkowym.
2. Optymalizacja algorytmu zalewania wodą. Nie jest konieczne przeglądanie całej tablicy po wykryciu nowej ścianki. Wystarczy jedynie zaktualizować tę część labiryntu, której ta ściana dotyczy. Przechowywanie labiryntu na stosie zamiast w tablicy.