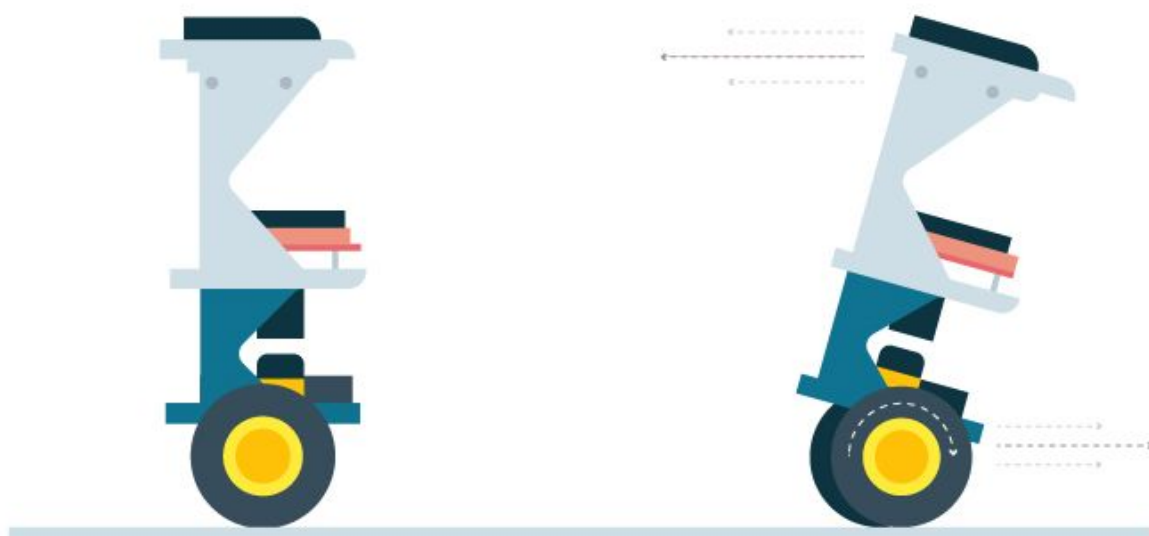


Balansujący pojazd dwukołowy

Cel:

Stworzenie samobalansującego pojazdu dwukołowego. Zamierzam go projektować na wzór Segway'a; moje urządzenie będzie w podobny sposób balansować. Do zaprogramowania Arduino użyję języka programowania C.

Ogólny sposób działania:



Stan stabilny

Wytrącony z równowagi

Etapy:

1. Zdobycie potrzebnych podzespołów m.in.
 - Arduino
 - silniki (serwomechanizm)
 - czujniki (żyroskop)
 - zasilanie oraz inne akcesoria potrzebne przy budowie robota
2. Podłączenie żyroskopu oraz przetestowanie go.
3. Podłączenie serwomechanizmu.
4. Zbudowanie konstrukcji robota oraz stanowiska do testowania.
5. Opracowanie działania algorytmu.
6. Programowanie.
7. Testy jako gotowy projekt.

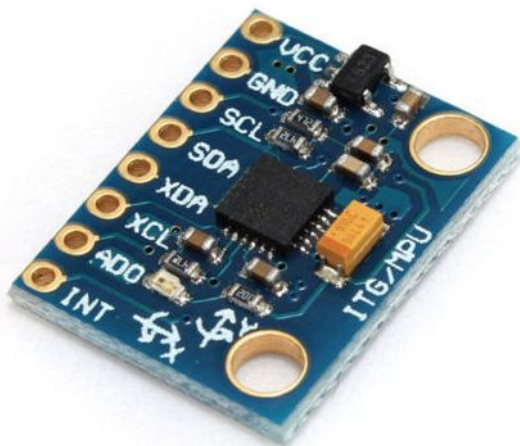
1. Zdobyć potrzebnych podzespołów
Serwomechanizm:



Serwo PowerHD AR-3603HB praca ciągła 360 stopni

Mocne serwo typu standard. Pracuje jako silnik z regulowaną prędkością. Zasilane napięciem 4,8 V - 6,0 V. Prędkość: 0,12 s./60°. Moment: 4,4 kg*cm. Wymiary: 40,5 x 20,0 x 38,0 mm. Masa: 36 g.

Żyroskop:



MPU-6050 3-osiowy akcelerometr i żyroskop I2C - moduł SparkFun

Czujnik do pomiaru przyspieszeń oraz prędkości kątowej w trzech osiach. Jest połączeniem 3-osiowego akcelerometru i żyroskopu. Charakteryzuje się prostą obsługą, komunikuje się poprzez magistralę I2C.

Arduino

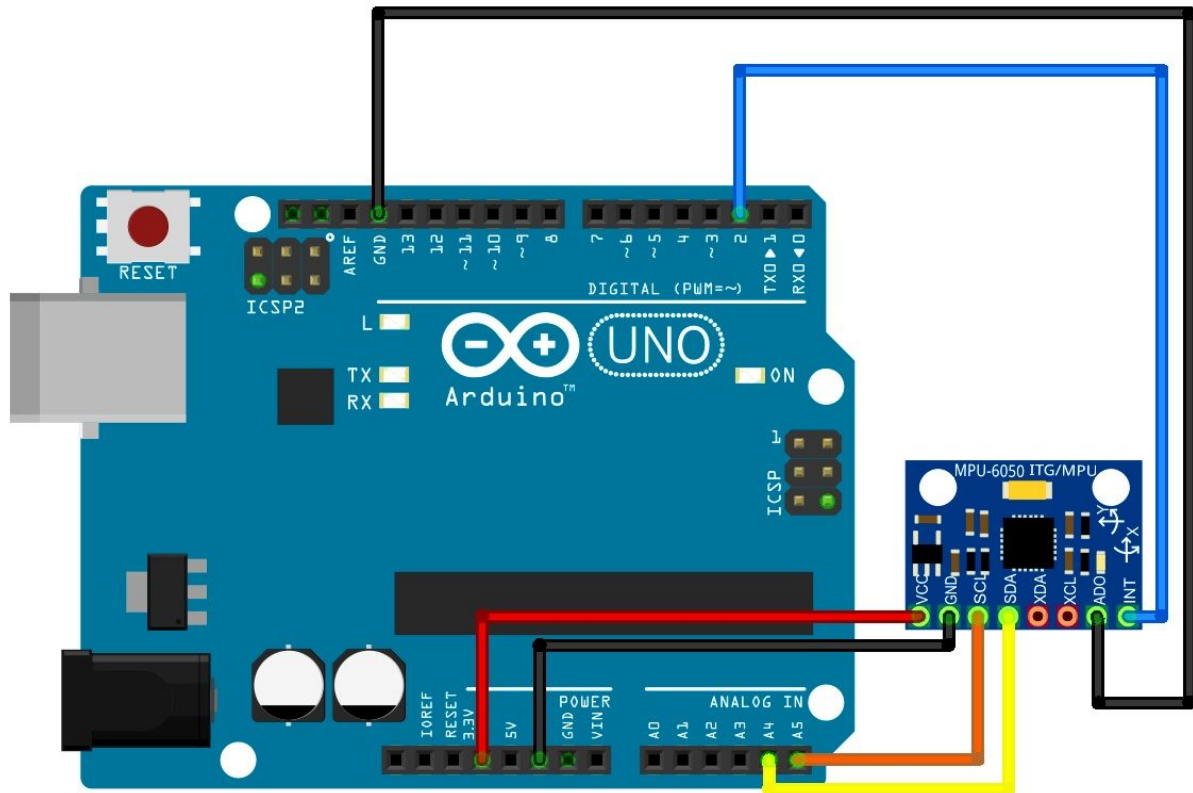


Arduino Uno Rev3

Popularny moduł z mikrokontrolerem AVR ATmega328 w wymiennej obudowie. Posiada 32 kB pamięci Flash, 2 kB RAM, 14 cyfrowych wejść/wyjść z czego 6 można wykorzystać jako kanały PWM, 6 wejść analogowych oraz popularne interfejsy komunikacyjne.

2. Podłączenie żyroskopu oraz przetestowanie go

Żyroskop **MPU-6050** został podłączony do Arduino w następujący sposób:



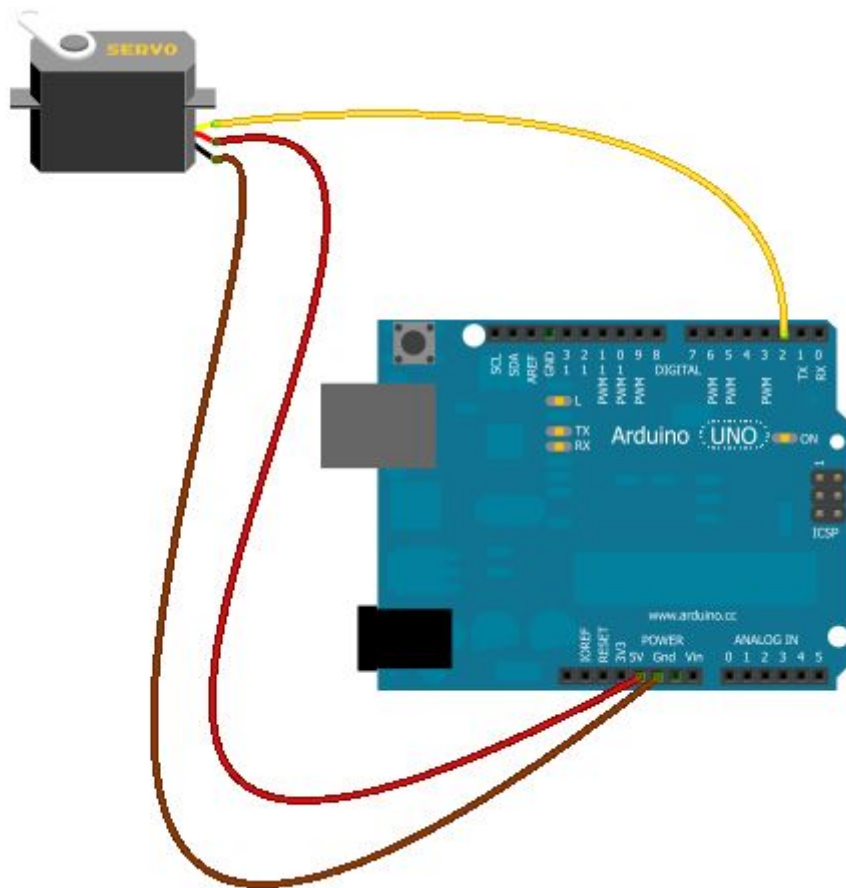
MPU-6050 komunikuje się z naszą płytką przy pomocy magistrali **I2C**. Dzięki czemu możemy podłączyć czujnik za pomocą 4 przewodów. Pisząc kod do tego czujnika skorzystamy z biblioteki **I2Cdev.h** oraz **MPU6050.h**.

Tworzymy obiekt typu **MPU6050** (w moim przypadku nazwany **mygyro**), służący do komunikacji z urządzeniem, oraz zmienne typu **int16_t** do których będą przypisywane wartości odczytane z akcelerometru. Wartości te odczytujemy za pomocą:

```
mygyro.getMotion6()
```

3. Podłączenie serwomechanizmu

Serwomechanizm **Servo PowerHD AR-3603HB** został podłączony w następujący sposób:



Do sterowania serwomechanizmem wykorzystuję bibliotekę **Servo.h**.
Tworzymy obiekt typu **Servo** (w moim przypadku nazwany myservo).
Pierwszy kod dotyczący serwomechanizmu (zrealizowany w setup):

```
myservo.attach()
```

W wyniku jej działania **Arduino** od teraz wie na którym wyjściu jest serwomechanizm.

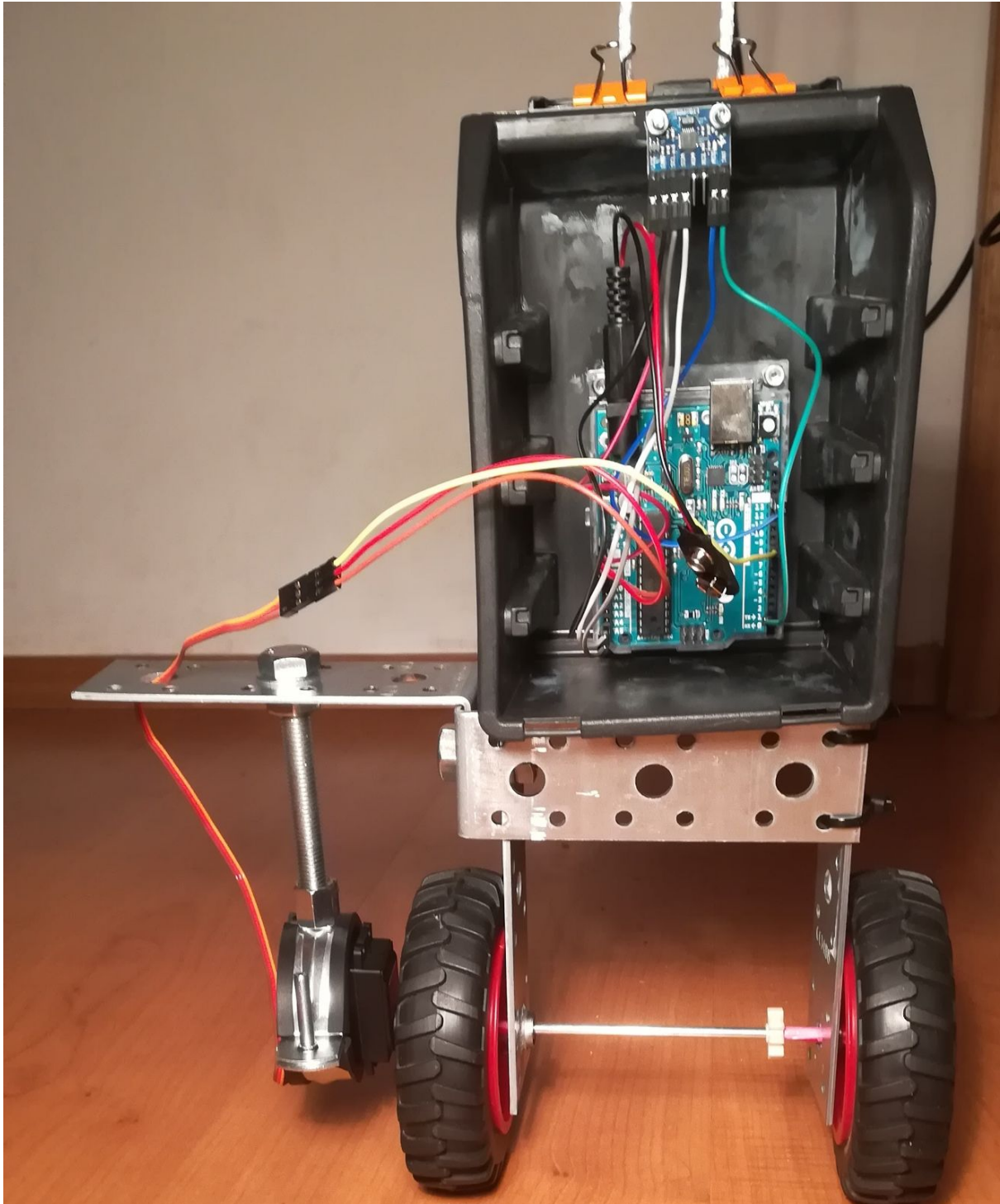
Obrót silnika uzyskujemy poprzez zmianę wartości argumentu funkcji (zrealizowany w loop):

```
myservo.write();
```


4. Zbudowanie konstrukcji robota oraz stanowiska do testowania.

Konstrukcje oparłem na częściach zdobytych w sklepie remontowo-budowlanym, które połączyłem w jedną kreatywną całość, spełniającą wymagania takie jak sztywność oraz ogólne zasady dotyczące budowania takich robotów.

Gotowa konstrukcja robota:



Podzespoły umieściłem w następujący sposób:

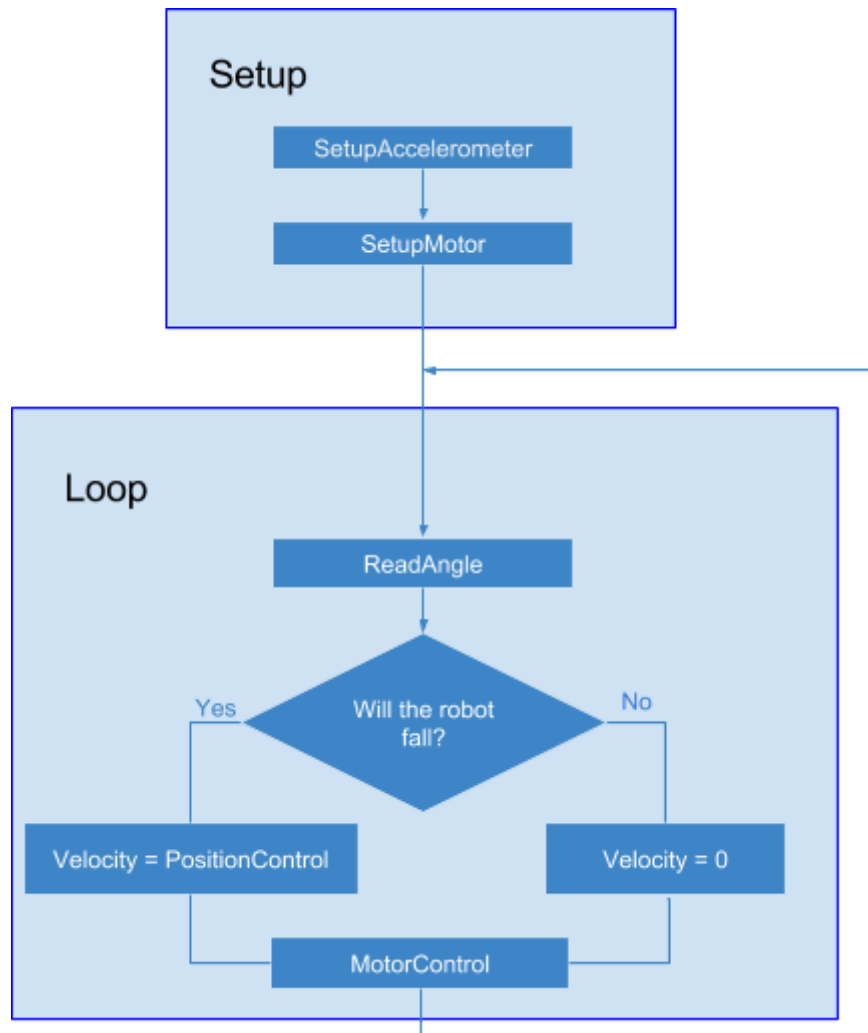
- Arduino - w środku plastikowej konstrukcji; jako najważniejszy element musi się znajdować w bezpiecznym miejscu, a jako element łączący wszystko w jedną całość musi się znajdować w miejscu równie oddalonym od pozostałych elementów
- Żyroskop - w najwyższym punkcie w celu dokładniejszego wyłapywania odchyłań konstrukcji
- Serwomotor - przy kołach robota

Do testowania działania programu potrzebowałem konstrukcji, umożliwiającej kontrolowanie przestrzeni poruszania się robota. W tym celu zaczepiłem pojazd za pomocą sznurków. Wraz z doskonaleniem algorytmu zwiększany będzie zakres poruszania się robota by w fazie końcowej uwolnić go i tym sposobem umożliwić swobodne poruszanie się.



5. Opracowanie działania algorytmu:

Algorytm przedstawiony w postaci schematu blokowego:



Bibliografia:

<http://www.instructables.com/>

<https://botland.com.pl/>

<https://diyhacking.com/>

<https://forbot.pl/blog/>

<https://create.arduino.cc>