

# Self-Balancing Robot

Date de contact: [Andrei-Lucian Bîrzîche](#)

## Introducere

Proiectul constă în realizarea unui robot pe două roți care reușește să se echilibreze singur.

De când am văzut prima oară un Hoverboard, am fost fascinat de modul în care se poate autobalansa. Iar dacă o echipă de ingineri cu master în sisteme de control automat au putut face așa ceva, m-am întrebat: "eu nu pot?". Desigur că nu, însă ceva la un nivel mai redus reprezintă în momentul de față scopul temei. Așa că am hotărât să urmez această idee și să pun în practică cunoștințele dobândite în cadrul laboratorului de PM și să realizez ceva, de care sunt sigur că mă voi bucura pe viitor.

## Descriere generală

Microcontroller-ul ATmega324 va primi date în mod continuu de la accelerometru și giroscop, fiind folosite pentru a cunoaște poziția curentă a robotului. Pe baza acestora decid dacă robotul se înclină spre față sau spre spate, iar cu ajutorul algoritmului PID voi încerca echilibrarea sa. Aceste informații ajung la driver-ul TB6612, iar în urma lor acesta va controla independent cele două motoare.

Schema bloc a proiectului:

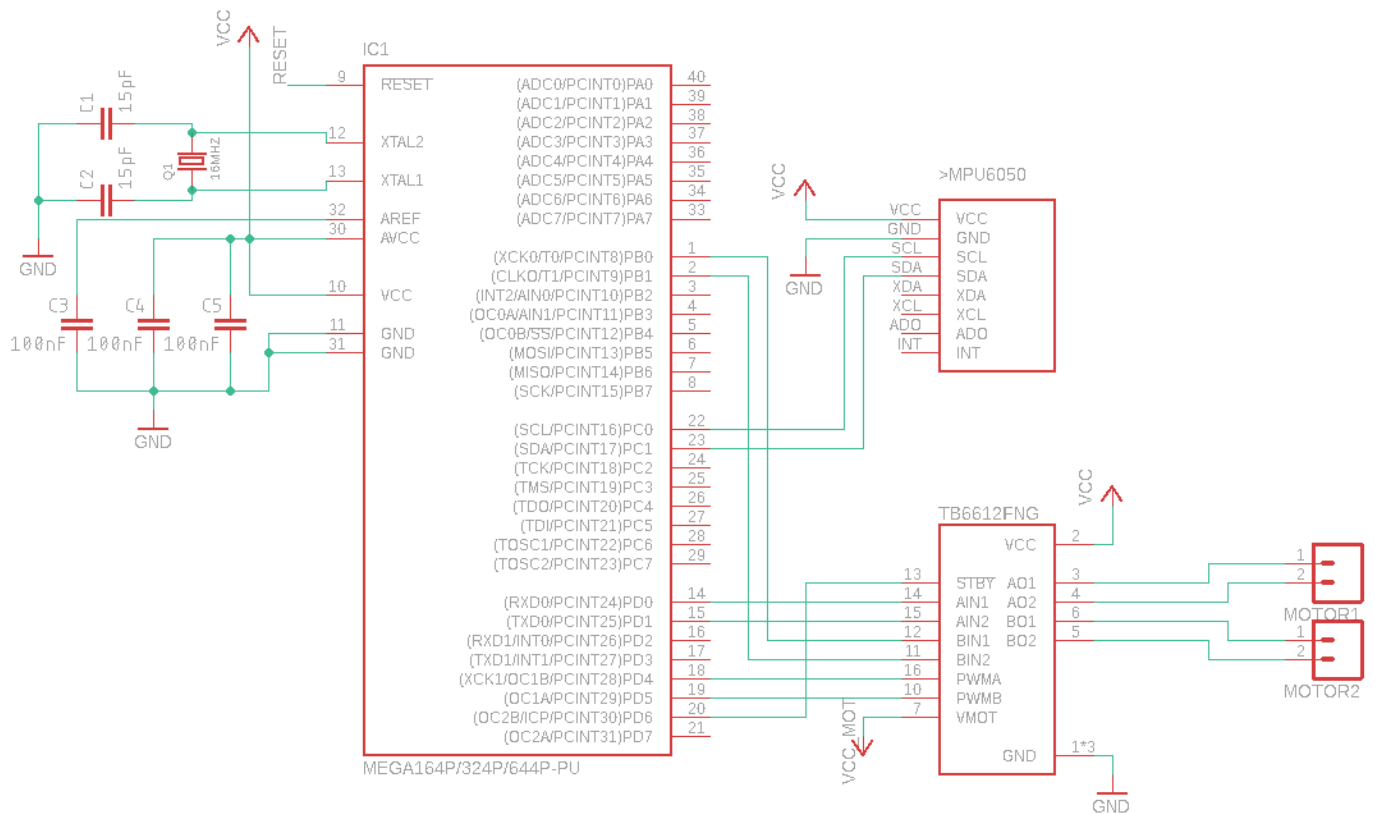


## Hardware Design

Lista de piese:

- Microcontroller ATmega 324
- Modul Accelerometru și Giroscop cu 3 Axe MPU6050
- Driver motoare Dual TB6612FNG
- Motor cu reductor și roată x2

Schema electrică:



## Software Design

Mediul de dezvoltare folosit:

- Sublime Text - pentru editare
- HIDBootFlash - pentru flash-uire
- avr-gcc - pentru compilare

Microcontroller-ul comunică cu senzorul IMU (accelerometru + giroscop) MPU6050, prin I2C, citind unghiul de înclinație pe axa Y. Pe baza acestuia, dar și a unor constante alese empiric, calculez variabilele P, I, D, necesare algoritmului PID, urmând să le însumez și să trimit rezultatul la driver-ul de motoare printr-un semnal PWM. Valoarea rezultată va fi folosită pentru a stabili direcția și viteza motoarelor.

## Rezultate Obținute



## Concluzii

Pot spune că rezultatul final a fost unul pe măsura timpului alocat. Am lăsat partea software pentru ultimele zile, iar acest lucru m-a costat, deoarece nu am reușit să fac debugging la valoarea unghiului de înclinație, pentru ca cele 3 constante ( $K_p$ ,  $K_i$ ,  $K_d$ ) să fie alese mai bine, drept urmare și perioada scurtă de timp în care robotul reușește să-și păstreze echilibrul.

## Download

[sbr\\_alb332ca.zip](#)

## Bibliografie/Resurse

<http://davidegironi.blogspot.ro/2013/02/avr-atmega-mpu6050-gyroscope-and.html#.WwQad0iFOUI> - Biblioteca pentru comunicarea I2C cu MPU6050; [Această pagină în format PDF](#) - Documentația.

From:

<http://ocw.cs.pub.ro/courses/> - **CS Open CourseWare**

Permanent link:

<http://ocw.cs.pub.ro/courses/pm/prj2019/abirlica/self-balancing-robot>

Last update: **2021/04/14 15:07**

