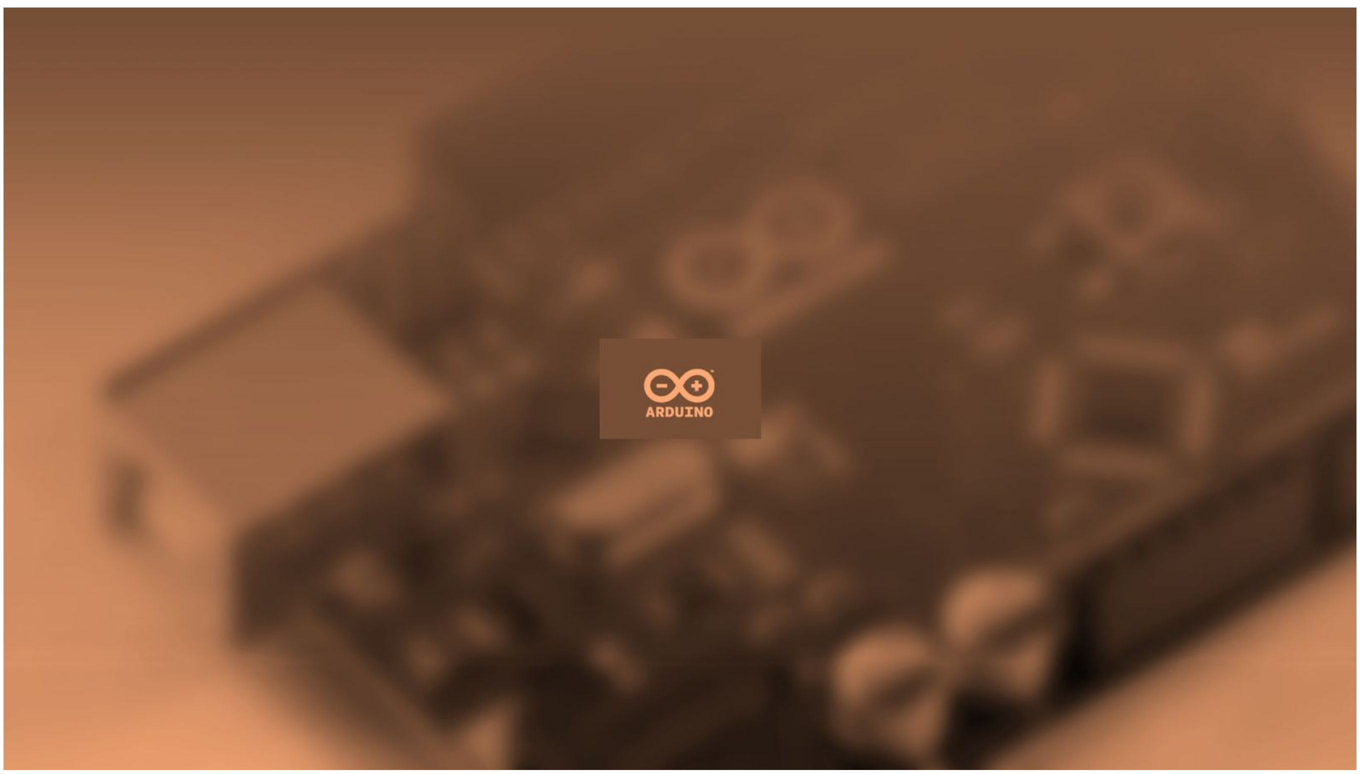
IED PROJECT

Two-Wheeled Self-Balancing Robot

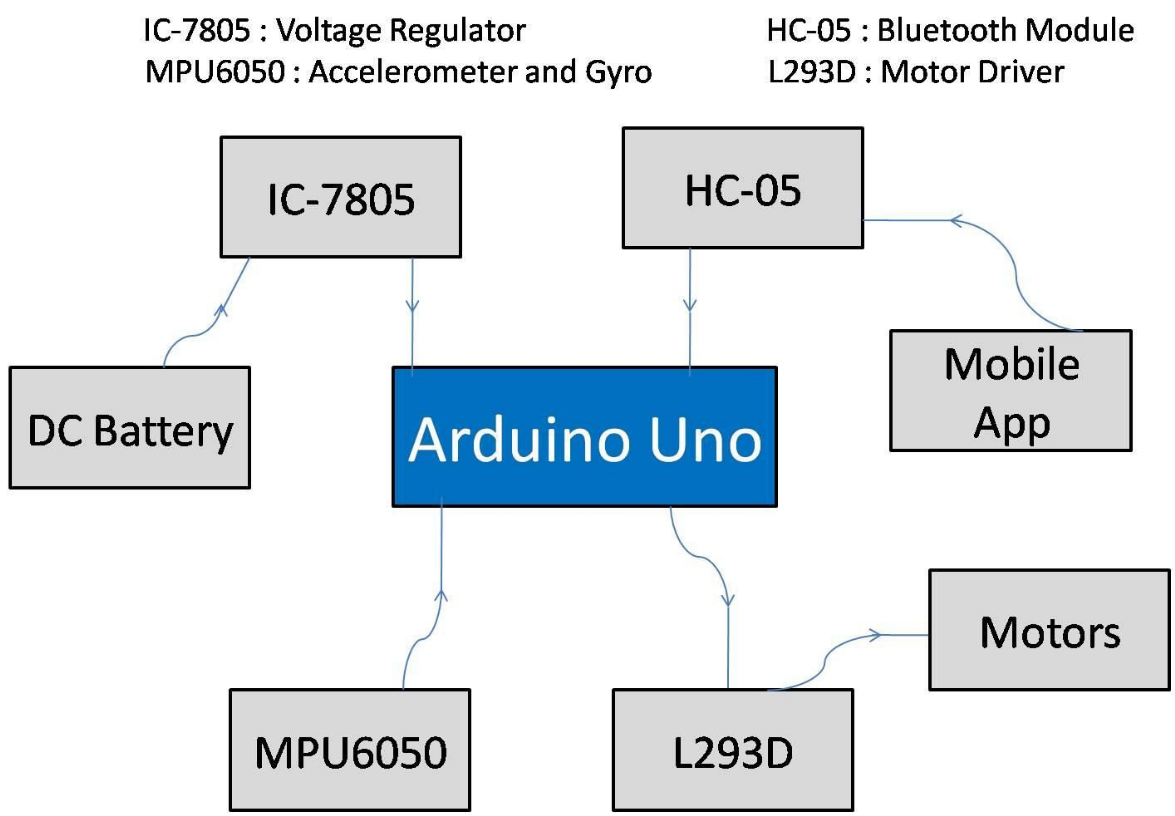


Aditya Adhikary (2015007)
Anannya Uberoi (2015014)
Lamha Goel (2015050)
Saksham Suri (2015082)

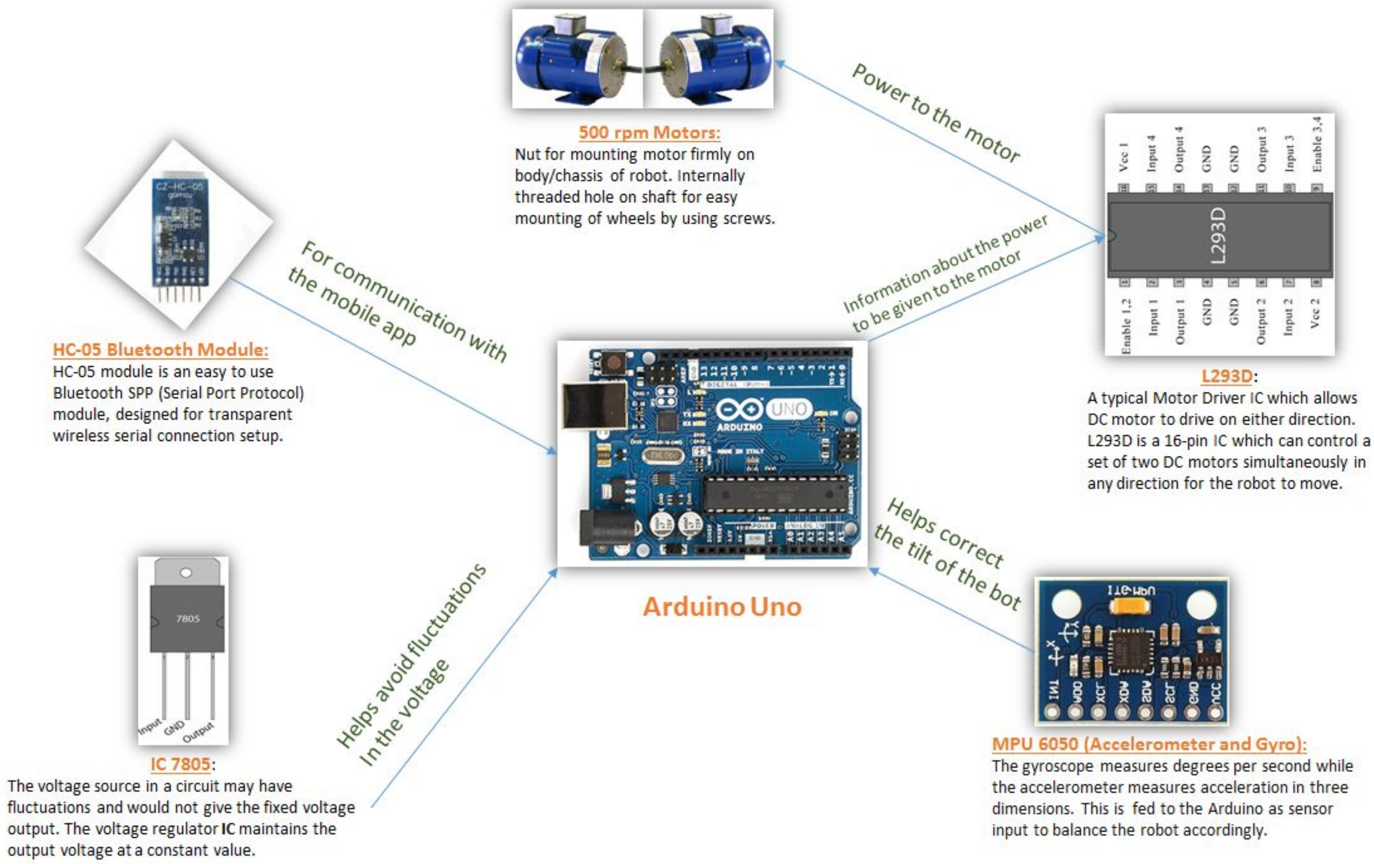
SCOPE:

We aim to build a self- balancing robot using Arduino Uno as the controller and a MPU6050 sensor to control the balance as it is moving. It can overcome the problem of maintaining balance while traversing a non-uniform terrain, with some load on it. This has handy applications as a part of intelligent trolleys in marts, hospitals etc. and guidance for the blind and disabled. Also, we aim to control it remotely using a Bluetooth device from a mobile or computer application.

BLOCK DIAGRAM OF THE PROJECT:



FUNCTIONAL DIAGRAM OF THE PROJECT:



LISTS OF SOURCES AND APPROXIMATE PRICES OF COMPONENTS:

1. UNO R3 Development Board (ATmega328P) with USB cable for Arduino - Rs. 500
2. Breadboard – Rs 100
3. MPU6050 Chip - Rs 450
4. HC-05 Bluetooth Module - Rs 400
5. 2x 500 rpm Motors - Rs 700
6. 9 V DC Batteries - Rs 400
7. L293D Motor Driver - Rs 300
8. Jumper Wires - Rs 80
9. Acrylic Sheet, Material for the chassis - Rs 500
10. Rubber wheels for Cars and Toys - Rs 400
11. IC-7805 Voltage Regulator - Rs. 30

Sources have not yet been finalized. We'll be purchasing most of the components from Amazon, Flipkart or EBay or from the local market.

INITIAL APPROACH TO THE PROBLEM:

The robot will be prevented from falling by giving acceleration to the wheels according to its inclination from the vertical. If the bot gets tilts by an angle, than in the frame of the wheels, the center of mass of the bot will experience a pseudo force which will apply a torque opposite to the direction of tilt.

It balances itself by taking measurements from sensors which are then summed as a feedback to microcontroller. Then microcontroller generates the required signal as a motor voltage, which is proportional to motor's torque.

We first researched about the functioning of the accelerometer and the gyro chip on the MPU6050 module. We also looked at some drawbacks of getting the data signal voltages from both, and possibly some filter algorithms which we can use to combine the data from both chips to predict the next position of the robot. The accelerometer output is prone to noise, whereas the Gyro values are much less noisy in the short term but will drift away from the true values in the longer term.
The two possible options for the filters systems are:

1. **Kalman Filter** - This is a set of mathematical equations that provides an efficient recursive computational means to estimate the state of a process, in a way that minimizes the mean of the squared error. The filter is very powerful in several aspect. It supports estimations of past, present, and even future states, and it can do so even when the precise nature of the modelled system is unknown.
2. **Complementary filter** - Using the accelerometer readings in the long term and the gyro readings in the short term, we can get a steady, accurate value for the angle of the pitch (or the rotation around the side-to-side axis). Such a filter is called a complementary filter.

We will use I2C protocol for communication between the Arduino board and the Bluetooth module and the MPU6050 module.

We also plan to implement a PID (Proportional, Integral, Derivative) controller which receives an error value. It tries to minimize the error by combining the past errors, present error and predicted future error. P represents the present error value of the system. I refers to integral, thus, maintains the sum of the previous errors. D-derivative accounts for possible future errors by calculating the current rate of change in error.

REFERENCES

1. http://www.instructables.com/id/Self-Balancing-Robot/
2. http://www.instructables.com/id/Chappie-Self-Balancing-Robot/?ALLSTEPS
3. https://youtu.be/ewYFrtir-Co
4. https://bayesianadventures.wordpress.com/2013/10/20/gyroscopes-accelerometers-and-the-complementary-filter/
5. http://home.iitk.ac.in/~vipgupta/self-balancing-bot