International Research Journal of Engineering and Technology (IRJET) e-ISSN: 2395-0056
Volume: 06 Issue: 04 | Apr 2019 www.irjet.net p-ISSN: 2395-0072

Self Balancing Robot using Arduino Uno

Maithilly Tripathi[1], Forum Bajariya[2], Suraj Vishwakarma[3], Yusuf Shaikh[4]

[1,2,3,4]*Student, Dept of Electronics and Telecommunication, Universal College of Engineering, Vasai, India*

---***---

Abstract - *An* A self balancing robot is creating a robot that is a replica of a human body. Traditional robots consisted of four wheels, were easily stabilized and comparatively bigger in size. A traditional robot uses four wheels and four motors for movement, while a self balancing robot uses only two wheels and motors for movement. A very famous application of the self balancing robot is the *Segway. Segway* is readily available in market since 2011 and is also termed as a "Human Transporter". It is used mostly to cover shorter distances. It is a self balancing robot where human weight and angle plays a vital role on controlling the movement of the *segway*. These types of robots are based on the physical theory of inverted pendulum. The system in itself requires active control in order to be stable. The main aim of this project is to use the self balancing robot interfaced with a camera for Search and Rescue purposes. Mapping, maneuvering and updating natural calamities, terror attacks, accidents explosions etc. To send the robot inside unpredictable and hard-to-access areas like unknown and narrow caves, where human intervention is not possible and is risky if tried blind eyed.

***Key Words*: Arduino Uno board,Raspberry Pi 3b, MPU6050 gyroscope and accelerometer,L293D Motor Driver, PI Camera**

1. INTRODUCTION

Two –wheeled or self balancing robot is an unstable dynamic system unlike other four wheeled stale robots that are in equilibrium state. By unstable, here, we mean that the robot is free to fall ahead or backward direction without any application of force. Self balancing means the robot balancing itself in an equilibrium state, 90 degrees upright position. This project works on the inverted pendulum concept. We are making use of Arduino Uno to build the self balancing robot. We are using the inertial measurement unit MPU6050 for measuring the current tilt angle. A PID controller will be able to control the pendulum angle. The Raspberry pi, pi camera will help us determine the surrounding conditions through live streaming and help in the search and rescue operations. Having a clear idea of the state of devastation prior to sending any human inside a natural calamity struck area is the sole purpose of this project. To provide important information about the surrounding is why we are using a two wheeled robot mounted with a camera and sensors which help him to move.

2. Hardware Details

2.1 Arduino Uno

Arduino is an open-source arrange used for structure devices adventures. Arduino contains both a physical programmable circuit board (consistently insinuated as a microcontroller) and a touch of programming, or IDE (Integrated Development Environment) that continues running on your PC, used to create and exchange PC code to the physical board. Arduino Uno is a microcontroller board dependent on the ATmega328P.It has 14 advanced information/yield pins (of which 6 can be utilized as PWM yields), 6 simple data sources, a 16 MHz quartz gem, a USB association, a power jack, an ICSP header and a reset catch. It contains everything expected to help the microcontroller; essentially associate it to a PC with a USB link or power it with an AC-to-DC connector or battery to begin.

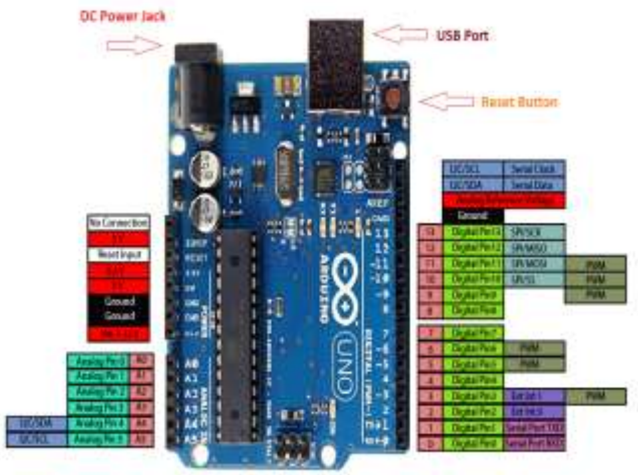

Figure 1 Arduino Uno Board

2.2 MPU6050

The MPU6050 is a Micro Electro-Mechanical Systems (MEMS) which comprises of a 3-hub Accelerometer and 3-hub Gyroscope inside it. This encourages us to gauge increasing speed, speed, introduction, uprooting and numerous other movement related parameter of a framework or article.

Figure 2 MPU6050

2.3 L293D Motor Driver

L293D is a run of the mill Motor driver or Motor Driver IC which permits DC engine to drive on either course. L293D is a 16-stick IC which can control a lot of two DC engines all the while toward any path. It implies that you can control two DC engine with a solitary L293D IC. Double H-connect Motor Driver incorporated circuit

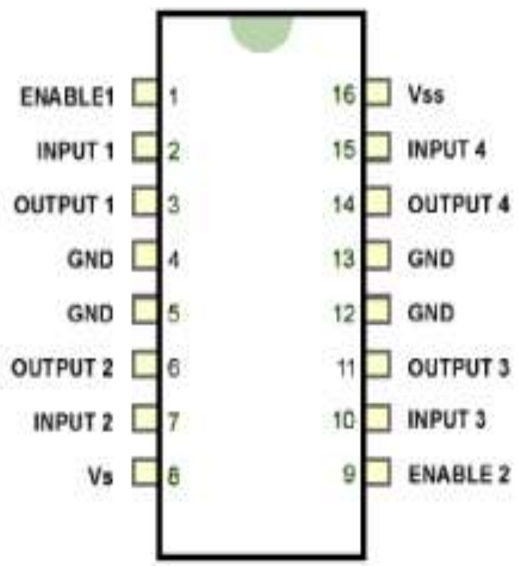

Figure 3 L293D Motor Driver

2.4 Raspberry Pi 3b

The Raspberry Pi display 3B comprises of 40 GPIO (General Purpose Input Output) stick. The primary motivation behind the GPIO pins is to associate I/O gadgets to the Raspberry pi. I/O gadgets, for example, Camera and Display as of now have explicit sloths on the raspberry pi board. So they don't have to GPIO pins to interface themselves to the board. There are distinctive elements of the 40 GPIO pins. Each stick has its own apacity.

Figure 4 Raspberry PI 3b

2.5 Pi Camera

The Raspberry Pi camera module can be utilized to take superior quality video, just as stills photos. It's anything but difficult to use for tenderfoots, yet has bounty to offer progressed clients in case you're hoping to grow your insight. There are bunches of models on the web of individuals utilizing it for time-slip by, moderate movement and other video cunning. You can likewise utilize the libraries we pack with the camera to make impacts.The module has a five megapixel fixed-center camera that bolsters 1080p30, 720p60 furthermore, VGA90 video modes, just as stills catch. It joins by means of a 15cm strip link to the CSI port on the Raspberry Pi. It very well may be gotten to through the MMAL and V4L APIs, and there are various outsider libraries worked for it, including the Pi camera python library.

3. Block Diagram and Working

The working of this project is divided between two controllers, the Arduino uno and the Raspberry pi. We use the Arduino uno mainly for the balancing of the robot. Whereas, the Raspberry Pi is used for the live streaming part.

The Arduino uno board with MPU6050 (gyroscope and accelerometer) and motor driver IC L239D are solely responsible for the balancing of the robot. The Arduino and the L298N Motor driver module is straightforwardly fuelled through the Vin stick and the 12V terminal separately. The on-board controller on the Arduino board will change over the information 7.4V to 5V and the ATmega IC and MPU6050 will be fuelled by it. The DC engines can keep running from voltage 5V to 12V. In any case, we will interface the 7.4V positive wire from battery to 12V info terminal of engine driver module. This will influence the engines to work with 7.4V. The accompanying table will list how the MPU6050 and L298N engine driver module is associated with Arduino.

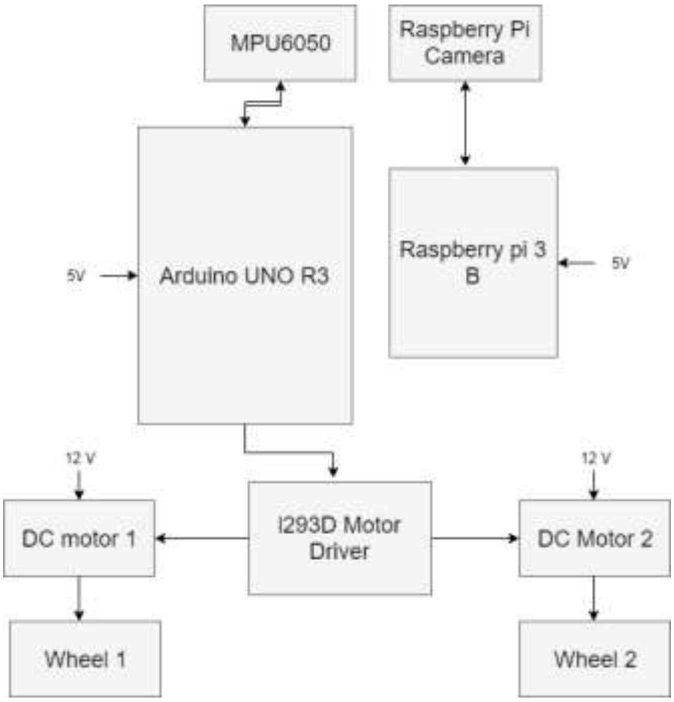

Figure 5 Block Diagram

The MPU6050 communicates with Arduino through I2C interface therefore we use the SPI pins A4 and A5 of Arduino. The two DC motors are connected to PWM pins D6, D9, D10 and D11. The need of connecting them to PWM pins is that we will be controlling the speed of the DC motor by varying the duty cycle of the PWM signals.

To keep the robot adjusted, the engines must neutralise the robot falling. This activity requires criticism and revising components. The input component is the MPU6050 (gyroscope and accelerometer), which gives both speeding up and pivot in each of the three tomahawks. The Arduino utilises this to know the redressing component is the engine and wheel mix.

This is done by the implementation of the PID controller, which is Proportional Integral Derivative Controller.

The proportional (P) term of the PID is based on the current angle difference from Point Zero. The integral (I) term of the PID is based on the current angle difference or error from point zero multiplied by the gain, which is accumulated over time. The integral control helps in balancing the robot if it is moving. The derivative (D) term of the PID is derived from the current rate of rotation.

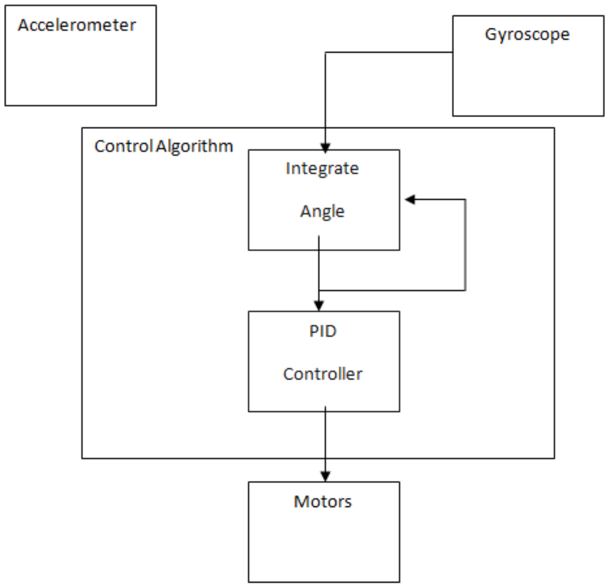

Figure 6 PID Algorithm

We need to check if the robot is inclining towards the front or towards the back utilizing the MPU6050 and after that if it's inclining towards the front we need to turn the wheels forward way and on the off chance that it is inclining towards the back we need to pivot the wheels in the invert bearing.

In the meantime we also need to control the speed at which wheels are pivoting. If the robot is somewhat muddled from focus position the wheels turn gradually and the speed increment as it makes tracks in an opposite direction from the middle position. To accomplish this rationale we utilize the PID calculation, which has the inside position as set-point and the dimension of bewilderment as the yield.

To know the present position of the robot we utilize the MPU6050, which is a 6-pivot accelerometer and gyrator sensor consolidated. So as to get a solid estimation of position from the sensor we have to utilize the estimation of both accelerometer and whirligig, in light of the fact that the qualities from accelerometer has commotion issues and the qualities from spinner will in general float with time. So we need to join both and get the estimation of yaw pitch and move of our robot, of which we will utilise just the estimation of yaw.

For the robot to balance properly, it should have a good centre of gravity. That can be achieved by properly placing all the components on the hardware of the robot. The PID controller values kp, kd, ki can be determined by trial and error method. Or by connecting you Arduino and bot by the serial port and testing the robot for desired angles. The values that we used are kp=80, kd=6 and ki=200.Not all the robots can work on the same values as mentioned.

Live streaming using Raspberry Pi and Pi camera:

We use Raspberry pi and a pi camera interfaced with it to record and live stream the surrounding of the robot, The pi and the pi camera will be mounted on the robot itself, which will act as eyes to the robot, The live streaming of from the pi camera will be seen on a monitor connected to the raspberry pi. The raspberry pi will work based on the python commands given to it through the terminal. We can live stream the recording on multiple monitors at the same time by sharing the IP address of the Raspberry pi.

4. CONCLUSIONS

The robot balances itself on two wheels without falling and with a slight jitter. The PID controller values are tried and tested values as follows:

Proportional(kp)=80,
Derivative(kd) =6 and
Integral(ki)=200.

Live streaming from the pi camera mounted on the robot can be successfully observed on the monitor.

ACKNOWLEDGEMENT

We as a group have taken many efforts for this project. However, it would not have been possible without the kind support and help of many individuals. I would like to extend my sincere thanks to them. I would like to thank our principle, Dr Ajoy Kumar Sir for inspiration and guidance. I am highly indebted to Prof. Kaveri Sawant Head of the Department, Electronics and Telecommunication for her guidance and constant supervision as well as providing us with necessary information regarding the project and also for his support in completing the project. I would like to express my special gratitude and thanks to Prof. Sandeep Dubey, our Project guide for providing me such attention and time and always helping me and my team members as well as guiding us throughout the project. I would like to express my gratitude towards my other team members for their kind co-operation and encouragement which helped me in completion of the project. My thanks and appreciation also goes to my classmates and friends who have always been supportive and helpful. Lastly, I would like to thank everyone who directly and indirectly helped me with this project.

REFERENCES

[1] MathWorks. 2012. Control Tutorials for MATLAB and SIMULINK, Inverted Pendulum. Available at: http://ctms.engin.umich.edu/CTMS/index.php.exampleInverted Pendulumsection-SystemModeling. [Accessed 21 April 2015

[2] Arduino Uno. 2015. Arduino - ArduinoBoardUno . Available at: http://www.arduino.cc/en/ main/arduinoBoardUno. [Accessed 22 April 2015]

[3] EMG30. 2010. Data sheet. Available at: http://www.robot–electronics.co.uk/htm/emg30.htm [Accessed 22 April 2015]

[4] Faragher, Ramsey. 2012. Understanding the Basis of the Kalman Filter Via a Simple and Intuitive Derivation. IEEE signal processing magazine.