

BALANCEBOT

SETUP GUIDE



CONTENTS

BalanceBot	04
How Does BalanceBot Stay Balanced?	04
How Does PID (Proportional, Integral, Derivative) Algorithm Work?	04
The Installation Steps	06
The Circuit Diagram	13
REX Main Board Pin Diagram	14
Arduino Code	15

BalanceBot

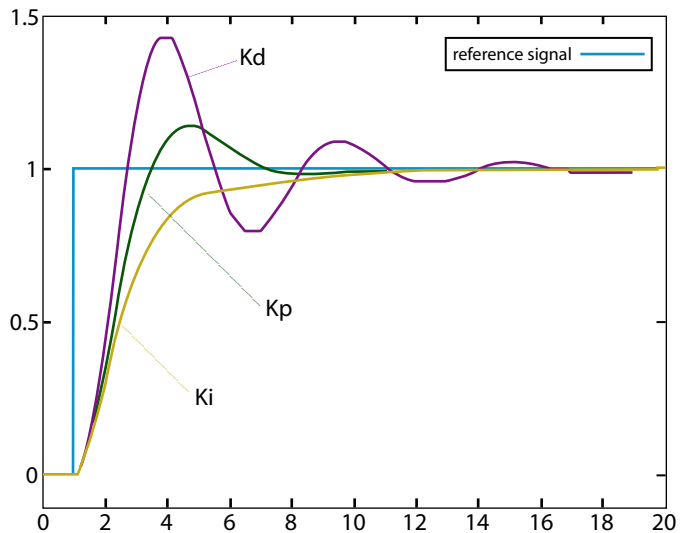
BalanceBot, REX kartı üzerinde bulunan MPU6050 sensörü sayesinde değişen çevresel faktörlere karşı dengede kalabilen REX robotudur.

Peki BalanceBot Nasıl Dengede Kalır ?

Various algorithms can be used to keep BalanceBot in balance. We will use the PID algorithm to keep BalanceBot in balance.

How Does PID (Proportional, Integral, Derivative) Algorithm Work?

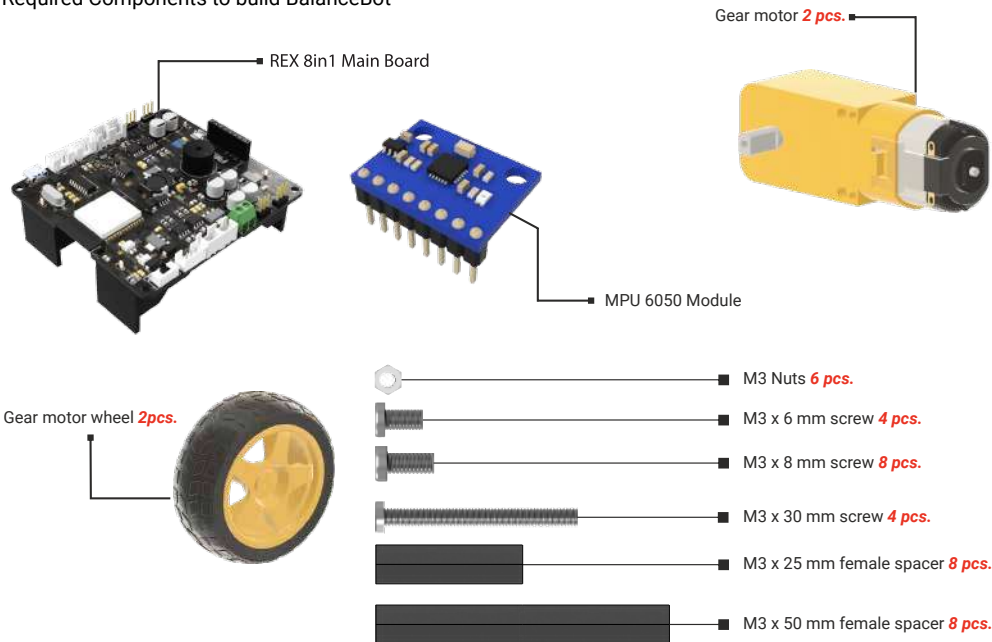
The difference between the data coming to the input signal with the feedback and the input signal is found. This difference generates the error. The error signal is sent to the PID controller and three different formulas are applied to the error signal with three different parameters. Then, it is sent back to the output signal. This process is looped until the error is minimized.



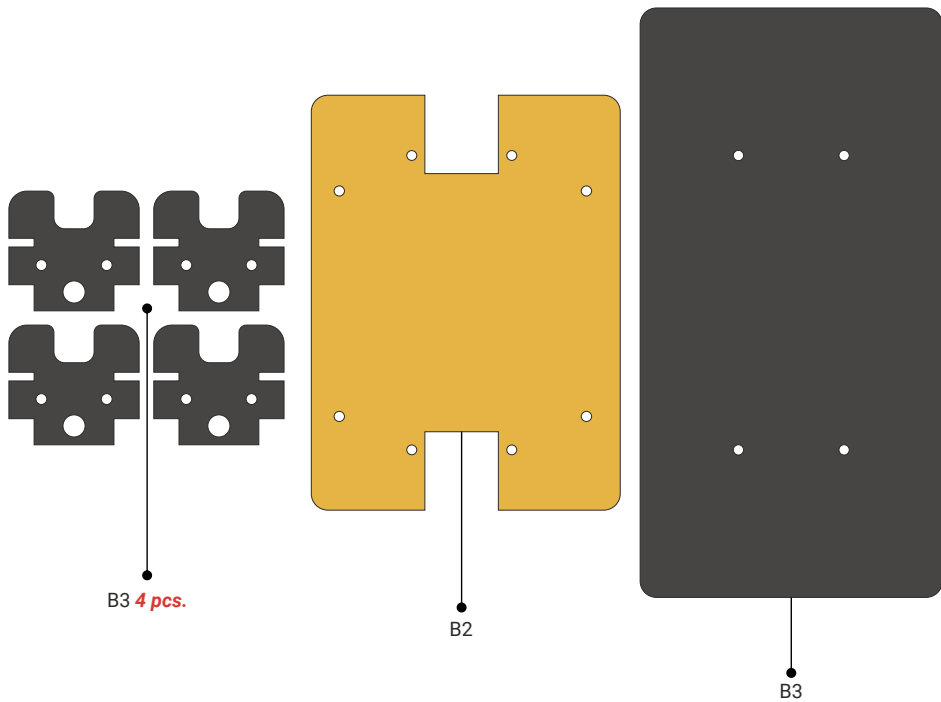
Environmental factors are calculated with the MPU6050(gyro/acceleration) sensor on the REX board and they are sent to the PID. The PID generates the output signal by performing the necessary operations to keep in balance the BalanceBot.

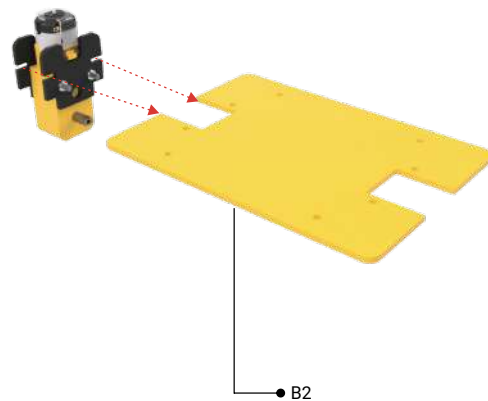
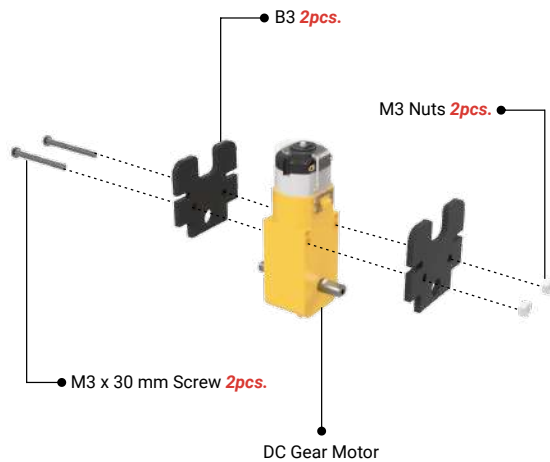
The Installation Steps

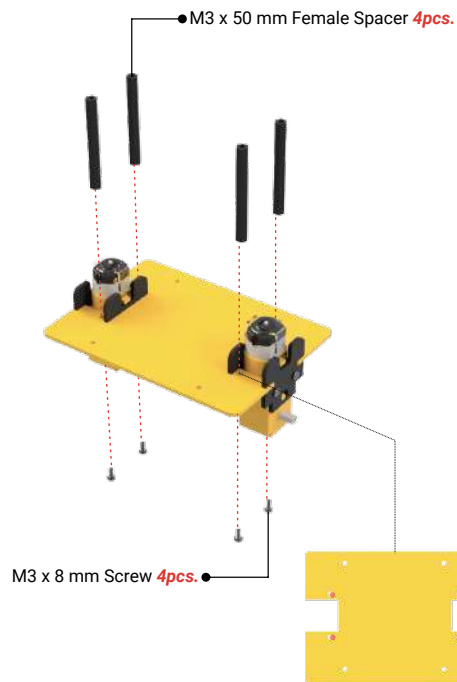
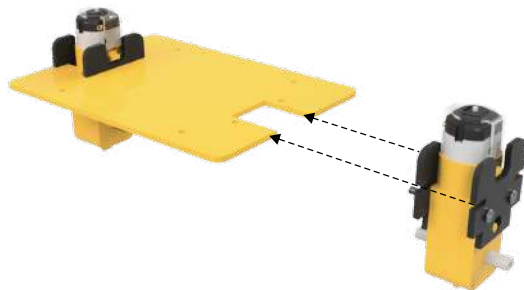
Required Components to build BalanceBot



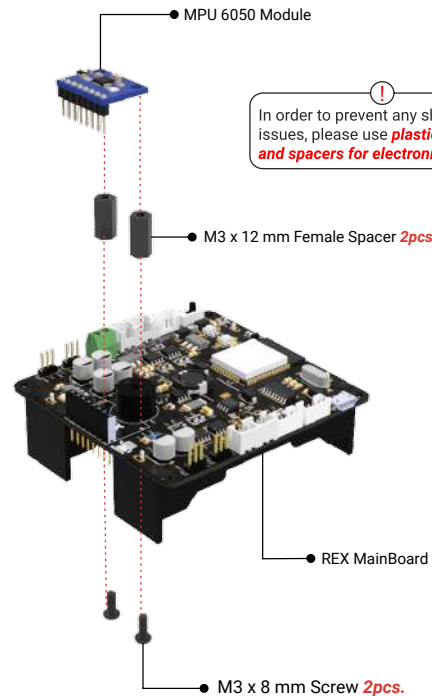
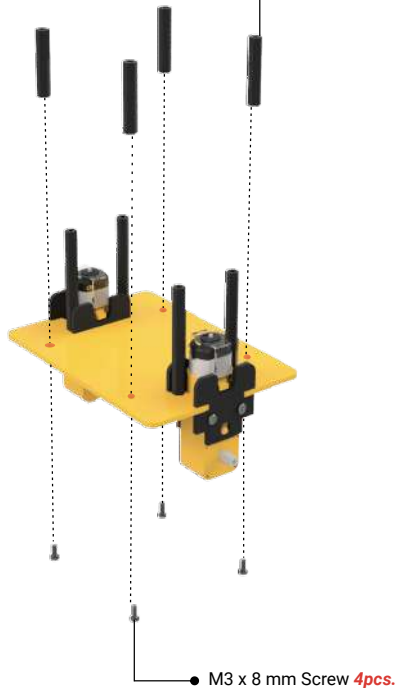
Required Parts



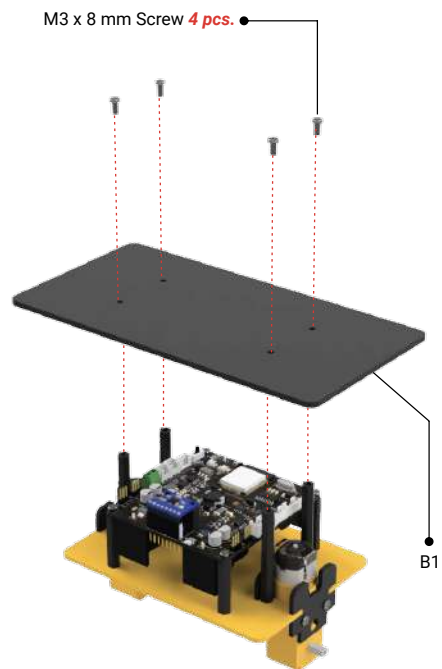
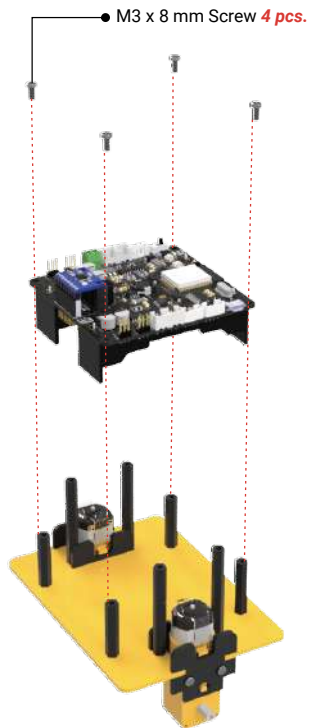


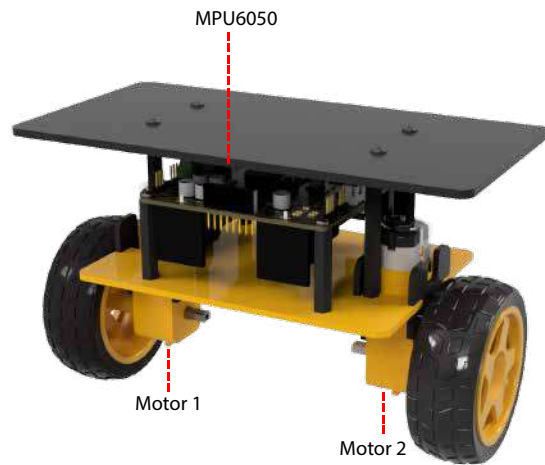


M3 x 25 mm Female Spacer **4pcs.**

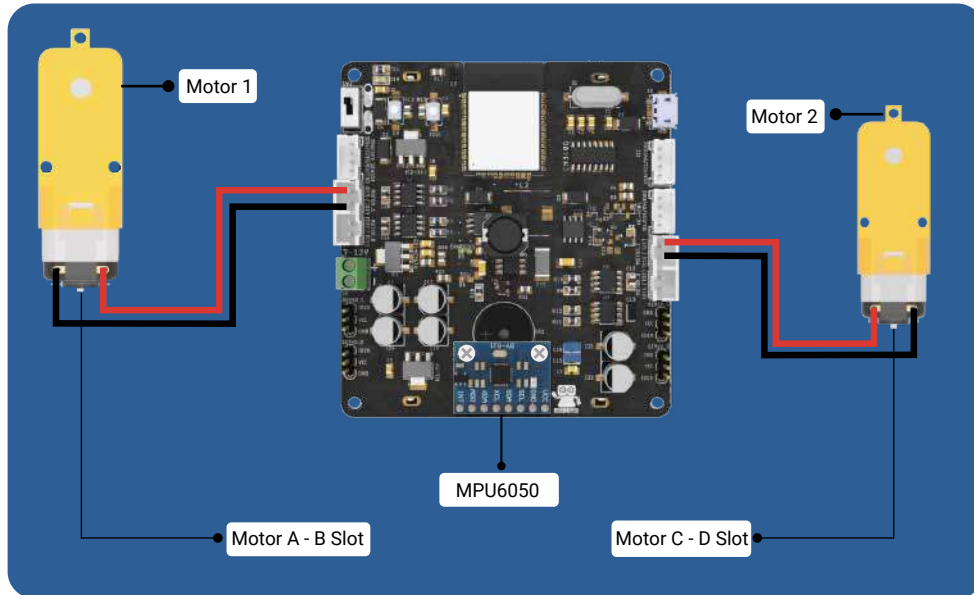


⚠
In order to prevent any short circuit issues, please use **plastic screws and spacers for electronic cards.**

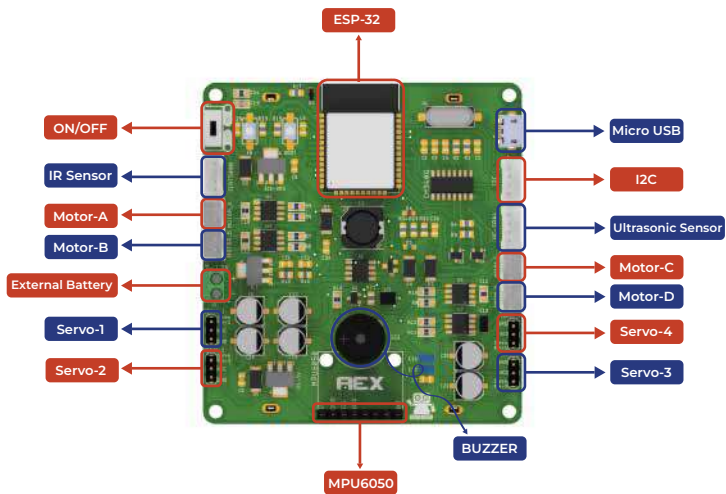




The Circuit Diagram



REX Main Board Pin Diagram



Arduino Code

```
1  //""""REX Mini Balance Bot""""
2  //Check the web site for Robots https://rex-rdt.readthedocs.io/en/latest/
3
4  #include "I2Cdev.h"
5  #include "PID_v1.h"
6  #include "MPU6050_6Axis_MotionApps20.h"
7  #include "Wire.h"
8
9  #define INTERRUPT_PIN 13
10
11  #define Motor_A1 16
12  #define Motor_A2 17
13  #define Motor_C1 23
14  #define Motor_C2 15
15
16  MPU6050 mpu;
17
18  bool dmpReady = false; // set true if DMP init was successful
19
20  uint8_t mpuIntStatus; // holds actual interrupt status byte from MPU
21  uint8_t devStatus; // return status after each device operation (0 = success, 10 = error)
22  uint16_t packetSize; // expected DMP packet size (default 16 42 bytes)
23  uint16_t fifoCount; // count of all bytes currently in FIFO
24  uint8_t fifoBuffer[64]; // FIFO storage buffer
25
26
27  // orientation/motion vars
28  Quaternion q; // [w, x, y, z] quaternion container
29  VectorFloat gravity; // [x, y, z] gravity vector
30  float ypr[3]; // [yaw, pitch, roll] yaw/pitch/roll container and gravity vector
31
32  //.....set following 4 values for your robot....
33  double setpoint = 175; //set the value when the bot is perpendicular to ground using serial monitor.(input value)
34  double Kp = 10; //Set this value first
35  double Kd = 0.20; //Set this value second
36  double Ki = 250; //finally set this value
37
38
39  double input, output;
40  PID pid(&input, &output, &setpoint, Kp, Ki, Kd, DIRECT);
41
42
43  volatile bool mpuInterrupt = false; // indicates whether MPU interrupt pin has gone high
```



Scan the QR code to go to the whole code and the necessary libraries.





GitHub

rbt.ist/rexgithub



Rex DOC

rbt.ist/rexrdt



Shop.robotistan/Rex

rbt.ist/rexrdt