

# Balancing Robot Manual

## Steps:

1. Buy items from bill of materials.
2. 3D Print .stl files from Thingiverse.
3. Wire it all up.
4. Set current limit on motor drivers.
5. Download Arduino code, download required libraries, and (Optional) edit MPU6050 library file.
6. Download Android app to control robot and create control panel on it.
7. Upload code to Teensy 4.0.
8. Assemble everything.
9. Test it.

See details below for more information.

Let me know if something is missing, incorrect, or if you need anything else.

## Bill of Materials:

- 3S Lipo of right size:  
[https://www.amazon.com/gp/product/B087CQR92J/ref=ppx\\_yo\\_dt\\_b\\_asin\\_title\\_o01\\_s00?ie=UTF8&psc=1](https://www.amazon.com/gp/product/B087CQR92J/ref=ppx_yo_dt_b_asin_title_o01_s00?ie=UTF8&psc=1)
- Or  
[https://www.amazon.com/gp/product/B0072AEYLW/ref=ppx\\_yo\\_dt\\_b\\_asin\\_title\\_o03\\_s00?ie=UTF8&psc=1](https://www.amazon.com/gp/product/B0072AEYLW/ref=ppx_yo_dt_b_asin_title_o03_s00?ie=UTF8&psc=1)
- Teensy 4.0:  
[https://www.amazon.com/gp/product/B08259KDHY/ref=ppx\\_yo\\_dt\\_b\\_asin\\_title\\_o04\\_s00?ie=UTF8&psc=1](https://www.amazon.com/gp/product/B08259KDHY/ref=ppx_yo_dt_b_asin_title_o04_s00?ie=UTF8&psc=1)
- MPU6050:  
[https://www.amazon.com/gp/product/B00LP25V1A/ref=ppx\\_yo\\_dt\\_b\\_asin\\_title\\_o04\\_s01?ie=UTF8&psc=1](https://www.amazon.com/gp/product/B00LP25V1A/ref=ppx_yo_dt_b_asin_title_o04_s01?ie=UTF8&psc=1)
- HC-05 BT Module:  
[https://www.amazon.com/gp/product/B07VL725T8/ref=ppx\\_yo\\_dt\\_b\\_asin\\_title\\_o05\\_s00?ie=UTF8&psc=1](https://www.amazon.com/gp/product/B07VL725T8/ref=ppx_yo_dt_b_asin_title_o05_s00?ie=UTF8&psc=1)
- Wires:  
[https://www.amazon.com/gp/product/B07GD2869Z/ref=ppx\\_yo\\_dt\\_b\\_asin\\_title\\_o06\\_s00?ie=UTF8&psc=1](https://www.amazon.com/gp/product/B07GD2869Z/ref=ppx_yo_dt_b_asin_title_o06_s00?ie=UTF8&psc=1)
- Breadboard Wires (Something like this):  
[https://www.amazon.com/MCIGICM-Breadboard-Jumper-Cables-Arduino/dp/B081GMJVPB/ref=sr\\_1\\_18?dchild=1&keywords=breadboard+wires&qid=1597876588&sr=8-18](https://www.amazon.com/MCIGICM-Breadboard-Jumper-Cables-Arduino/dp/B081GMJVPB/ref=sr_1_18?dchild=1&keywords=breadboard+wires&qid=1597876588&sr=8-18)
- M3 Screws and Nuts:  
[https://www.amazon.com/gp/product/B01J7NM9JA/ref=ppx\\_yo\\_dt\\_b\\_asin\\_title\\_o09\\_s00?ie=UTF8&psc=1](https://www.amazon.com/gp/product/B01J7NM9JA/ref=ppx_yo_dt_b_asin_title_o09_s00?ie=UTF8&psc=1)

- Breadboard:  
[https://www.amazon.com/gp/product/B00LSG5BJK/ref=ppx\\_yo\\_dt\\_b\\_asin\\_title\\_o09\\_s01?ie=UTF8&psc=1](https://www.amazon.com/gp/product/B00LSG5BJK/ref=ppx_yo_dt_b_asin_title_o09_s01?ie=UTF8&psc=1)
- Stepper Motor (x2):  
[https://www.amazon.com/gp/product/B07LF898KN/ref=ppx\\_yo\\_dt\\_b\\_asin\\_title\\_o00\\_s00?ie=UTF8&psc=1](https://www.amazon.com/gp/product/B07LF898KN/ref=ppx_yo_dt_b_asin_title_o00_s00?ie=UTF8&psc=1)
- Stepper Motor Drivers:  
[https://www.amazon.com/gp/product/B07QJVCFRZ/ref=ppx\\_yo\\_dt\\_b\\_asin\\_title\\_o00\\_s01?ie=UTF8&psc=1](https://www.amazon.com/gp/product/B07QJVCFRZ/ref=ppx_yo_dt_b_asin_title_o00_s01?ie=UTF8&psc=1)
- Step-Down Converter: <https://www.banggood.com/Hobbywing-3A-UBEC-5V-6V-Switch-Mode-BEC-For-RC-Models-p-915037.html?rmmds=myorder>
- XT-60 Connectors: [https://www.amazon.com/MCIGICM-Female-Bullet-Connectors-Battery/dp/B07DVKL42/ref=sr\\_1\\_8?crid=2WXW32645U3EL&dchild=1&keywords=xt60+connectors&qid=1597876895&srefix=xt60+c%2Caps%2C203&sr=8-8](https://www.amazon.com/MCIGICM-Female-Bullet-Connectors-Battery/dp/B07DVKL42/ref=sr_1_8?crid=2WXW32645U3EL&dchild=1&keywords=xt60+connectors&qid=1597876895&srefix=xt60+c%2Caps%2C203&sr=8-8)
- Wheels (I used some I had lying around the house, this product may or may not be the same. You may have to 3D design and print your own wheels instead):  
[https://www.amazon.com/Training-Reinforced-Prevent-Breaking-Bending/dp/B07RV5T18G/ref=sr\\_1\\_4?encoding=UTF8&c=ts&dchild=1&keywords=Bicycle+Training+Wheels&qid=1598560966&s=outdoor-recreation&sr=1-4&ts\\_id=1266043011](https://www.amazon.com/Training-Reinforced-Prevent-Breaking-Bending/dp/B07RV5T18G/ref=sr_1_4?encoding=UTF8&c=ts&dchild=1&keywords=Bicycle+Training+Wheels&qid=1598560966&s=outdoor-recreation&sr=1-4&ts_id=1266043011)

**Additional requirements:** Soldering iron, solder, heat shrink, 3D printer, 100uF Capacitors x2, Android Phone, additional wiring (If needed).

## Android Phone App for control:

### Bluetooth Electronics – Google Play Store

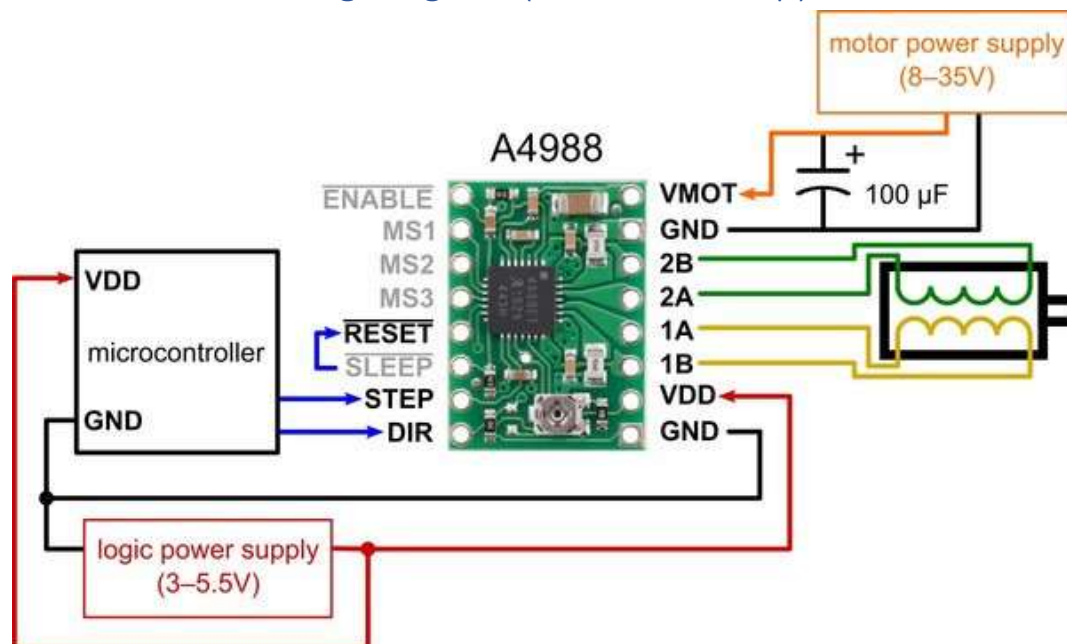
- Edit Panel 9
- Add Joystick pad (Third from left under pads)
- Add On/Off Button Switch
- For Joystick: Min value 0, Max value 1500, select magnitude and angle (deg), start with (<), end with (>), send continuously while pressed, repeat every 150ms.
- For On/Off button: Turn on text (<On>), Turn off text (<Off>), repeat every 100ms, current switch state (Off).

## Wiring:

Wiring		Notes:
3S Lipo (+)	UBEC Input (+), Motor Drivers (VMOT)	*See motor driver notes for additional help
3S Lipo (-)	UBEC Input (-), Motor Drivers (GND)	*See motor driver notes for additional help
Motor Wires	Motor Drivers 2B-1B	*See motor driver notes for additional help
Motor Drivers (RESET)	Motor Drivers (SLEEP)	*See motor driver notes for additional help
100 uF Capacitor (+)	Motor Drivers (VMOT)	*See motor driver notes for additional help
101 uF Capacitor (-)	Motor Drivers (GND)	*See motor driver notes for additional help
Teensy 4.0 (3V)	Motor Drivers (VDD), MPU6050 (VCC)	*See motor driver notes for additional help
Teensy 4.0 (GND)	Motor Drivers (GND), MPU6050 (GND)	*See motor driver notes for additional help

Right Motor Driver (DIR)	Teensy 4.0 (Pin 5)	*See motor driver notes for additional help
Right Motor Driver (STEP)	Teensy 4.0 (Pin 6)	*See motor driver notes for additional help
Left Motor Driver (DIR)	Teensy 4.0 (Pin 7)	*See motor driver notes for additional help
Left Motor Driver (STEP)	Teensy 4.0 (Pin 8)	*See motor driver notes for additional help
Right Motor Driver (MS1)	Teensy 4.0 (Pin 9)	*See motor driver notes for additional help
Right Motor Driver (MS2)	Teensy 4.0 (Pin 10)	*See motor driver notes for additional help
Left Motor Driver (MS1)	Teensy 4.0 (Pin 20)	*See motor driver notes for additional help
Left Motor Driver (MS2)	Teensy 4.0 (Pin 21)	*See motor driver notes for additional help
UBEC - Connect 5V Jumper on module		
UBEC Output (+)	Teensy 4.0 (5V), HC-05 (VCC)	
UBEC Output (-)	Teensy 4.0 (GND), HC-05 (GND)	
MPU6050 (SCL)	Teensy 4.0 (Pin 19)	
MPU6050 (SDA)	Teensy 4.0 (Pin 18)	
MPU6050 (INT)	Teensy 4.0 (Pin 2)	
HC-05 (RXD)	Teensy 4.0 (Pin 14)	
HC-05 (TXD)	Teensy 4.0 (Pin 15)	

## Motor Driver Wiring Diagram (Additional Help)



The diagram above shows the wiring needed. The Teensy has the logic power supply built-in so just assume the VDD and GND wires are going straight to the motor drivers as described in the wiring table.

**Motor Driver Details:** <https://www.pololu.com/product/1182> (One used in project from bill of materials is basically the same but has max amp draw of 1A for motor).

**MAKE SURE TO TURN CURRENT LIMITING DIAL TO THE RIGHT AS FAR AS IT GOES EASILY (BE GENTLE).**  
**This is to set the current limit to 1A so the motors have enough power.**

## Code:

Use the attached Arduino file for the code (Ignore most of the comments, they are left-over from the MPU6050 example code).

Must also download the Teensyduino add-on library as well as the I2C library:

I2C library: <https://github.com/jrowberg/i2cdevlib> (Read the README)

Teensyduino: [https://www.pjrc.com/teensy/td\\_download.html](https://www.pjrc.com/teensy/td_download.html) (Follow instructions)

You also need to calibrate your MPU6050. Use the IMU Zero example included with the I2C library to determine offsets and then add them into the code where it says to do so.

## Removing MPU6050 motion drift:

(If you do not perform the following step then you must comment out the line that has setMyVar in it)

You can also edit the MPU6050\_6Axis\_MotionApps20.h file to help remove the MPU6050 drift that occurs when driving. This step is not necessary but it can help improve the driving performance of the robot. Go to wherever your Arduino file is then >libraries>MPU6050>MPU6050\_6Axis\_MotionApps20.h. You can just edit it with any text file editor. Open it up and find the following line:

```
uint8_t MPU6050::dmpGetYawPitchRoll(float *data, Quaternion *q, VectorFloat *gravity) {
```

There should be two identical lines like this. Go the second one and where it says data[2] = something, change it to:

```
data[2]= atan2(gravity -> y - ( cos(aadj+(PI/2)) * cos(aadj+(PI/2)) / sin(aadj+(PI/2)) ) , gravity -> z +  
cos(aadj+(PI/2)) );
```

Go to the first instance of the previous line and paste the following above the #ifdef statement above that line:

```
void setMyVar(float p_value){ aadj=p_value; }
```

Next, find this line:

```
#define MPU6050_DMP_UPDATES_SIZE 47 // dmpUpdates[]
```

And paste the following below it on the next line:

```
float aadj;
```

What all of this does is basically subtracts the expected x - acceleration of the robot from the actual x-acceleration MPU6050 readings. This makes it so that there is basically no x-acceleration and results in the angle calculation being more accurate after movement.