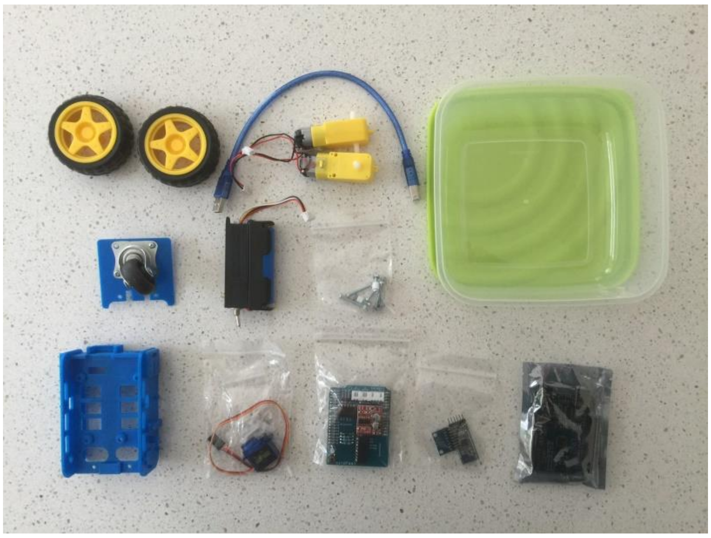
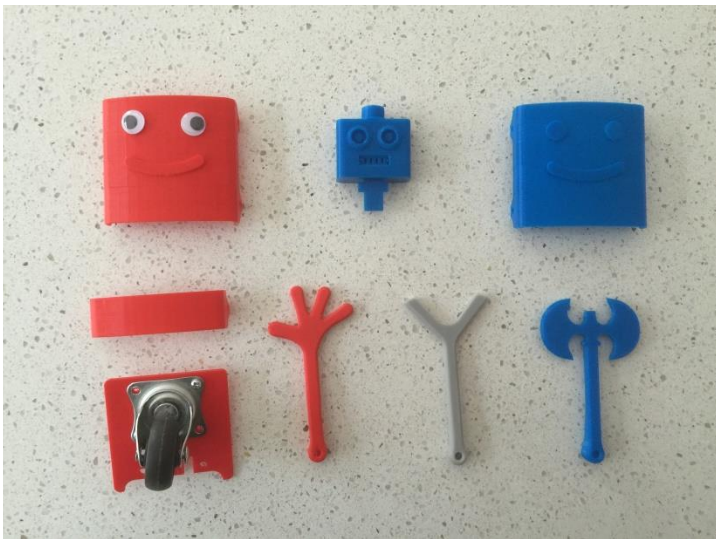
Lets Make a Robot

Ready to have some fun – lets go and make a robot !

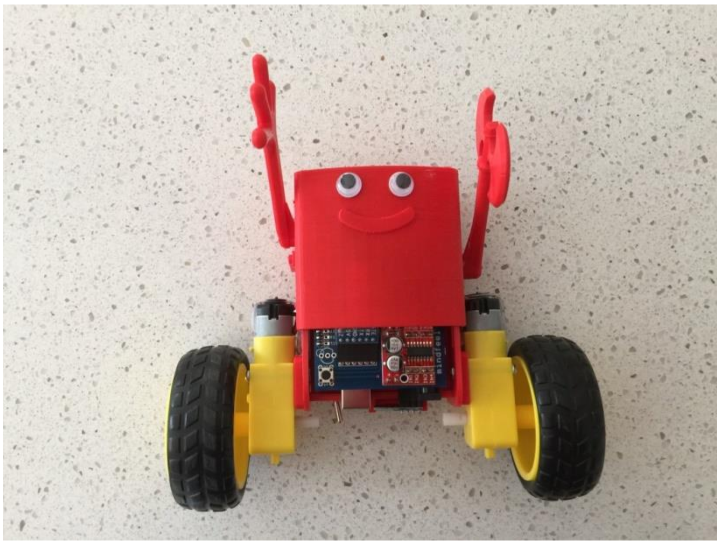


These are the parts for the base robot – all of these parts are included in your pack.

Note the container in the top right corner was used for the after school program where kits were made up for the children and supplied to them in the container.

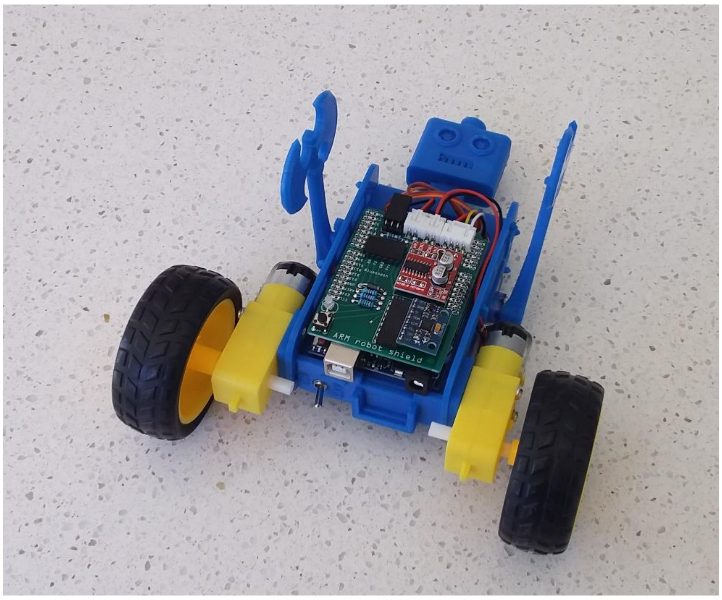


The great thing is, it is easy to make accessories for the robot – these are available in TinkerCAD for you to 3D print or you can create your own.

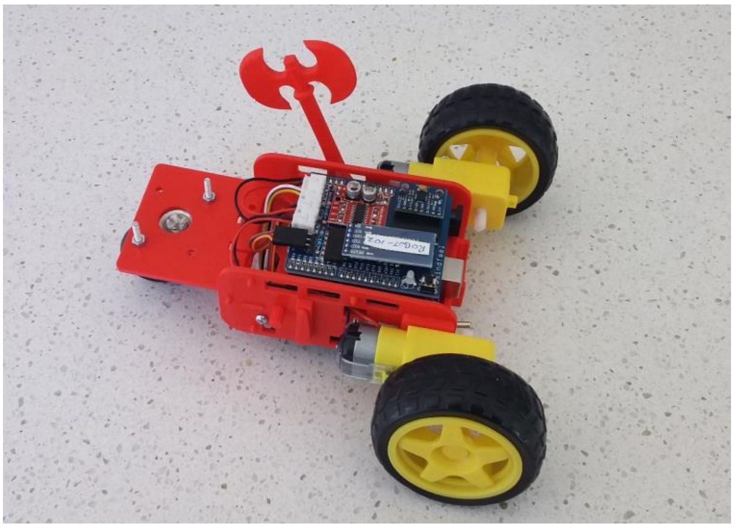


Here's some examples of how the robot can look:

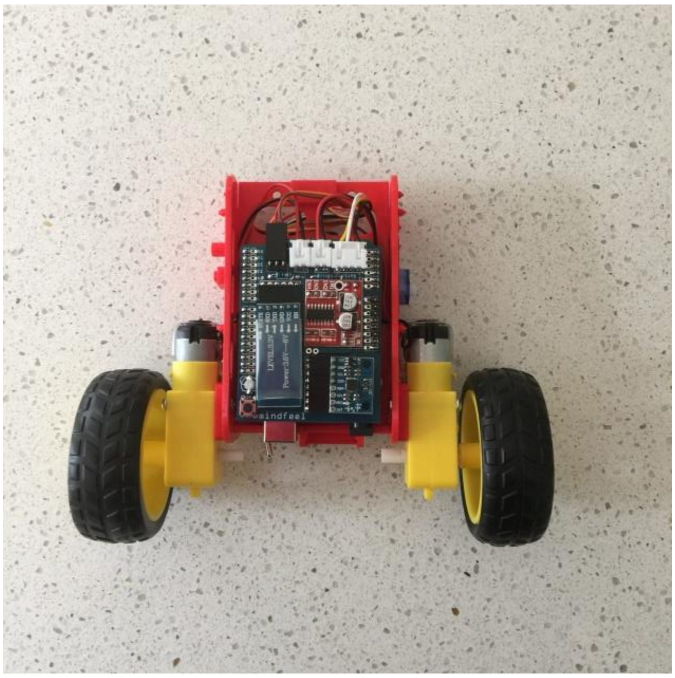
Mr Slick - 3D print your own covers and faces



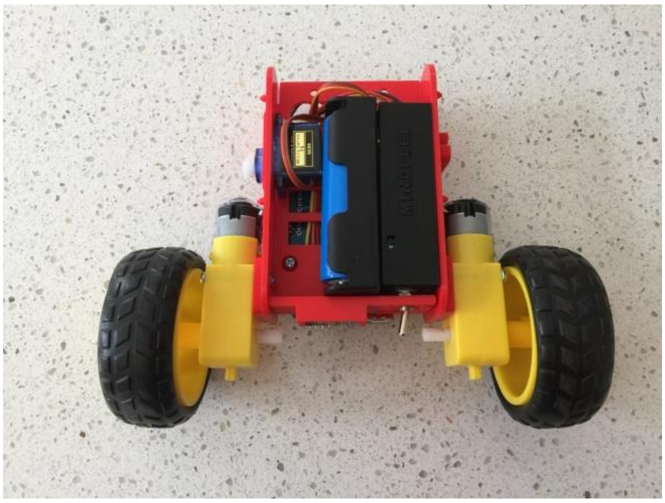
Mr Serious - 3D print your own robot heads



Mr Speedy - insert the 3rd wheel extension to become a speedy three wheeler.



Front



Back

This is how the finished base robot looks – front and back.

On the front side you can see the robot control shield which is plugged into an Arduino Uno underneath.

On the reverse side, the large black item is the lithium rechargeable battery power pack. It uses a standard USB charger (as used by most Android phones or PC) to recharge.

A charger is not included, however you can charge from your PC USB port or most Android phone chargers are suitable (it requires a 5W 1Amp charger, which is the same as most Android phones). Chargers can also be bought inexpensively from the likes of the Warehouse.

The Robot is controlled by a smartphone using a free Bluetooth app.

This robot uses a Lithium-Ion battery. It is important to look after these types of batteries by:

- Charge the battery before the LED on the robot shield turns RED.
- Do not remove the battery from its holder because you may damage the battery.
- Do not physically damage the battery.
- Do not expose to water (e.g. do not use the robot outside when wet)
- Do not expose the battery to heat. If the battery is hot, turn off the robot and take it outside.
- Do not use the Li Ion battery if it has not been charged for more than 6 months.

Robot Control Shield

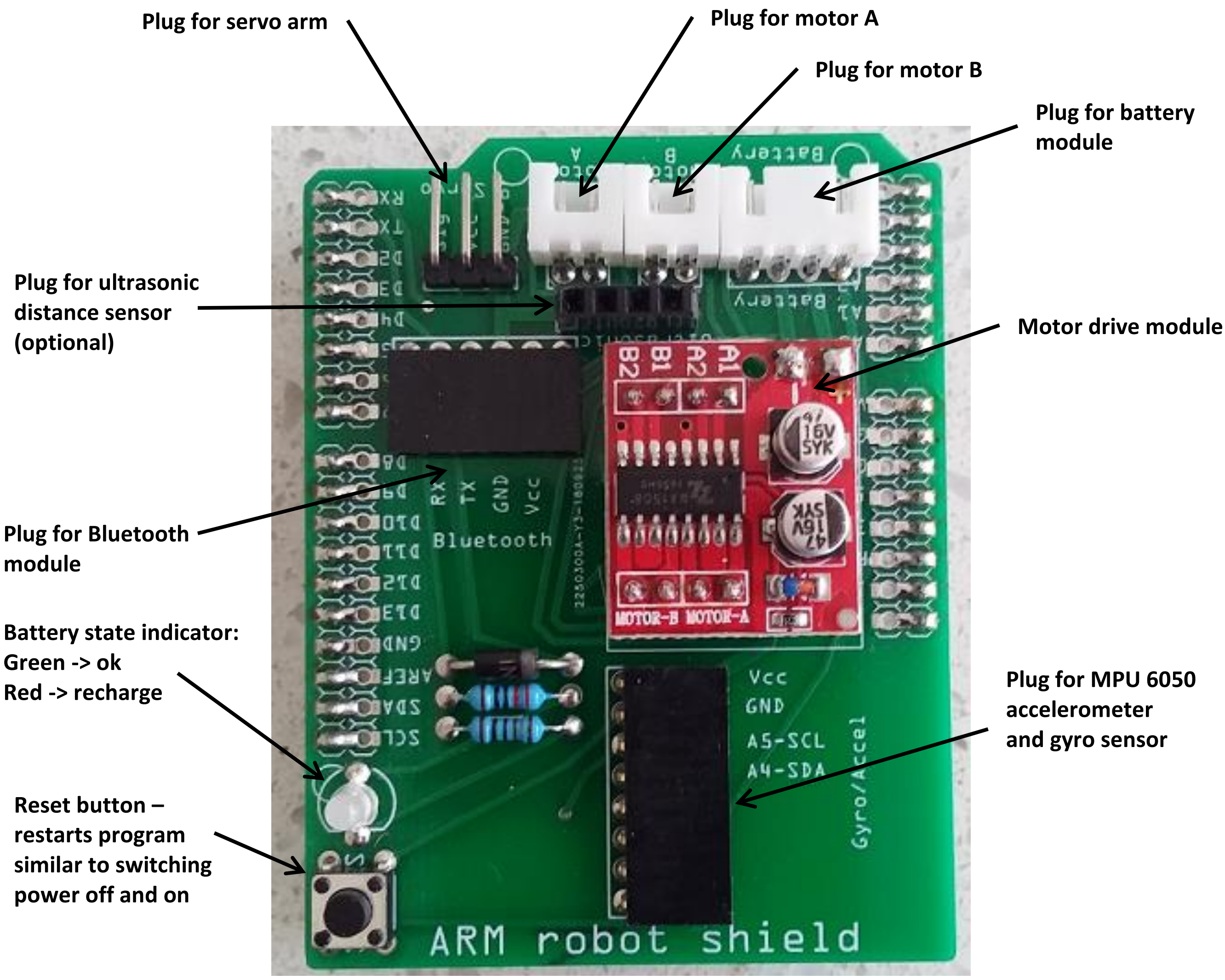
The robot uses a shield (circuit board) which plugs on top of the Arduino Uno card.

The key components on the robot control shield are shown in the diagram below.

Take special note of:

- The battery state indictor – this is an LED which can glow REB or GREEN. It indicates the charge state of the Li Ion battery - when RED recharge the battery.

- The reset button. This restarts the software sketch in the Arduino. It is often more convenient to press the reset button than to switch the power off and on to the robot.



Coding Exercises

As you construct the robot you can do programming exercises along the way if you want to. These exercises along with explanations are available on Arduino Create – the links below take you to the Arduino Create exercises – you can then open and save the exercise in your Arduino Create login.

To upload sketches to the robot make sure your phone is not connected to the robot by Bluetooth - a Bluetooth connection prevents uploading occurring.

Although generally not needed, the pin for the Bluetooth module is 123456.

Exercise 1: Arduino Basics –blink the LEDs on the robot control shield red and green.
You can do this exercise after Step (3) in the construction.
https://create.arduino.cc/editor/murcha/77bd0da8-1baf-4eb0-84b0-a369dce7666c/preview

Exercise 2: Gyro Sensor – getting familiar with gryos and accelerometers.
You can do this exercise after Step (4) in the construction.
You need to use the "Serial Monitor", with baud rate set to 115200.
https://create.arduino.cc/editor/murcha/46c50801-7e67-4b1f-96aa-9b97652ca1bc/preview

Exercise 3: Bluetooth Link – establish a Bluetooth link, use RemoteXY smart phone app to turn on and off the LEDs on the robot control shield.
You can do this exercise after Step (5) in the construction.
https://create.arduino.cc/editor/murcha/b29cfff6-a81c-41a4-bbfb-2b96b16da351/preview

Exercise 4: Ultrasonic distance sensor (optional) – getting familiar with the ultrasonic sensor.
You can do this exercise after Step (5) in the construction.
You need to use the "Serial Monitor", with baud rate set to 115200.
https://create.arduino.cc/editor/murcha/96e51fb2-6ab3-477b-89b2-426092f63ee7/preview

Exercise 5: Servo-mechanism – getting familiar with the servo mechanism and moving the arm, use RemoteXY smart phone app to control the angle of the servo arm.
You can do this exercise after Step (8) in the construction.
You need to use the "Serial Monitor", with baud rate set to 115200.
https://create.arduino.cc/editor/murcha/6e492462-b0df-4197-ae3b-4a267891061b/preview

Exercise 6: Drive motors – getting familiar with motors, run the drive motors forward and backwards.
Needs the battery pack to be switched on.
You need to use the "Serial Monitor", with baud rate set to 115200.
https://create.arduino.cc/editor/murcha/617cf6fc-18d4-487e-8c90-e8e20077df89/preview

Exercise 7: Basic Car – build a simple three wheel car (robot with 3rd wheel attachment), we use RemoteXY smart phone app to control the car.
You can do this at the same point in the construction as above.
Needs the battery to be switched on and insert the 3rd wheel attachment.
https://create.arduino.cc/editor/murcha/beadea02-2265-4a10-ae79-c1fe94842417/preview

Exercise 8: Full balancing robot – the code for the full balancing / three wheel robot.
Use the smart phone app "RemoteXY" to control the robot.
https://create.arduino.cc/editor/murcha/c0c055b6-dc3c-4f4c-b9fc-aa002d017b9d/preview

Exercise 9: Line Tracing robot (note: needs a modified 3 wheel attachment). It is possible to add two line tracing sensors, and use the ultrasonic plug to connect the line tracing sensors to the robot. Note, the sensors are connected to digital pins D2 and D8. (note you need a special front wheel attachment with line following sensors to run this sketch).
https://create.arduino.cc/editor/murcha/093021f1-1ef2-4fdd-a8bd-6a79a9acea25/preview

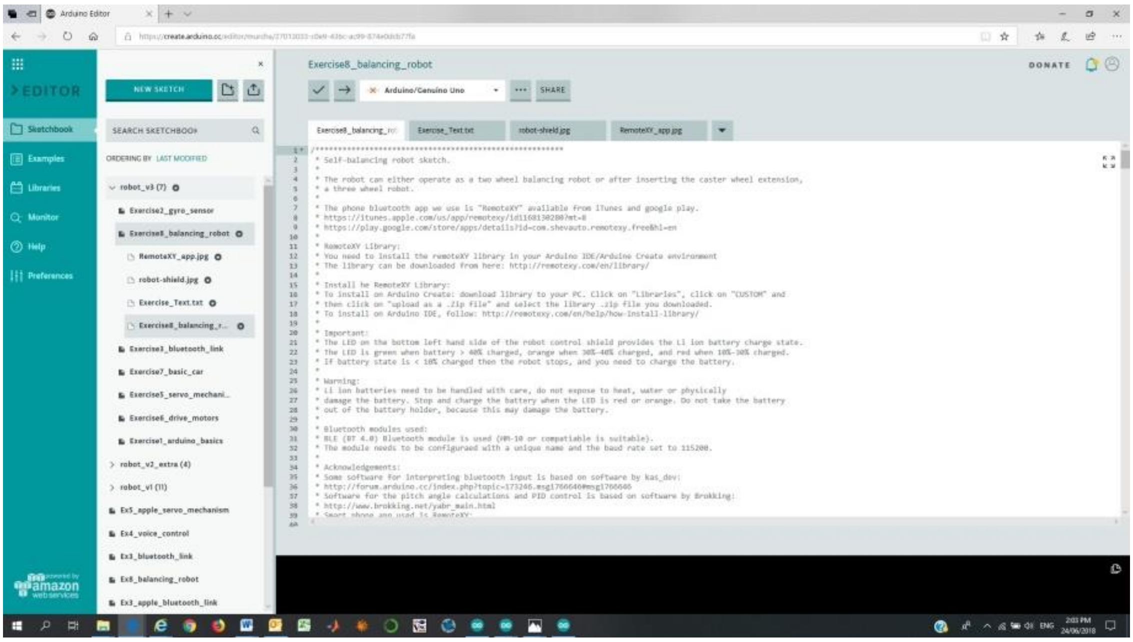
Exercise 10: Bluetooth Control. Using Bluetooth and a phone app (RemoteXY) to control the robot LEDs and the servo-mechanism. In this exercise students learn about Bluetooth, how to use a phone app to control real world things and learn about LEDs and servo-mechanisms.
https://create.arduino.cc/editor/murcha/c0d17e13-9a11-4d7d-ae1f-fb183260be42/preview

Exercise 11: Hand following robot. This sketch uses the ultrasonic sensor to determine how far away your hand is from the robot sensor and the robot moves backwards or forwards to follow your hand position. You can connect via Bluetooth using RemoteXY to display the distance measurements from the ultrasonic sensor.
https://create.arduino.cc/editor/murcha/be21c181-a891-4374-8445-fe3fac5b3eba/preview

Exercise 12: Line Tracing with Colour Sensor Robot (note: needs a modified 3 wheel attachment). This is a line tracing robot with a colour sensor to control the direction at an intersection. The colour sensor is a TCS34725 using I2C interface. The sketch has been designed to detect R, G, B colours. (note you need a special front wheel attachment with line following sensors to run this sketch).
https://create.arduino.cc/editor/murcha/00e91332-cc37-4ace-9453-7509029de616/preview

LOFI Blocks Sketch – to use the LOFI Blocks app upload this sketch into the robot.
You can then programme the robot using programming blocks similar to SCRATCH, , see https://lofiblocks.com/en/ for details. The LOFI Block app is available for both IOS (iPhone) and Andriod phones.
https://create.arduino.cc/editor/murcha/b2e6d9ce-2ff5-480c-9bfd-fecb1519a21f/preview

Arduino Create IDE



Arduino Create is a cloud based Arduino IDE (Integrated Development Environment). As mentioned on the previous page, the coding exercises are stored on the Arduino Create IDE, and the links on the previous page take you to these exercises.

If you haven't already, create a login to Arduino Create (its free): go to Arduino Create https://create.arduino.cc/ and click on "Arduino Web Editor", and click on "create a new account".

Once you have created a login, you can click on the coding exercise links on the previous pages and then click on the link "ADD TO MY SKETCHBOOK". This stores the code in your login environment and you can upload or edit the code as you wish. Alternatively, if you prefer you could copy and paste the code from the Arduino Create IDE into the PC based Arduino IDE and upload and edit the code there.

If when you login there's a yellow banner saying you need to download Arduino drivers, click on the banner and follow the instructions – this will download the driver for the Arduino board. You may also need to install the USB2.0-Serial driver available from:
http://www.wch.cn/download/CH341SER_ZIP.html and select CH341SER.EXE

Install the RemoteXY Library:

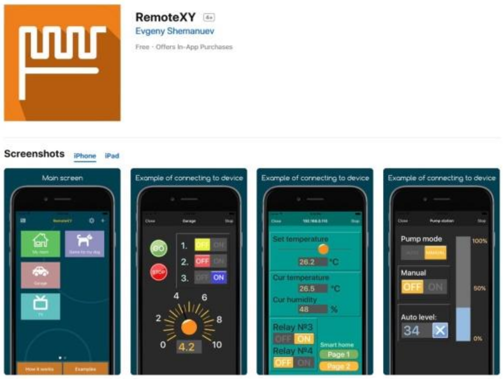
For Exercise 8 Full balancing robot sketch, you need to install the RemoteXY library to the IDE you are using. The library can be downloaded from here: http://remotexy.com/en/library/

To install on the Arduino Create IDE: first download the RemoteXY library to your PC, click on "Libraries" in Arduino Create, then click on "CUSTOM" and then click on "upload as a .Zip file" and select the library .zip file you downloaded to your PC.

To install on the PC based Arduino IDE, follow these instructions:
http://remotexy.com/en/help/how-install-library/

RemoteXY phone App

Exercises 3, 5, 7 and 8 use Bluetooth communication with a smart phone. The app used on the smartphone is RemoteXY available free from iTunes (for iPhones) and Google Play (for Android phones).



Search for “RemoteXY” and install the RemoteXY app.

Although generally not needed, the pin for the Bluetooth module is 123456.

The RemoteXY app has a web site which includes a GUI editor, which allows you to create your own graphical user interface (GUI) for your phone or tablet to control an Arduino. The editor also generates the code required to add to your Arduino sketch. More information about the RemoteXY app is available here: http://remotexy.com/en/help/

https://itunes.apple.com/us/app/remotexy/id1168130280?mt=8
https://play.google.com/store/apps/details?id=com.shevauto.remotexy.free&hl=en

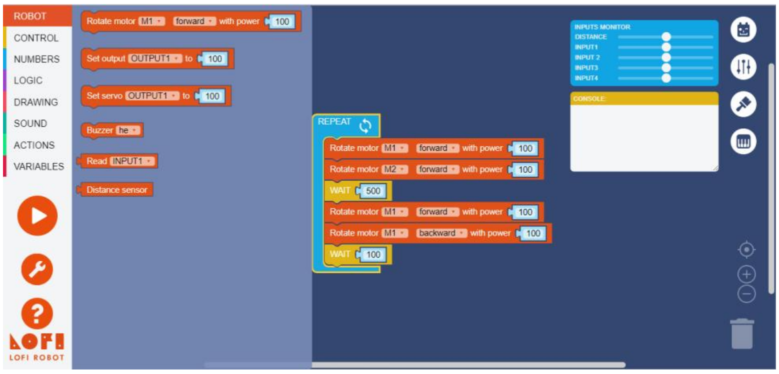
The RemoteXY app supports both the older version of Bluetooth (Bluetooth 2.0) and the new version of Bluetooth (Bluetooth 4.0 BLE).

iPhones only support Bluetooth BLE, while most Android phones support both Bluetooth 2.0 and BLE.

The robot is supplied with a Bluetooth BLE module. If you have an older Andriod device it may only support the older version of Bluetooth, Bluetooth 2.0. If you have an older Andriod device that does not support Bluetooth BLE then you need to obtain a Bluetooth 2.0 module for the robot - your supervisor may have some available.

LOFI Blocks – based on SCRATCH

The balancing robot operating as a three wheel robot can be programmed using LOFI Blocks. LOFI Blocks is a block programming environment similar to SCRATCH.
https://lofiblocks.com/
http://www.lofirobot.com/lofi-blocks/#more-1843
http://www.lofirobot.com/edubox/scratchx-chrome/

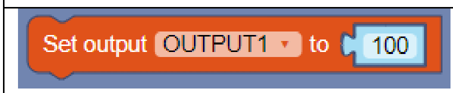


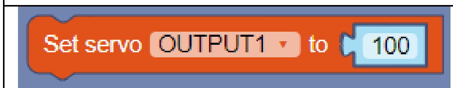
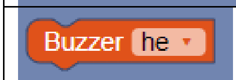
The easiest way to programme in LOFI Blocks is to install the LOFI Blocks app on **your iPhone or iPad** and connect to the robot using the robot **BLE Bluetooth module**.
https://itunes.apple.com/us/app/lofi-blocks/id1209534702?mt=8

You also need to upload the LOFI Blocks Arduino sketch into your robot, it is available here:
https://create.arduino.cc/editor/murcha/974b6eb5-ab51-4e97-9ceb-754eaa42670c/preview

There is also a LOFI Blocks app **for Android devices** and a LOFI Blocks extension for the Chrome browser, but they use the older version of Bluetooth. To use these you need to plug in **a Bluetooth 2.0 module** into the robot.
https://play.google.com/store/apps/details?id=com.lofiblocks.lofi.robot&hl=en
https://chrome.google.com/webstore/detail/lofi-robot-extension/opdjdfckgbogbagnkbkpjgficbampcel?utm_source=chrome-ntp-icon

The following functions are available in LOFI Blocks for the 3 wheel robot:

<table>
<tr><th>LOFI Block name</th><th>What it does</th></tr>
<tr><td>
</td><td>Controls motor 1 and motor 2 - can select power from 0 (PWM of 0) through 100 (PWM 255)</td></tr>
<tr><td>

Can selection OUTPUTS 1 through 4
Can select number 0 through 100, for LEDs:
0 = off, any other number = on</td><td>OUTPUT1 - turn RED LED on or off
OUTPUT2 - turn GREEN LED on or off
OUTPUT3 - turn left by the number of degrees specified
OUTPUT4 - turn right by the number of degrees specified</td></tr>
<tr><td>
</td><td>all four servo OUTPUTS turns the robot servo arm to the angle specified</td></tr>
<tr><td>
</td><td>turns RED LED on or off (the robot does not have a buzzer)</td></tr>
<tr><td>
</td><td>INPUT1 - battery capacity remaining (in percentage)
INPUT2- the angle in degrees the robot is turning to
INPUT3 - uptime in minutes
INPUT4 – measures distance – see Distance sensor</td></tr>
<tr><td>
</td><td>If you have plugged in the ultrasonic sensor this measures distance in cm (max is 125 cm)</td></tr>
</table>

Customise your RemoteXY app GUI

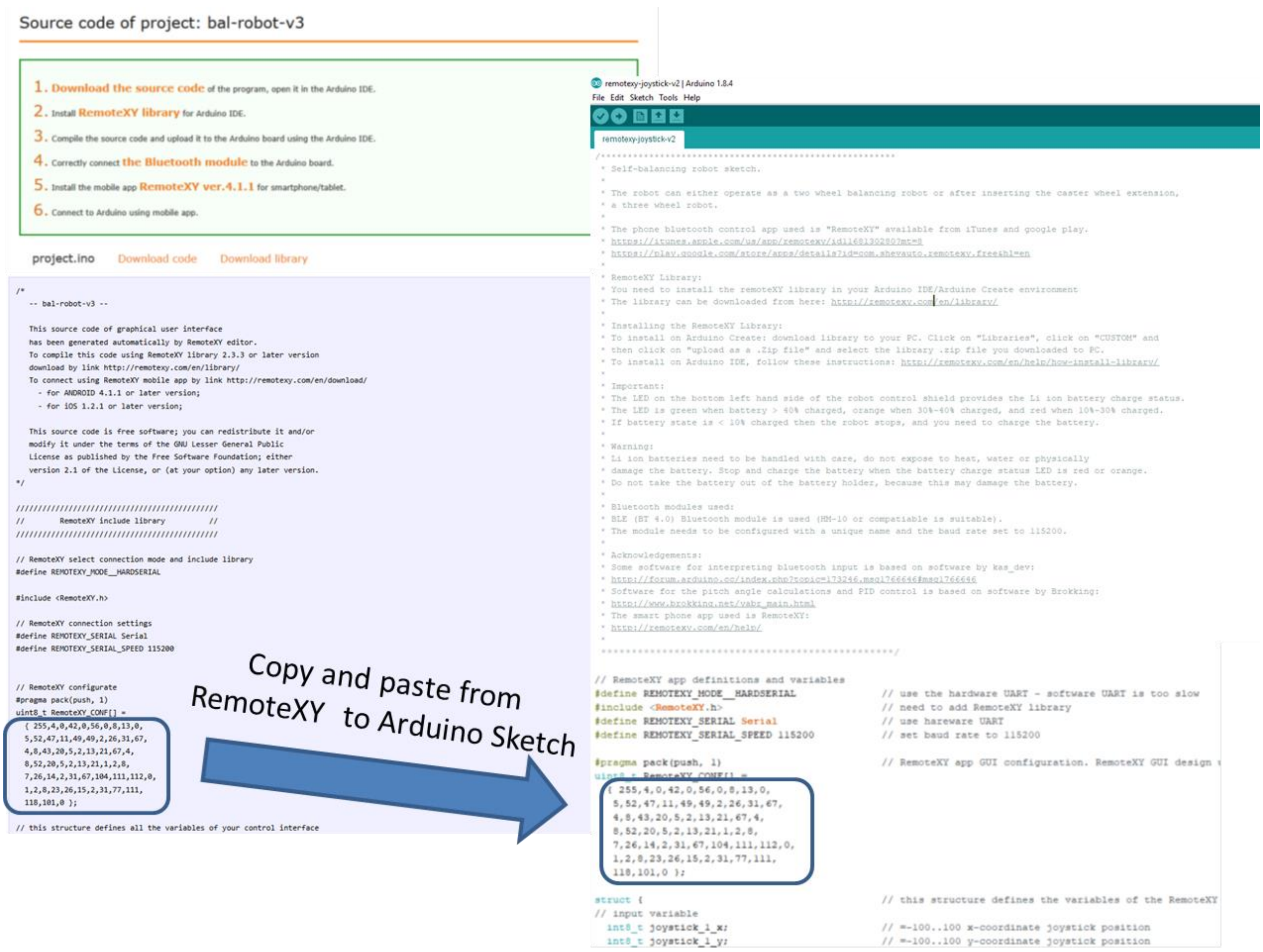
It is possible to change the RemoteXY app GUI colours, position, size and the shape of the controls.

You can edit the default GUI design for the balancing robot here. Note you will need to create a login to see the code.

https://remotexy.com/en/editor/c2f7710a6b494c7da66bdec7e2baa440/

As long as you do not add or delete GUI controls or change the name of the variables used, all you need to do is:

- make your GUI changes in the RemoteXY editor,
- click on “Get source code”
- copy the RemoteXY_CONF[] data structure and
- paste it into the Exercise 8: Full Balancing Robot sketch. See below:



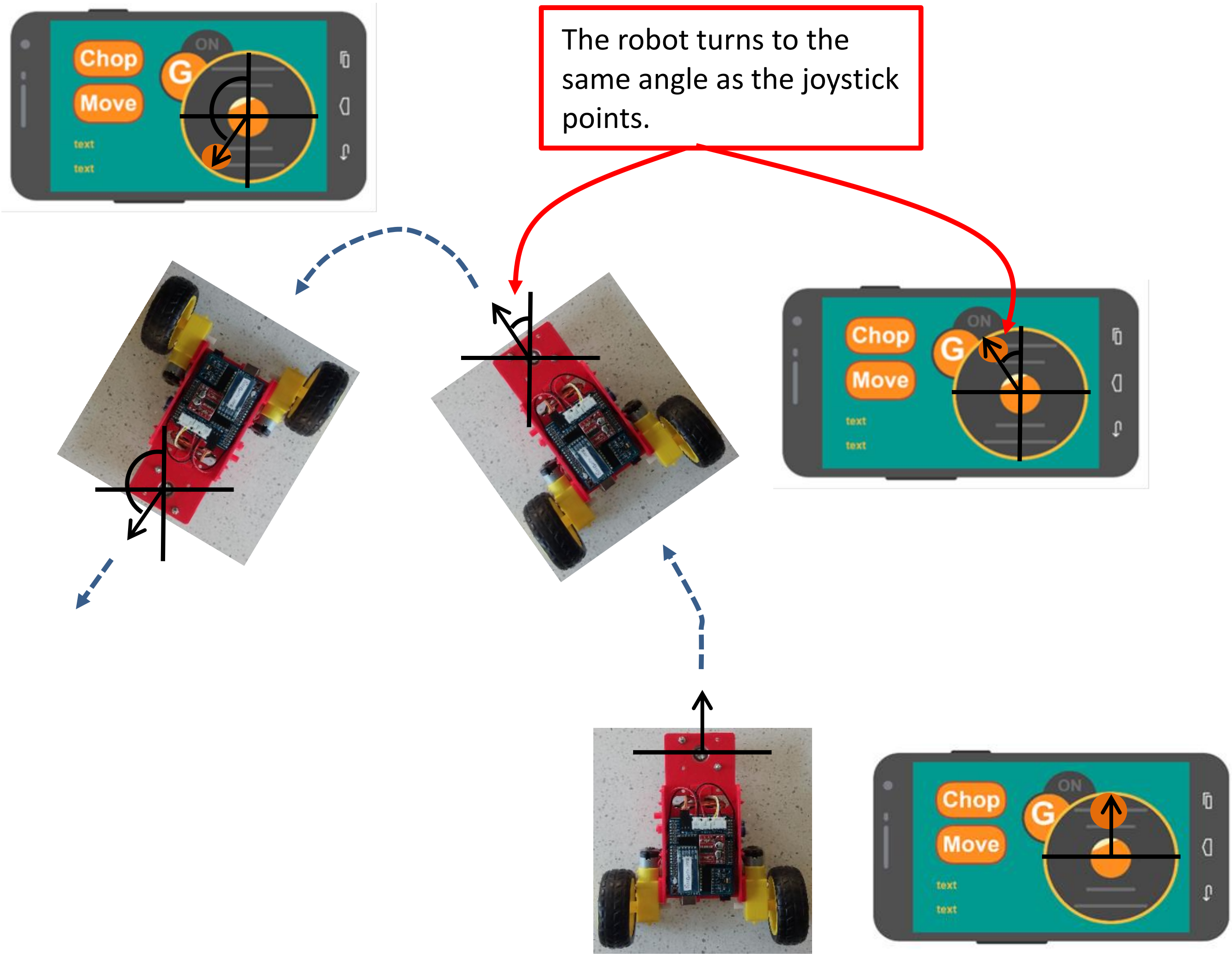
The Joystick

When a joystick is used to control a remote controlled vehicle there are a couple of ways the joystick can work.

(1) One way is when you push the joystick, for example left, the vehicle turns left and continues turning left until you bring the joystick back to the middle. The further your push the joystick left the faster the vehicle turns left. This is the way many toy RC cars work.

(2) Another way is when you push the joystick, the vehicle moves to the particular direction the joystick is pointing. For example, when you push the joystick left, the vehicle rotates to the same angle as the joystick. This is how this robot works. The robot can do this because the robot has a gyroscope.

The diagram below explains this more clearly - the robot moves to the same angle and direction as the joystick.



What do the GUI text fields mean ?

The RemoteXY GUI has two fields which display information about the robot in real time. When you switch the robot on the default fields are: “Battery” and either “Goto angle” (if operating as a 3 wheel robot) or “Pitch angle” (if operating as a balancing robot).

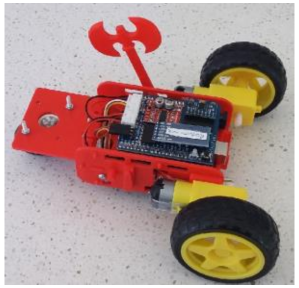
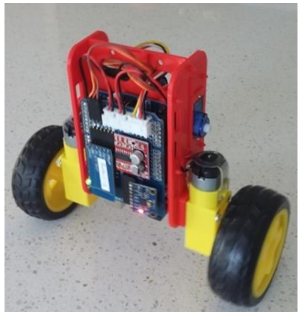
However there are four other GUI fields you can select ! These other fields provide interesting insight into the robot operation. They are selected by pressing the “Chop” or “Move” buttons for 2 seconds. See the table on the next page for a full explanation of all the fields – it looks a bit complicated but it’s not really !

Here are the fields when the robot is operating as a 3 wheel robot:

These are the default fields when you switch on the robot.

These are the fields when you press the “Chop” button for 2 seconds – it gives the length and angle of the joystick control.

These are the fields when you press the “Move” button for 2 seconds – if gives you the PWM values sent to the motors.

Robot operating as a Balancing Robot or as a 3 Wheel Robot	**How selected**	**Field name displayed**	**What does it mean !**
When operating as a 3 wheeled robot 	When you switch on the robot (default)	Bat (Battery)	Displays the % charge remaining in the battery.
		Ang (Goto angle)	Displays the angle (in degrees) that the robot it being commanded to rotate to by the joystick (note: displays distance in cm when the ultrasonic module is plugged in).
	Press "Chop" button for 2 seconds	len (joystick length)	Displays the length of the joystick vector. The position of the joystick is given as an x and y coordinate, and the joystick length is the length from the origin to the current position.
		ang (joystick angle)	Displays the angle of the joystick vector. The angle is between -180 to 180 degrees.
	Press "Move" button for 2 seconds	rm (right motor)	Displays the PWM (pulse width modulation) value sent to the right hand motor.
		lm (left motor)	Displays the PWM (pulse width modulation) value sent to the left hand motor.
When operating as a balancing robot 	When you switch on the robot (default)	Bat (Battery)	Displays the % charge remaining in the battery.
		Ang (Pitch angle)	Displays the angle (in degrees) of the robot from the vertical (i.e. how much the robot is leaning over).
	Press "Chop" button for 2 seconds	PID (PID output)	Displays the speed the robot is moving forward or backwards (numbers less than 5 is slow speed, numbers greater than 10 is fast speed).
		spt (setpoint)	Displays the amount the joystick is telling the robot to lean forward or backwards. By making the robot lean over is how the balancing robot moves forward or backwards.
	Press "Move" button for 2 seconds	time (uptime)	The duration in seconds that the robot has been switched on and running (known as uptime).
		dur (duration)	The program sketch in the balancing robot runs every 4 ms. The duration field displays the actual length of time this program sketch runs for – it needs to be less than 4ms.

User Configuration Settings

There are several robot configuration settings you can change if they wish to. They are at the top of the sketch just under the opening comment section, see below:

```
 ************************************************/
// user configuation settings
#define three_wheel_max_speed 150    // in 3 wheel mode - this sets the 
#define joystick_sensitivity 2       // this sets the number of 360 degre
#define chop_time 10                 // sets the duration of chop arm act
#define retain_turn_angle 0          // in 3 wheel mode - this sets wheth
```

The default values work well, however you might wish to increase the maximum speed of the robot when in 3 wheel mode. Note: the default value for three_wheel_max_speed is set less than the max possible to reduce the risk of the robot crashing into objects and damaging them, change this value at your own risk !

The user configuration settings are:

Setting name	**Default value**	**Description**
three_wheel_max_speed	150	In 3 wheel mode, this sets the max speed of the robot. The value can be set to any value between 100 and 255. A value of 100 is slowest and a value of 255 is fastest. Note, this value is the PWM value for max speed.
joystick_sensitivity	2	This sets the number of 360 degree turns of the joystick needed to make the robot do one 360 degree turn. It sets how sensitive the movement of the joystick is on the robot. A value of 2 gives a natural feel to the joystick, and means two turns of the joystick is required to make the robot do one 360 degree turn.
chop_time	10	This sets the duration of the chop arm action. The duration of the chop action = chop_time x 0.1 seconds (e.g. for chop_time = 10, the duration of the chop arm action is 10 x 0.1 = 1 second.)
retain_turn_angle	0	In 3 wheel mode, this sets whether the robot remembers the goto turn angle when not moving. A value of 0 means is does not remember, 1 means it does remember. When set to 0, if the robot is picked up and turned, the robot will not return to its original direction, whereas if set to 1 the robot will return to its original direction.
IMU_orientation	0	When the MPU-6050 is plugged into the shield then set IMU_orientation = 0, when MPU-6050 is positioned in the slot underneath the robot then set IMU_orientation = 1

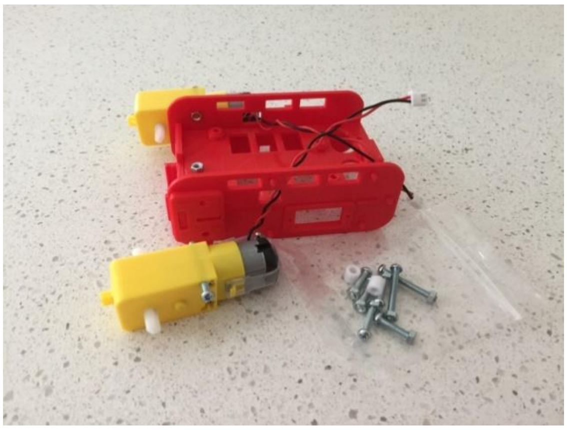
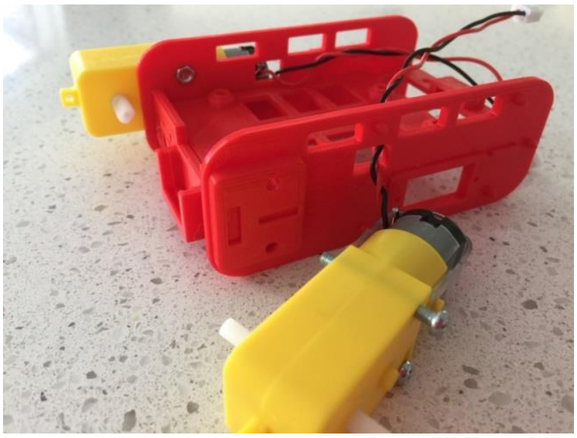
Trouble Shooting

The table below gives some trouble shooting guidance.

If these suggestions do not help then try loading the Exercises 1 through 6 one at a time, and use a process of elimination to try and identify which module or aspect of the robot is not working.

Trouble	**Possible cause**
Robot does nothing.	Check that the power switch is on. Check that the battery is charged. Check that all the plugs and modules are connected correctly (double check with the Makes Steps). Check that the Robot control shield pins are not misaligned (double check Make Step 3). Try a process of elimination by loading each Exercise 1 through 6 one at a time to identify the cause.
Cannot upload sketch to robot.	Make sure there is NOT a Bluetooth connection between phone and robot. A Bluetooth connection prevents uploading occurring.
Unable to make Bluetooth connection with phone.	If a Bluetooth connection has been recently made to the robot, it can take some time for the Bluetooth module to disconnect. Switch the power off and on to the robot. When the robot is switched on you need to wait until the calibration period is complete (the LED stops blinking), a Bluetooth connection cannot be made when the robot is calibrating. Wait until the LED turns green then make a Bluetooth connection to the robot.
Robot power switches off by itself.	The robot battery switches off automatically if the battery is flat. Switch off the robot power and charge the battery.
My phone is asking for a Bluetooth pin.	Although generally not needed, the pin for the Bluetooth module is 123456.
Compile errors when uploading Exercise 8 Full Balancing Robot	Make sure you have installed the RemoteXY library in your Arduino IDE. See Make Step 12.
Battery not charging	Make sure you have switched off the power switch to the robot.
Serial Monitor not working	Make sure the baud rate is set to 115200.
The control shield LED is blinking and robot does not work.	Make sure the robot is sitting on the ground and not moving, it should take about 5 seconds to calibrate.
The control shield LED is RED and the robot does not work.	The battery is flat and needs to be recharged.

Make Steps

<table>
<tr>
<td>



</td>
<td>

Step (1)

Mount the two drive motors on the sides of the chassis.

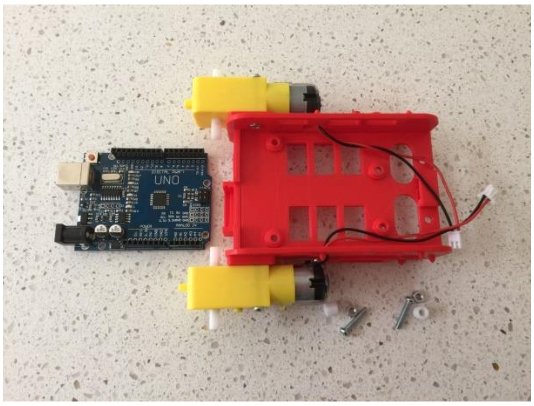
Insert the motor wiring plug through the square hole as shown.

Use the 25mm (the long bolts) bolts and nuts from the plastic packet.

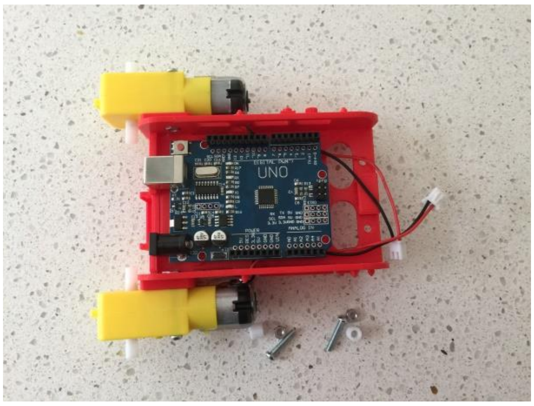
Tighten so the bolts are firm, BUT not too tight, with the provided screw driver.

</td>
</tr>
</table>

Step (2)



Mount the Arduino Uno board.

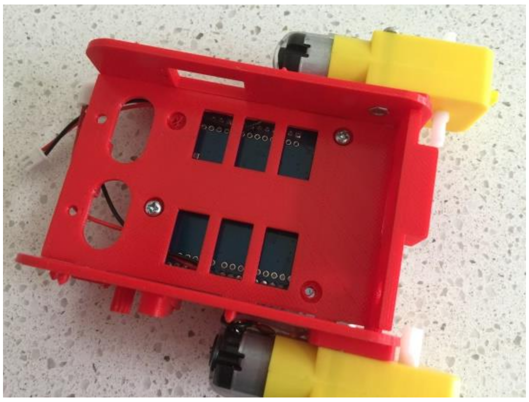


Place the Arduino Uno board on the chassis - on top of the raised cylinder shaped stand-offs.

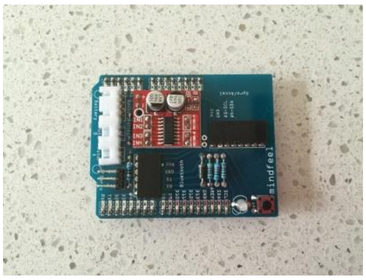
Insert two 15mm (the shorter bolts) bolts from underneath the chassis and through the Arduino board.

Note, we will only use two bolts to mount the Arduino Uno as shown in the photos.

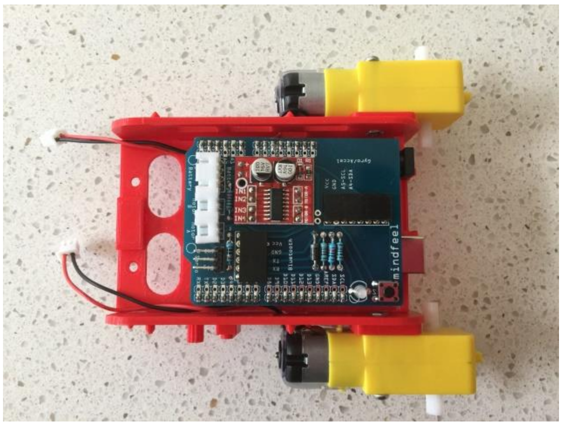
Place the white washers over the bolts on the Arduino side and screw on the nuts (NOT too tight) - the white washes protect the Arduino board from the nuts.



Step (3)

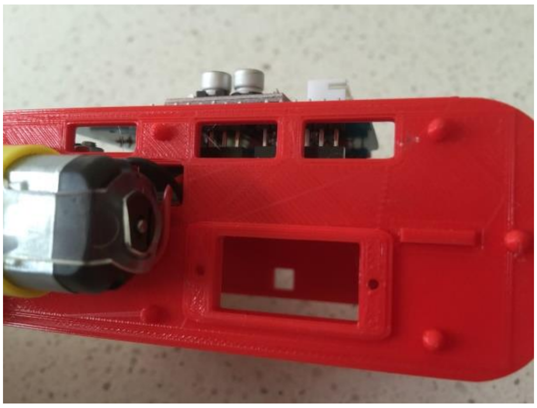


Mount the Robot Control Board (shield) on top of the Arduino Uno - after this step you can do coding **Exercise 1.**

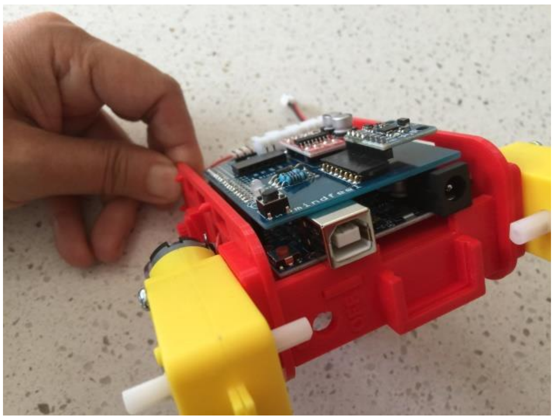


Carefully plug in the Robot Control shield into the Arduino Uno.

NOTE: it is a common mistake to misalign the pins of the Robot Control shield and the Arduino Uno connector pins – take special care to make sure they do align.

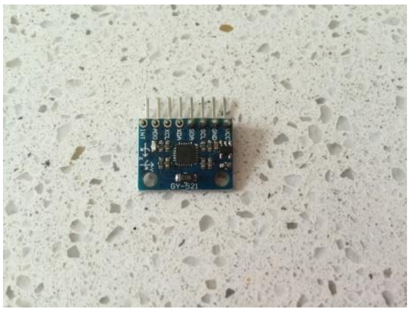


Make sure the pins are aligned correctly by looking through the square openings in the sides and that it is pushed in fully.

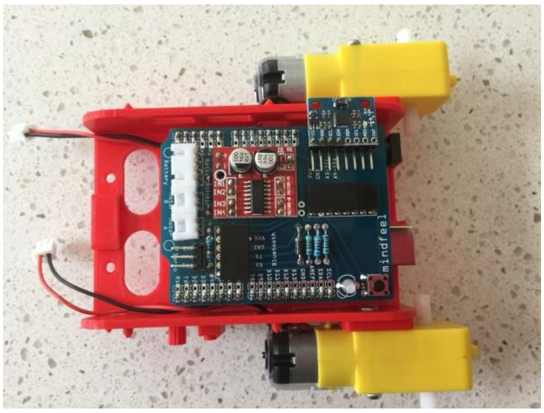


The Robot Control shield when fully inserted should be touching the top of the USB connector on the Arduino Uno board.

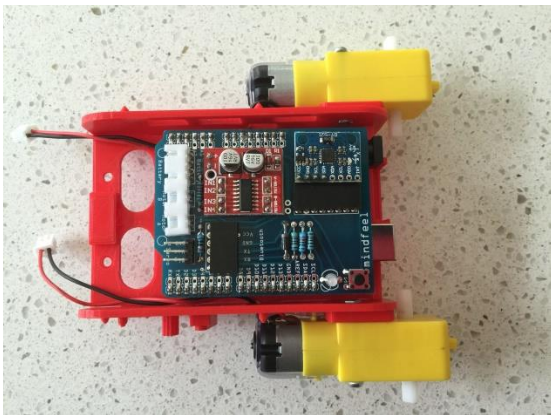
Step (4)

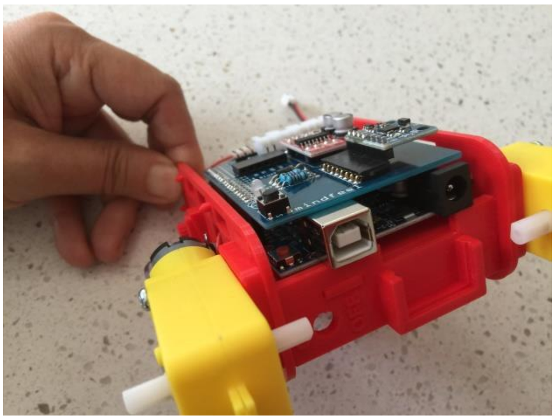


Insert the MPU 6050 sensor module into the control board (shield) – after this step you can do coding **Exercise 2**.



Slide the MPU 6050 sensor module into the 8 pin plug in the Robot Control shield, as shown.



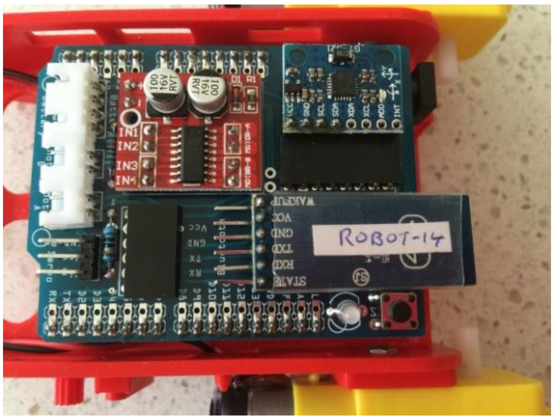


Step (5)

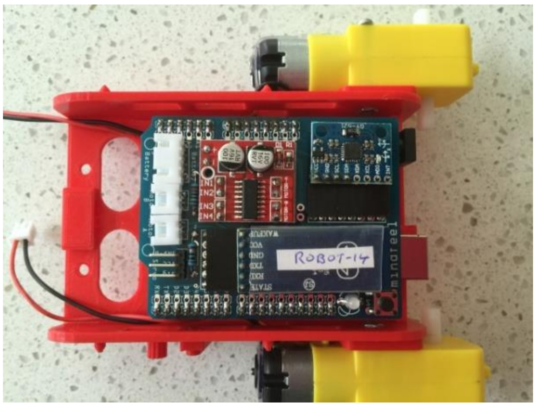
Insert the Bluetooth module into the Robot Control shield – after this step you can do coding **Exercise 3** and **Exercise 4** (note exercise 4 is optional, it requires and ultrasonic sensor which may not be supplied in your kit – see step 14)

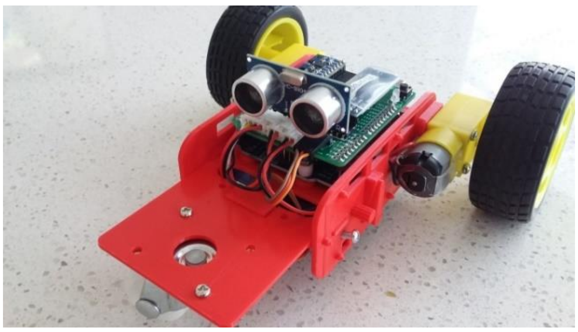
Note that the Bluetooth (BT) module has a sticker on one side giving the BT name of your robot – every robot has a different BT name.

Although generally not needed, the pin for the Bluetooth module is 123456.

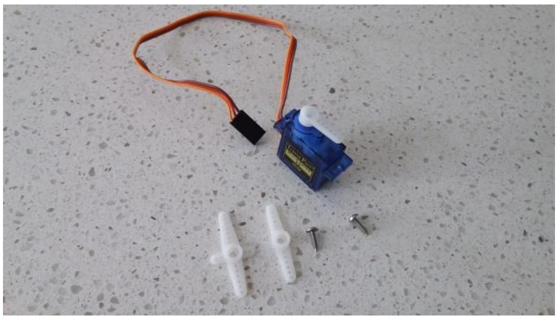


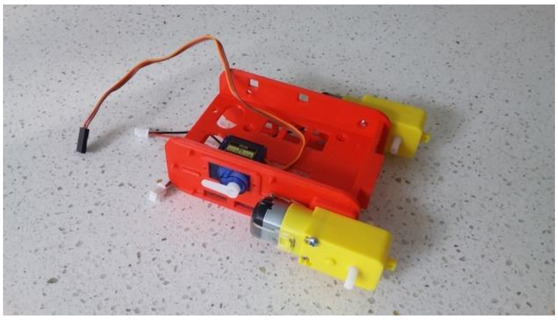
Slide the Bluetooth module (with the bluetooth name upwards so it is visible) into the 6 pin plug in the Robot Control shield, as shown.

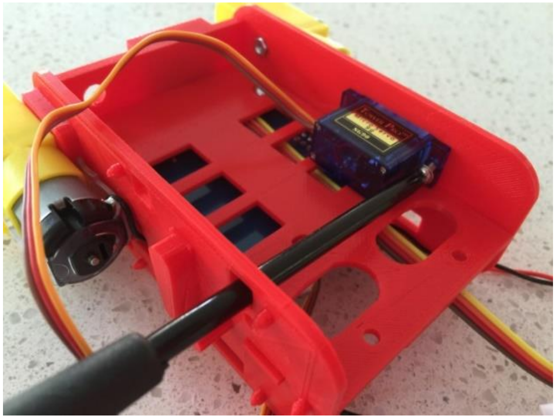


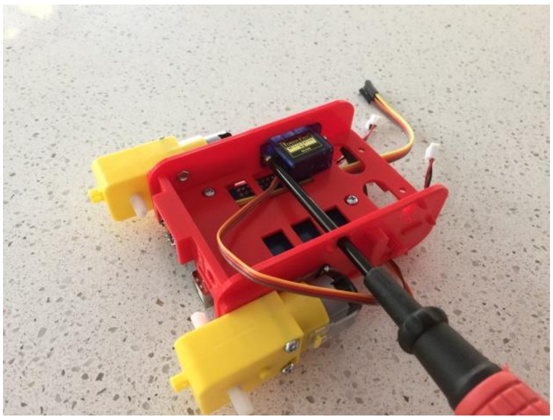


If you have an ultrasonic sensor you can plug it into the shield as shown in the photo and do Exercise 4. The ultrasonic sensor is optional and is not required for the balancing robot.









Step (6)

Mount the servo mechanism.

Unpack the servo from its bag.

It contains two screws and three white levers - one of levers is already mounted on the servo drive shaft – keep the other two levers for possible future use.

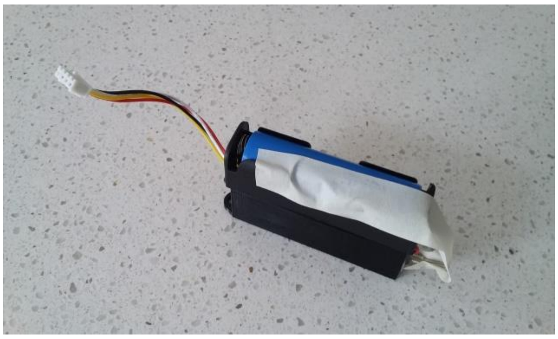
The servo uses the two screws to mount the servo onto the chassis.

Push the servo into the rectangular slot in the side of the chassis.

The lever mounted on the servo should be pointing towards the top of the robot (see photo).

The heads of the screws can be reached by positioning the screwdriver through the square holes on the opposite side of the chassis as shown.

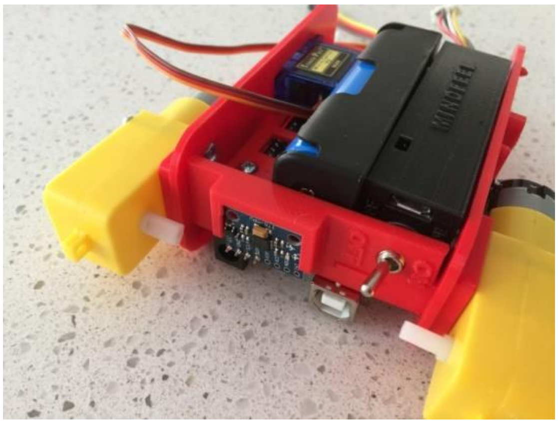
Step (7)



Mount the battery pack.

The battery pack comes with masking tape holding the power switch in the "off" position.

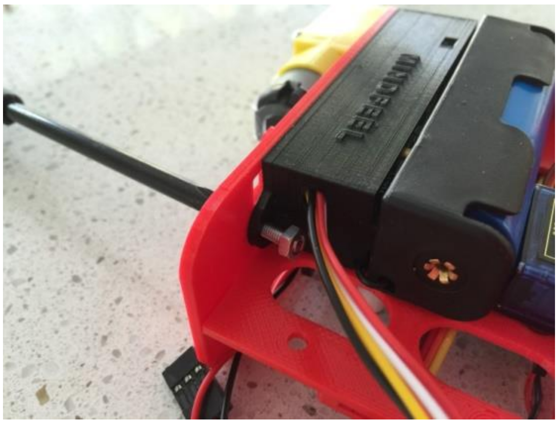
Remove the masking tape and make sure the switch is in the OFF position – "off" and "on" are labelled on the battery pack.



Mount the battery pack beside the servo, as shown.

Push the switch through the hole in the bottom of the robot.

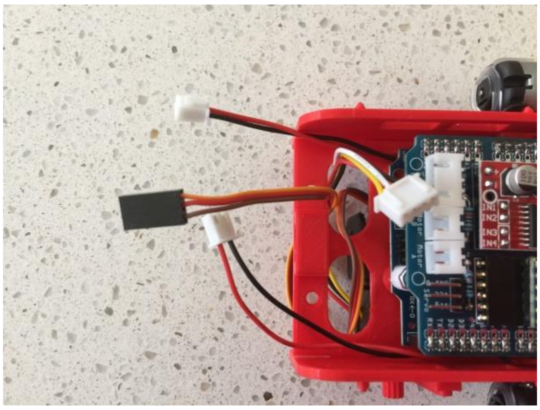
Use a 15mm (the shorter bolt) bolt and nut through the side of the chassis to secure the battery pack.



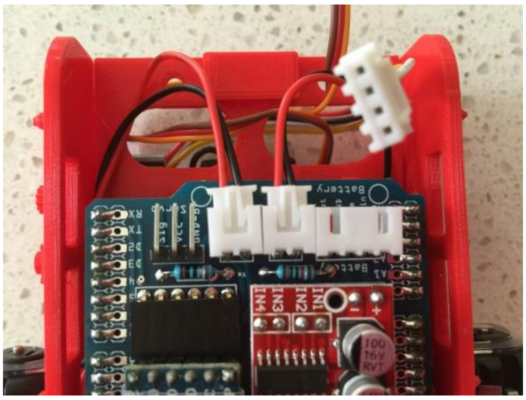
This robot uses a Lithium Ion battery. It is important to look after these types of batteries by:

- Charge the battery before the LED on the robot shield turns RED.
- Do not remove the battery from its holder because this may damage the battery.
- Do not physically damage the battery.
- Do not expose to water (e.g. do not use the robot outside when wet)
- Do not expose the battery to heat. If the battery is hot, turn off the robot and take it outside.
- Do not use the Li Ion battery if it has not been charged for more than 6 months.

Step (8)

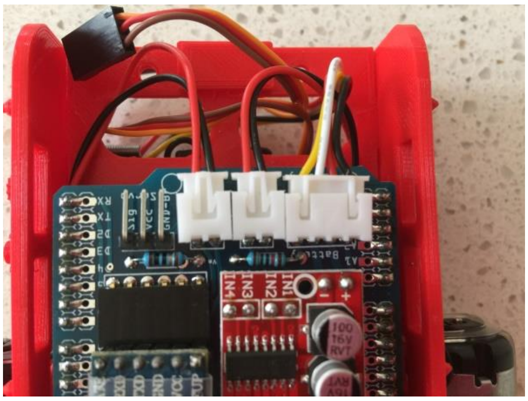


Plug in the cables into the Robot Control shield – after this step you can do coding **Exercises 5, 6, and 7.** Note: you need to switch the battery power pack switch ON for these exercises to work.

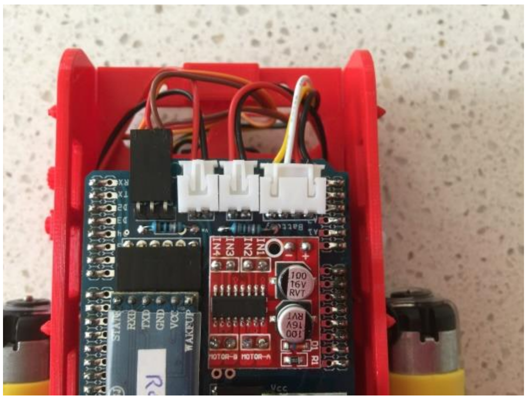


Plug in the motor cables.

The RHS motor plugs into the RHS socket, and the LHS motor plugs into the LHS socket.



Plug in the four wire battery cable – make sure the power switch is OFF.



Plug the servo cable plug into the servo socket.

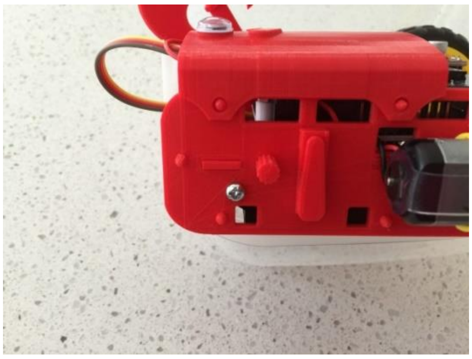
Make sure the brown wire aligns to the label on the robot shield – the brown wire of the three wire servo cable should plug into the pin labelled “GND-Br”.

Because the servo cable is quite long it can be wrapped around the centre member between the two oval openings at the top of the chassis – see photos.

Step (9)

Mount the two arms.

One of the arms has a slot which pushes onto the servo lever. Push this arm onto the servo lever which is already attached to the servo motor.



The other arm can either have a round toothed mount (like a cog) or have the same servo style mount – this is the fixed arm and mounts on the opposite side of the robot to the servo.

On the opposite side of the robot are fittings that you can attach either type of fixed arm:

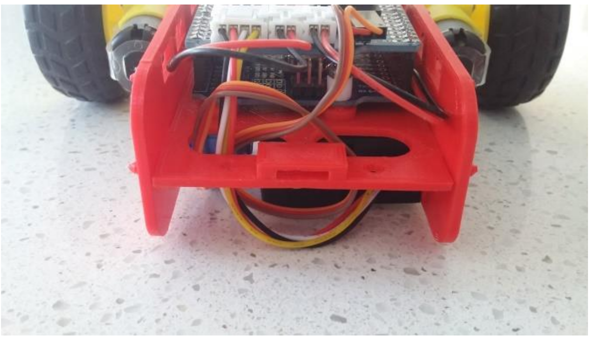
- there is a mounting for arms with servo style mounts and
- there is a mounting for arms with spline (round toothed) mounts.

You may have either a toothed mount or servo mount fixed arm. Attach the arm you have to the appropriate mounting on the robot.

Step (10)

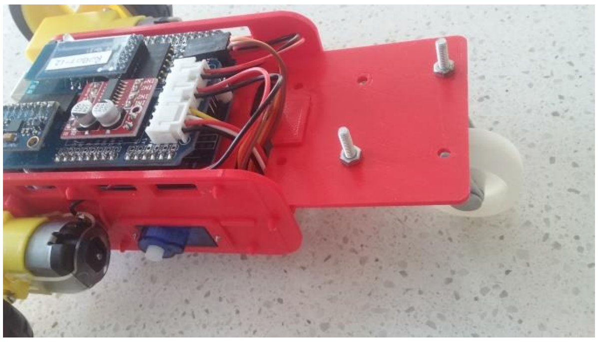
Fitting accessories.

There is a slot in the top of the robot and pins on the side of the robot for clipping on accessories.



The third wheel accessory (to turn the robot into a three wheel robot) clips into the slot at the top of the robot – as shown.

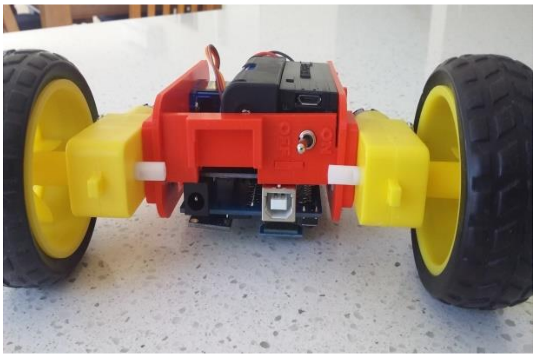
Lay the robot on all three wheels and the robot automatically becomes a three wheel car.

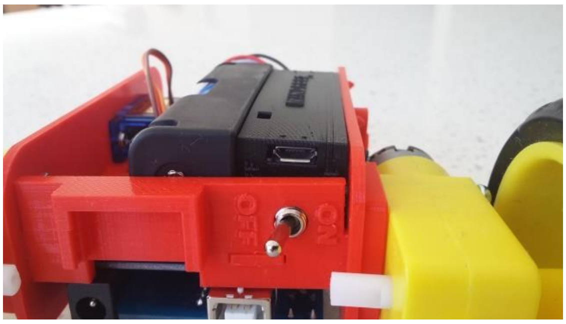


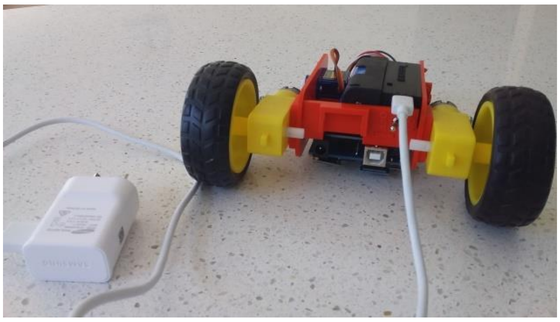
This same slot can also be used to slot on robot heads, these are not provided in the kit but you could 3D print these for yourself.

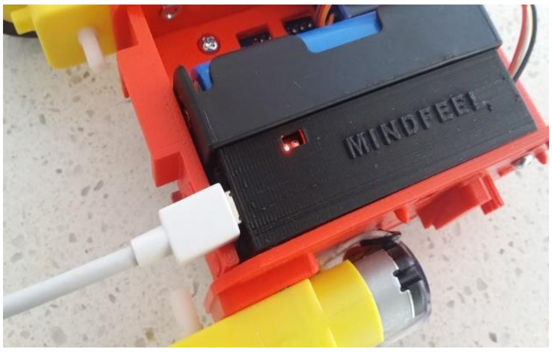
The face clips onto the pins on the side – as shown.

Step (11)









Charging the battery.

There is a micro-USB connector on the battery pack for charging.).

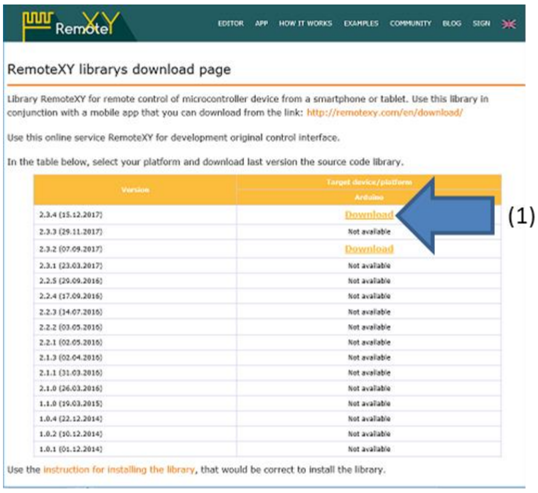
Use a micro-USB charger (1 Amp or larger), such as a phone charger or a PC.

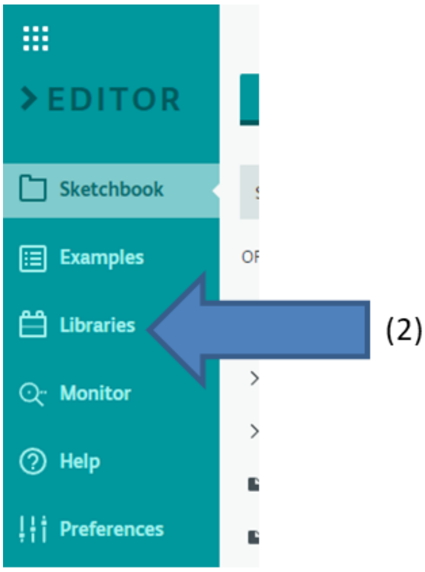
Before charging the battery make sure you switch off the robot.

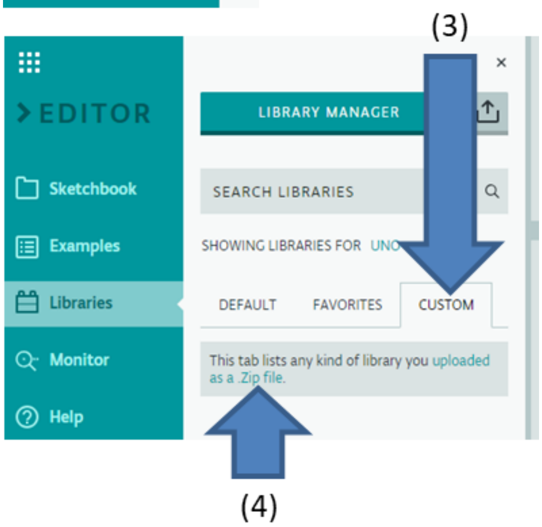
Note, a charger is not included in the robot kit. If you need a charger they can be bought inexpensively from the likes of the Warehouse.

The red LED on the battery pack indicates the battery is charging.

The Led turns green or blue when the battery is fully charged.







Step (12)

Upload **Exercise 8: Full Balancing Robot** Sketch to your robot.

Add the **Exercise8: Full Balancing Robot** sketch to your Arduino Create sketchbook: https://create.arduino.cc/editor/murcha/27013033-c0e9-436c-ac99-874e0dcb77fa/preview

Install the **RemoteXY library** on your Arduino Create IDE: first download the **RemoteXY library** to your PC from http://remotexy.com/en/library/, then click on "Libraries" in Arduino Create IDE, then click on "CUSTOM" and then click on "upload as a .Zip file" and select the library .zip file you downloaded to your PC.

Upload **Exercise 8 Full balancing robot** to your robot.

Install the **RemoteXY** app on your Android or iPhone.

Open **RemoteXY** on your phone, click on the “+” symbol in top right hand corner, select “Bluetooth BLE” and then select the name of your robot (written on the Bluetooth module).

The app should connect to your robot and the joystick GUI should open – you can now control your robot.

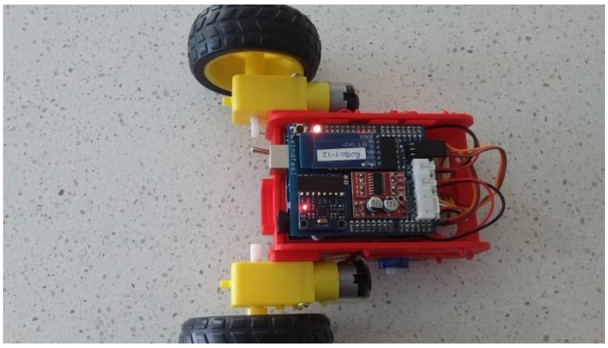
Notes:

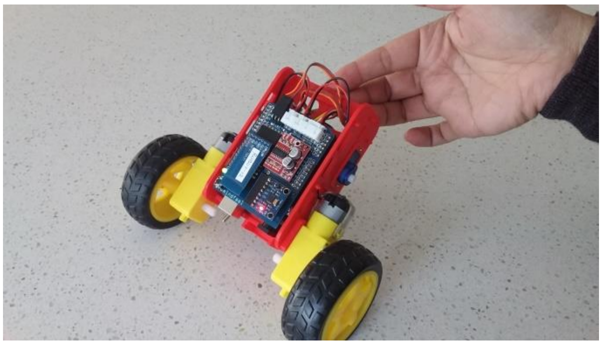
(1) To upload sketches to the robot make sure the phone is not connected to the robot by Bluetooth - a Bluetooth connection prevents uploading occurring.

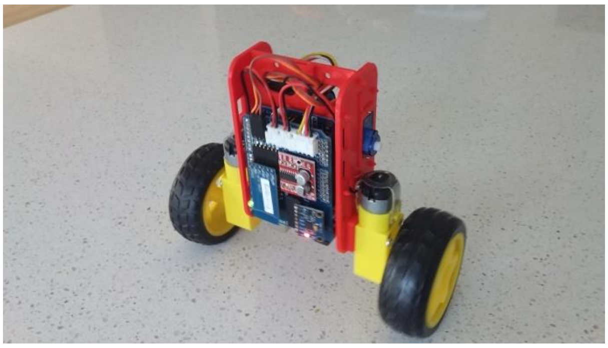
(2) Make sure you wait until the calibration period is complete (the LED stops blinking), a Bluetooth connection cannot be made when the robot is calibrating. Wait until the LED turns green then make a Bluetooth connection to the robot.

(3) Although generally not needed, the pin for the Bluetooth module is 123456.

Step (13)







Operating instructions.

Upload the code in **Exercise 8 Full balancing robot.** Note, make sure you have downloaded the RemoteXY library – see page 6.

Lay the robot on its battery pack on a still surface. Switch the robot on.

If all is working correctly the LED on the robot shield will start blinking orange for about 5 seconds and then turn green.

During this time the robot is calibrating its gyros, it is important to be on a still surface otherwise the gyros will not calibrate and the LED will not turn green.

Once the LED stops blinking it turns green.

Note, the small black push button beside the LED resets the robot - this is similar to switching the robot “off” and “on”.

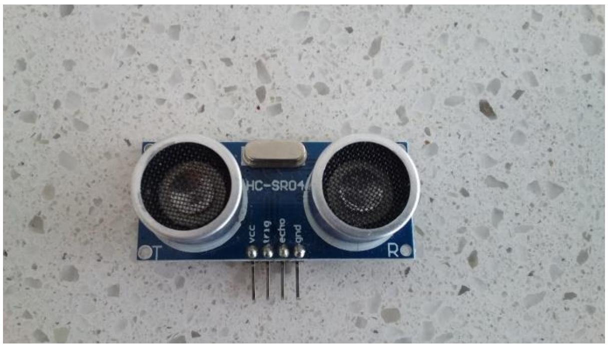
You can now make a Bluetooth connection from your phone app. Although generally not needed, the pin for the Bluetooth module is 123456.

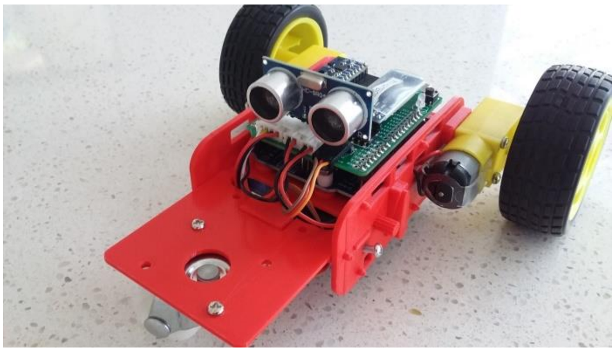
Lift the robot to its vertical position and the robot will start to self-balance.

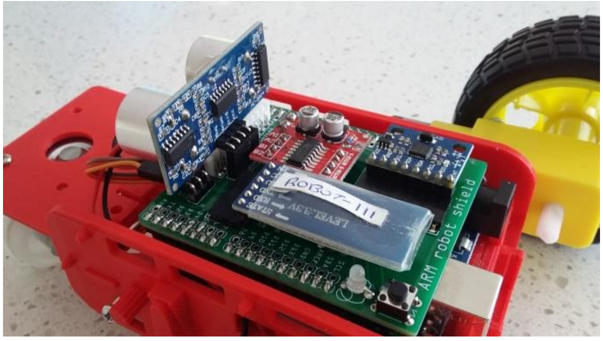
When the battery is about 70% consumed (30% remaining) the LED will turn orange/red. This is an indication to charge the robot.

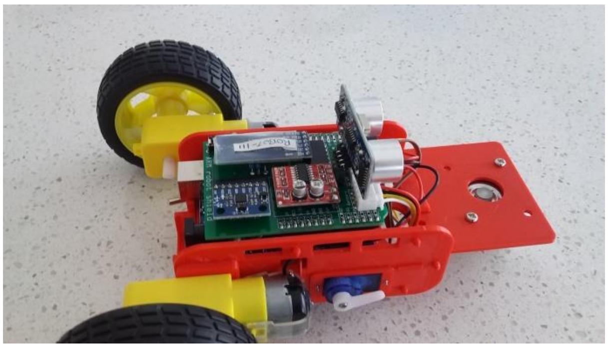
When the battery is 90% consumed (10% remaining) the LED will turn red and the robot will stop working.

This indicates the battery is now flat and needs recharging. In this state the robot will not work.









Step (14) - optional

Plugging in the Ultrasonic sensor (optional)

Note, the ultrasonic sensor only works in 3-wheel mode.

The ultrasonic sensor is optional, the robot works with or without the sensor plugged in.

When the ultrasonic sensor is plugged in the robot detects obstacles and automatically tries to avoid them.

Note, when the ultrasonic sensor is plugged in the “Angle” text message changes to “Dist” and displays the distance (in cm) to objects instead of the robot turning angle.

The plug on the shield for the ultrasonic sensor was added after the first design of the robot, which is why the previous photos in this document does not show the ultrasonic sensor plug.

IMPORTANT: Make sure the ultrasonic sensor is plugged in the correct way round (the transducers facing the caster wheel). If you plug the sensor in the reverse direction it causes a short circuit on the 5V power rail which results in the power supply heating up and will burn out the power supply eventually.

Appendix - To know more

This appendix provides some more information about the robot if you are interested.

Firstly, there are a lot of sites about learning Arduino.

I have found the links below to provide useful well-structured info.

https://www.tutorialspoint.com/arduino/index.htm - well-structured Arduino tutorial

https://www.arduino.cc/en/Reference/HomePage - official Arduino coding language reference

https://www.youtube.com/channel/UCfYfK0tzHZTpNFrc_NDKfTA - good youtube tutorials

https://www.youtube.com/user/educ8s - good youtube project ideas

3D Printing the Robot Pieces

The balancing robot chassis and accessories have been designed in TinkerCAD https://www.tinkercad.com/

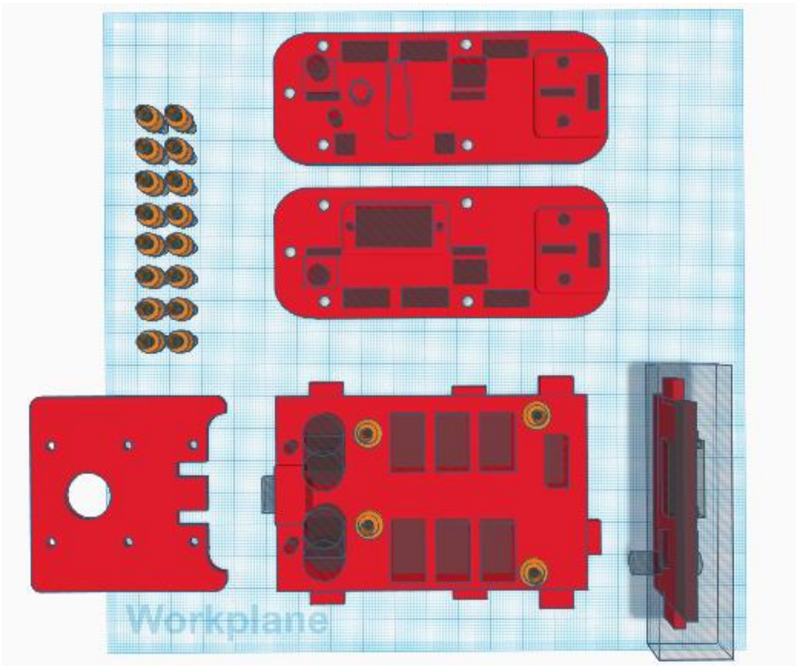
The chassis and accessories are public and can be obtained by searching in "3D Designs" for "balrobot" in the tinkerCAD gallery – use 1.75mm PLA filament. Note, you need an account and log into tinkerCAD to search for the robot designs.

The following pictures show what is available in the tinkerCAD gallery.

The chassis consists of four main parts: the centre, two sides and the bottom. https://www.tinkercad.com/things/7kurIVi1mjH-robot-chassis

There are tabs on the centre piece which push into slots on the sides and bottom pieces. The slots can be quite tight and sometimes need to be widened, for example by pushing a flat screw driver blade into the slot to ease it slightly wider.

Also included are the cylinder shaped standoffs and the 3rd wheel extension piece.

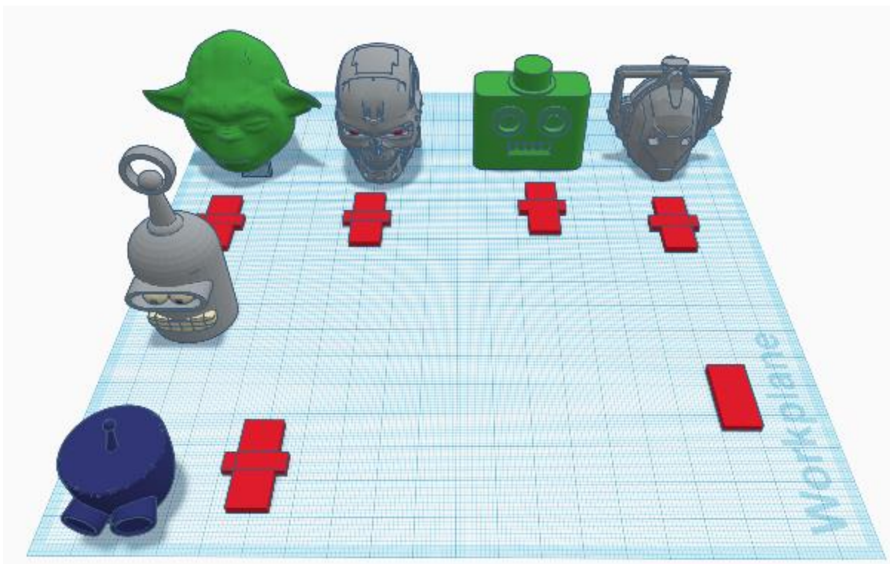


A number of heads are included.

https://www.tinkercad.com/things/jeIGw2IF9BG-robot-heads

These heads can slot into the top of the robot.

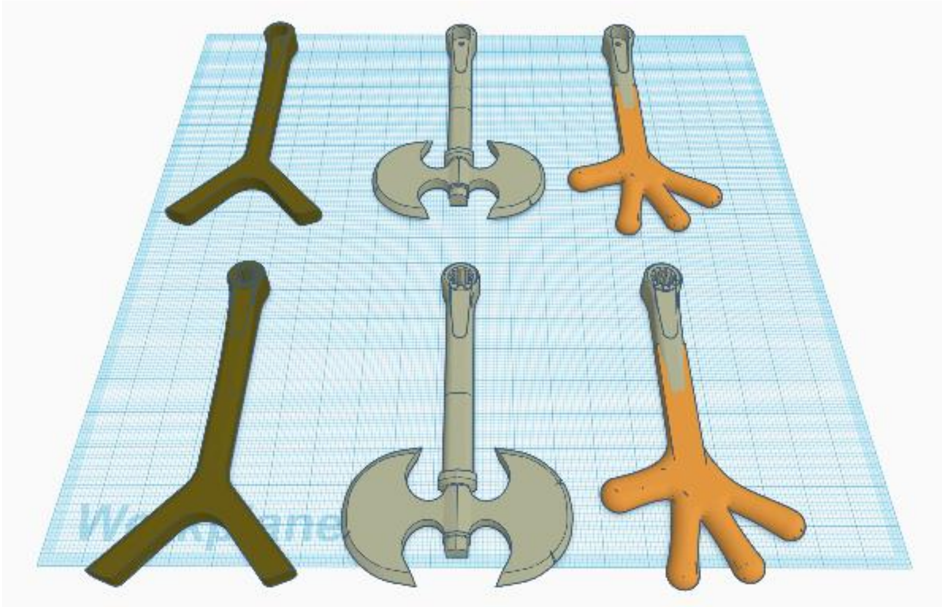
To mount the head, the cross shaped pieces slide into the bottom of the head and into the slot at the top of the robot.



A number of arms are included.

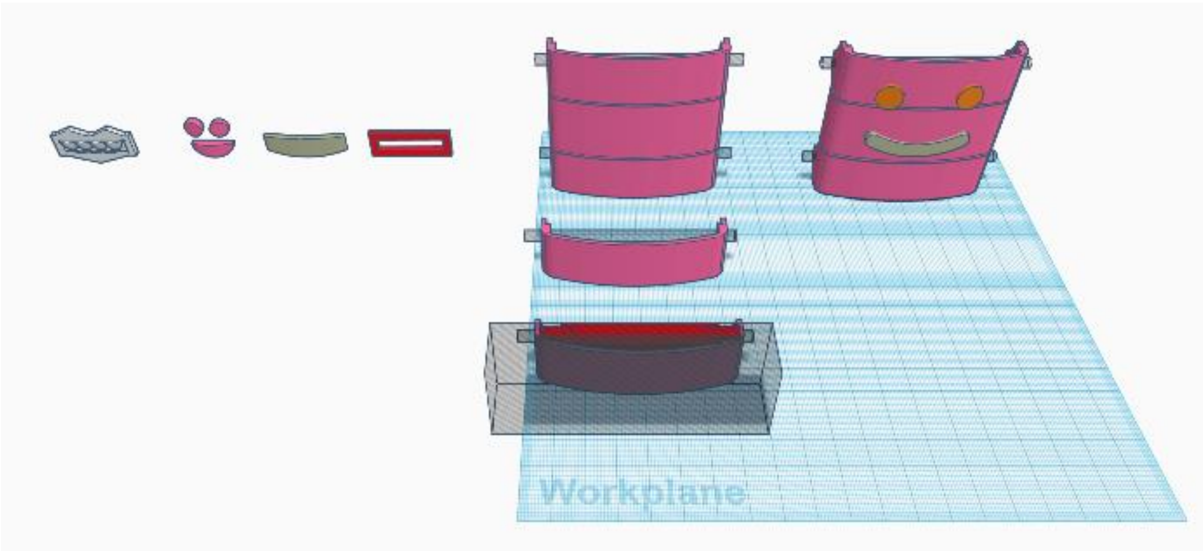
https://www.tinkercad.com/things/0i2t49V2aLT-robot-arms

Note there are two types of arm fittings: the servo style fitting and the spline (toothed) type fittings.



A number of covers and faces are included.

https://www.tinkercad.com/things/eSeVbkUcfF0-balancing-robot-covers



There are two plain covers (small and large) and a number of faces profiles.

Other 3D parts are:

https://www.tinkercad.com/things/cN6ua5f3QU7-battery-pack

https://www.tinkercad.com/things/kJSFXLdhV7v-battery-pack-narrow-tab

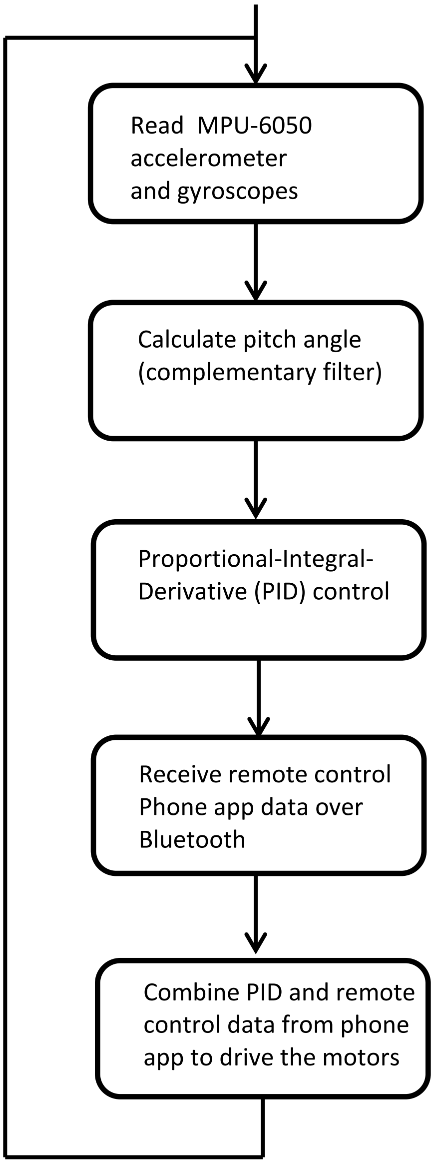
https://www.tinkercad.com/things/0Yxd4PHsSIg-raspberry-pi-zero-w-camera-mount

https://www.tinkercad.com/things/1ErXgednt3t-n20-motor-mount

https://www.tinkercad.com/things/4Gxd4K6ipSd-6xaa-battery-holder-clip

Balancing Robot Program Blocks

The main program blocks for the balancing robot are shown here. If you look carefully at your Arduino sketch you might be able to make out these blocks in the loop() function.



This code runs every 4ms (250 times per second).

This blocks reads the three accelerometers and three gyroscopes (along the x, y, and z axises) from the MPU-6050 sensor.

This block calculates the pitch angle (how much the robot is leaning) of the robot. It combines angle calculations from the accelerometers and the gyroscopes to obtain a stable pitch angle, through a process called a complementary filter. Trigometry and integration is used to calculate the angle – see the maths section.

Using the angle calculated above, this block determines how the motors should be controlled to keep the robot upright. It uses an algorithm called Proportional-Integral-Derivative (PID) control. It is a common algorithm used in many control situations.

This block receives the remote control data from the phone app joystick (and buttons). The x and y coordinates of the joystick position are converted into a length (using Pythagoras) and an angle (using the trig function Tan) which are then used to determine the speed and direction of the robot.

This block combines the PID motor control (which keeps the robot upright) with the controls received from the phone app and drives the motors so that the robot balances and moves according to the phone app joystick. The motors are controlled using Pulse Width Modulation (PWM).

Other phone apps

For your information, there are other free Bluetooth phone apps available. Here are three I have used in the past but note they may no longer be supported.

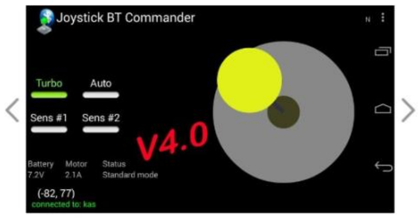
Joystick Bluetooth Commander

This app used the older Bluetooth 2.0 standard and was only available for Android phones.

https://forum.arduino.cc/t/android-bluetooth-joystick/169246

I did develop a version of the robot to use this app, if you are interested here are the details.

Suitable Arduino Bluetooth modules are the HC05 or HC06 Bluetooth modules. You will need to change the configuration on the HC05 or HC06 module to have a unique name and set the baud rate to 115200.



To configure the app:

- Go to the options page (tap the "cog" symbol in the right hand corner of the app and select "options") and set up the following:
- Go to "Button Properties" and select two buttons to display, and label the buttons: "Chop" and "Move" (these are the arm actions)
- Go back to options and go to "Data Field properties". Label the three data fields: "cal", "bat" and "angle"   Note, the "cal" field indicates when the robot is calibrated. The "bat" field indicates when the battery needs recharging. The "angle" field gives the angle from vertical of the robot.

You are now done with configuring the app.

You also need to upload a different Arduino balancing robot sketch into the robot, the sketch is available from: https://create.arduino.cc/editor/murcha/0f13a8f4-2178-4a88-9ab1-724f0c343b1b/preview

To make a BT connection to the robot, go to the connect page on the app (tap the cog and select "connect") and click on the BT name of your robot. Note, you will need to pair with the Bluetooth card on the robot first (go to Bluetooth setup on your phone). The first time you pair with Bluetooth you will be asked for a PIN, the PIN is "1234" – after that you won't be asked for a PIN again.

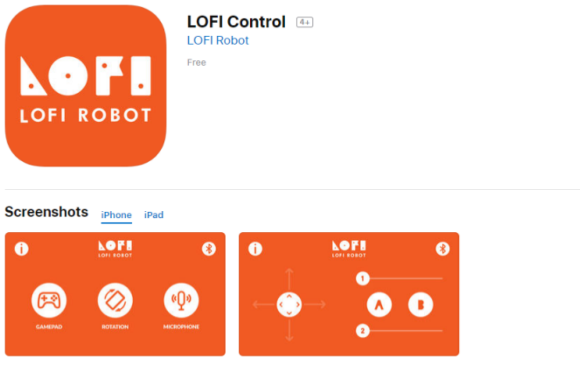
LOFI Robot and BLE Joystick phone apps are also apps which work quite well and are worth mentioning.

They are easy to use and do not need a library as is the case for the RemoteXY app.

LOFI Robot app

The “LOFI Robot” app is available from iTunes (for iPhones) and Google Play (for Android phones).

https://play.google.com/store/apps/details?id=com.LOFIRobot.Control&hl=en_US
https://itunes.apple.com/us/app/lofi-control/id1116978084?mt=8

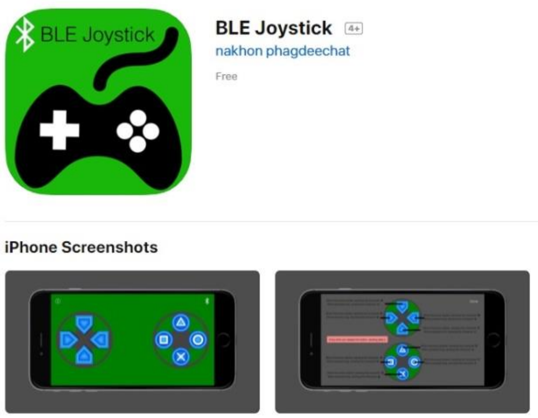


BLE Joystick app

The “BLE Joystick” app operates in a simple way, sending a single character when a button is pressed.

The “BLE Joystick” app, available from iTunes (for iPhones) and Google Play (for Android phones).

Note: there seems to be a bug with the “ BLE Joystick” app where it may not work for iPhones (depends on the manufacturer of the BLE module used in the robot).



https://itunes.apple.com/us/app/ble-joystick/id1098200556?mt=8

https://play.google.com/store/apps/details?id=iyok.com.blejoystick&hl=en_US

Robot Shield Circuit

The circuit for the robot control shield and its connections to the Arduino Uno are shown on the following page.

The key things are:

- the Bluetooth module (AT09 or compatible) RX and TX pins are connected to the hardware Arduino Uno UART pins D0 (TX) and D1 (RX). The Bluetooth module is configured with a unique name and the baud rate is set to 115200. Although generally not needed, the PIN for the Bluetooth module is 123456.

- the ultrasonic module Trig pin is connected to D8, and the Echo pin is connected to D2 (an interrupt is attached to D2 in the robot sketch)

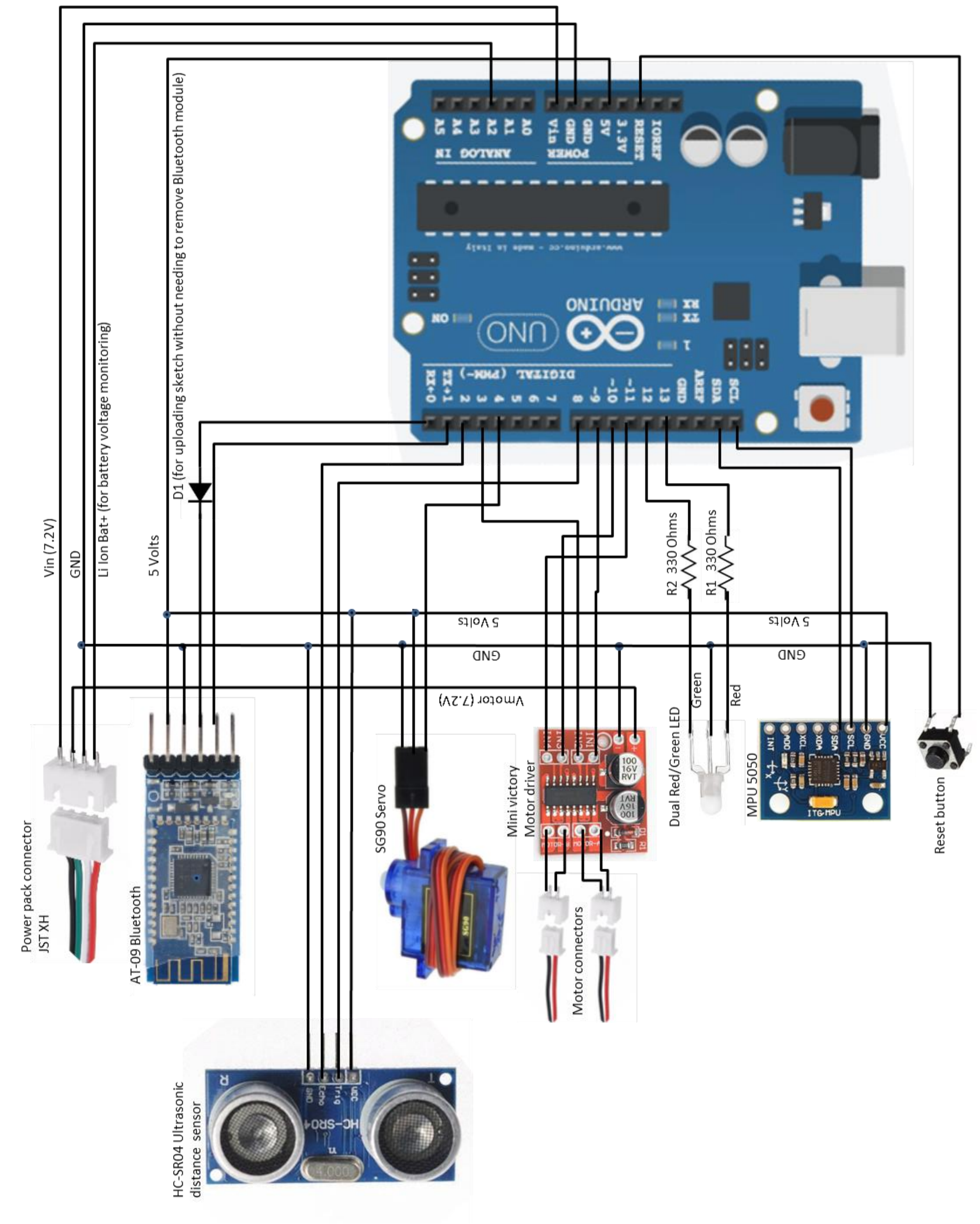
- the dual colour LED is connected to Arduino Uno pins D13 (red) and D12 (green).

- the signal/control pin to the SG90 servo is connected to Arduino Uno pin D4.

- the motor driver board input pin connections are:
  IN1 connected to Uno pin D9
  IN2 connected to Uno pin D3
  IN3 connected to Uno pin D10
  IN4 connected to Uno pin D11
  See also http://www.instructables.com/id/Tutorial-for-Dual-Channel-DC-Motor-Driver-Board-PW/

- the Li battery power pack has a 4 wire connector:
  (1) Li battery +ve (used to monitor battery voltage via Uno pin A2)
  (2) GND (also connected to Li battery -ve)
  (3) Vmotor: to drive the motors
  (4) Vin: to Vin of the Arduino Uno (the Uno supplies 5V to the MPU6050 and the BT module)

- the battery and motor connector plug/socket are JST XH type connectors. The battery connector is a 4-pin JST XH, and the motor connector is a 2-pin JST XH. (Note Vin and Vmotor are set to 7.2V)



Vin (7.2V)
GND
Li Ion Bat+ (for battery voltage monitoring)
5 Volts
D1 (for uploading sketch without needing to remove Bluetooth module)
5 Volts
GND
Vmotor (7.2V)
R2 330 Ohms
R1 330 Ohms
5 Volts
GND
Green
Red
Power pack connector
JST XH
AT-09 Bluetooth
SG90 Servo
Mini victory
Motor driver
Motor connectors
Dual Red/Green LED
MPU 5050
Reset button
HC-SR04 Ultrasonic
distance sensor

Li Ion Battery Pack

The Li Ion battery pack provides two separate power supplies: Vin for the Arduino and other electronics and servo, and Vmotor for the drive motors.

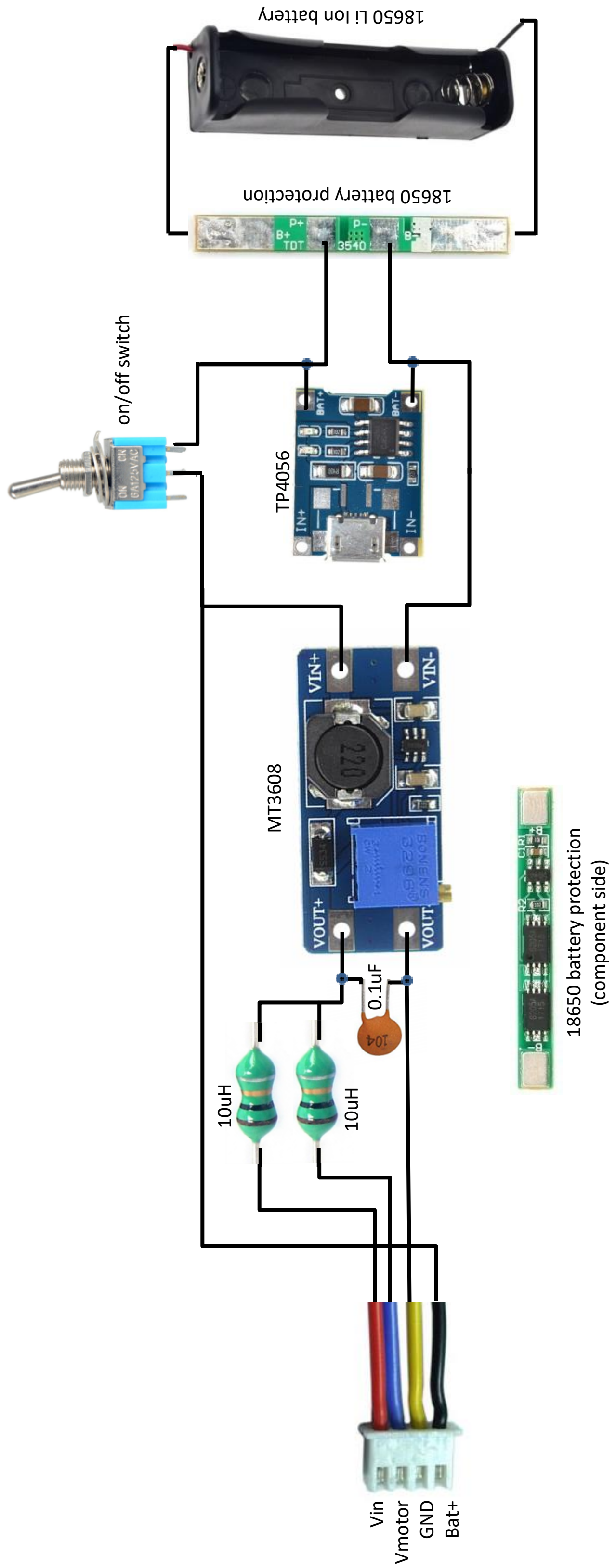
The battery pack consists of a battery protection circuit, a charging module (TP4056), a DC-DC step up converter (MT3608) and a filter circuit (inductor and capacitor).

The protection circuit protects the Li Ion battery from overvoltage during charging, under voltage during discharge and limits the max current from the battery to 4 Amps.

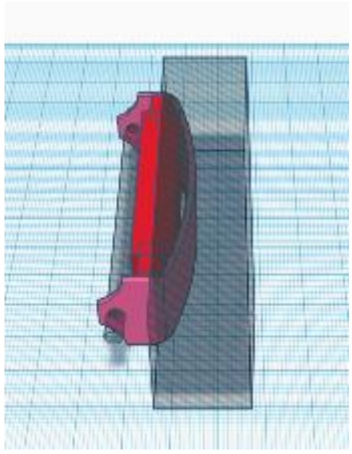
The MT3608 ten turn potentiometer is adjusted for a VOUT+ of 7.2 Volts.

Note, there is a 0.1uF ceramic capacitor soldered between VOUT+ and VOUT- and there is a 10uH inductor in series with VOUT+ and connected to Vin and Vmotor outputs. The inductor is to reduce noise from the Vmotor supply interfering with Vin to the Arduino.

The battery pack container is available on TinkerCAD (search for “balrobot” in the TinkerCAD gallery).



AA-Battery Pack



It is also possible to use a non-rechargeable battery pack instead of the Li battery pack.

A 6 x AA battery holder can be positioned on the back of the robot and held in place by two clips. The clip can be downloaded from TinkerCAD (search in “3D designs” for “balrobot” in the TinkerCAD gallery).

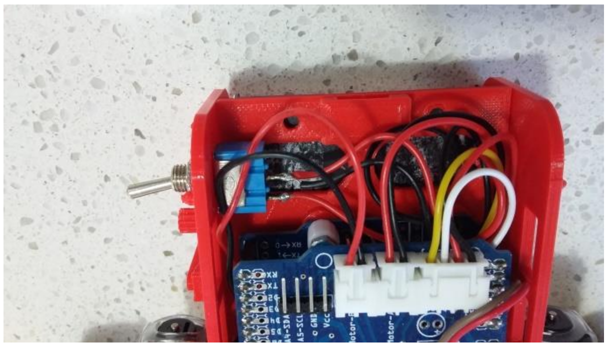
https://www.tinkercad.com/things/4Gxd4K6ipSd-6xaa-battery-holder-clip

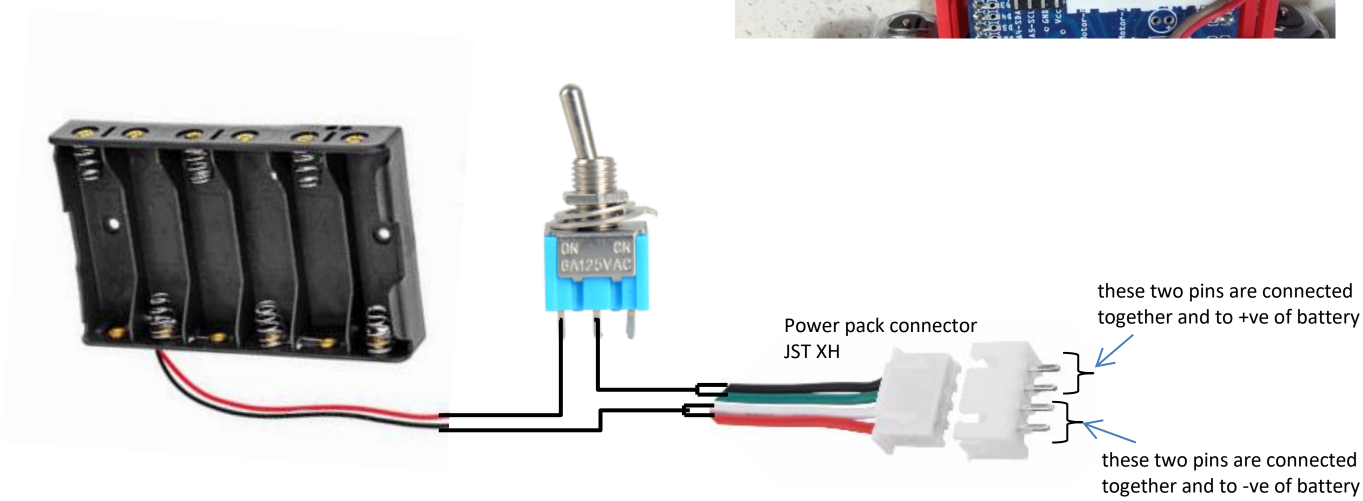
Note, the servo for the arm needs to be removed, so there will no longer be a movable arm.

The power switch is positioned through the hole already incorporated in the sides of the chassis at the top of the robot.

On the four pin power plug, connect pins 1 and 2, and pins 3 and 4 together as shown in the circuit below.

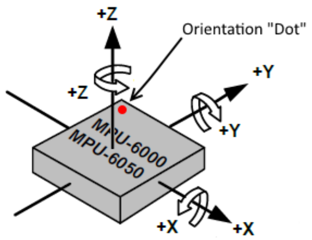
Note also, the charge status LED on the robot shield will always be green and will no longer indicate the status of the battery charge.



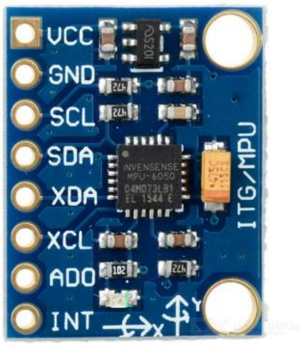


Balancing Robot Maths !

The maths behind the balancing robot is simpler and more interesting than you might think !



Firstly let's have a quick look at the MPU-6050 accelerometer and gyroscope sensor. The x, y and z directions of the MPU-6050 have an accelerometer and gyroscope senor as shown in the diagram – note also the x and y directions are marked on the board.



Accelerometers are devices that measure acceleration - the rate of change of the velocity of an object, measured in units of m/s^2 (meters per second per second) or sometimes expressed in G-forces (g). The G-force for us here on Earth is 9.8 m/s^2. So, when the MPU-6050 is lying on the desk the accelerometer in the z direction will measure 9.8 m/s^2 (or 1 g), and 0 m/s^2 (or 0 g) in the x and y directions.

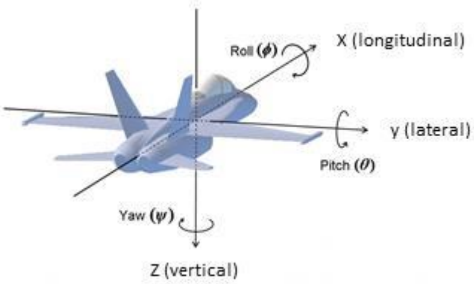
Gyro sensors, also known as angular velocity sensors, are devices that measure angular velocity. Angular velocity is the change in rotational angle per unit of time, generally measured in units of deg/s (degrees per second).

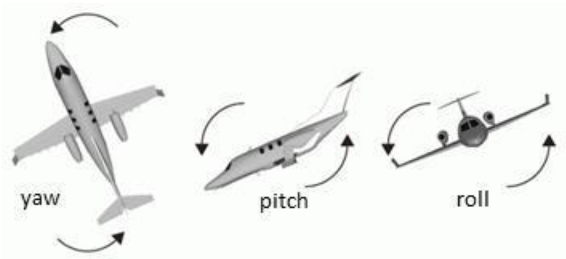
Accelerometers give an absolute way of finding angles (because gravity can be used as a reference direction), but are sensitive to noise (vibration). Gyroscopes are less sensitive to vibrations, but tend to drift over time. We need to combine the two to obtain stable measurements (called fusion).

Maths for calculating pitch angle from accelerometers

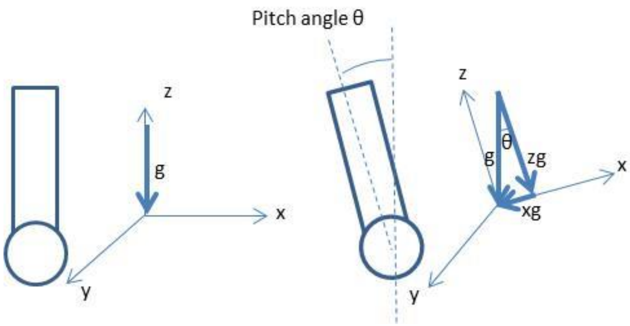
Vehicles that are free to move in three dimensions can change their direction around three axes:

- longitudinal (x) axis (the axis the vehicle is moving in) - motion about this axis is called roll
- lateral (y) axes - motion about this axis is called pitch
- vertical (z) axis - motion about this axes is called yaw





On earth the acceleration due to gravity g ($g=9.8m/s^2$) is always acting downwards, so we can use this as a reference direction to calculate pitch, roll and yaw angles from the accelerometers.



To keep our robot stable we are only interested in the pitch angle (we use this angle to drive the robot wheels to keep the robot upright).

When the robot is upright and stationary, the accelerometer in the z direction measures the full g value and the x and y accelerators measure zero. However when the robot tilts (pitches), the z and x accelerators each measure a component of g (zg and xg). So to calculate the pitch angle θ we can use trigonometry:

$$xg = g \times Sin(\theta)$$
$$\theta = \sin^{-1}(\frac{xg}{g})$$

In Arduino code this is:

```
acc_x_data = Wire.read()<<8|Wire.read();  // read the x accelerometer from MPU-6050
pitch_angle_acc = asin(acc_x_data/9.8) * 57.29578 ;  // pitch angle in degrees
```

Note: asin() is the Arduino function for $\sin^{-1}()$, it returns results in Radians. To convert to degrees multiply by $\frac{360}{2\pi}$ = 57.29578.

Maths for calculating pitch angle from gyros

We said earlier that accelerometers are sensitive to noise (vibration), and although gyros are less sensitive to noise, they tend to drift over time. So if we combine the two we obtain a stable pitch angle measurement.

Therefore we also need to calculate the pitch angle from the gyros. Again we only need to measure the pitch angle (not roll and yaw), and we can do this by taking gyro measurements about the y axis.

Gyros measure angular velocity (in degrees/sec). So to calculate an angle through which an object has rotated we need to integrate the gyro measurement over time (just like integrating velocity to obtain distance travelled).

To do this integration numerically we sample (take measurements) the y gyro on the MPU6050 at regular intervals (at period = T secs, e.g. at 4 milliseconds), add them all up and multiply by the period T:

$$Angle_{rotated} = T \times \sum_{t=start}^{current\ t} Gyro_{measurement}$$

A very simple way to do this calculation is to add the previous value $of\ Angle_{rotated}$ to the new $Gyro_{measurement} \times T$. In Arduino code this is (note the += in the second line of code):

```
gyro_y_data = Wire.read()<<8|Wire.read();       // read the y gyro from MPU-6050
pitch_angle_gyro += gyro_y_data * T;            // integrate to obtain an angle
```

Putting it all together - Obtaining a stable pitch angle

We have now calculated the pitch angle from both the accelerometers and the gyros. We now need to combine these to obtain a stable pitch angle, called fusion.

Over the short term, we want to use the gyro data, because it is not susceptible to noise (vibration), but over the long term we want to use the accelerometer data as it does not drift.

There is an optimal way to do this called Kalman filtering, but this is complicated to tune and requires a lot of code. A simpler method which is nearly as good for our robot is called a Complementary filter.

A complementary filter weights the pitch angles from the acc and gyro and adds them together:

$$pitch\_angle = (1 - \alpha) \times pitch\_angle\_gyro + \alpha \times pitch\_angle\_acc$$

and then sets the *pitch_angle_gyro* to the new filtered *pitch_angle*, so the drift in the gyro pitch angle is corrected over time. The weighting factor α is made small so that individual measurement noise in the acc derived pitch angle does not have a large affect and averages out over time.

Putting all the acc and gyro pitch angle Arduino code together:

```
// calculate pitch angle using acc data
acc_x_data = Wire.read()<<8|Wire.read();  // read the x accelerometer from MPU-6050
pitch_angle_acc = asin(acc_x_data/9.8) * 57.29578;   // pitch angle in degrees

// calculate pitch angle using gyro data
gyro_y_data = Wire.read()<<8|Wire.read(); // read the y gyro from MPU-6050
pitch_angle_gyro += gyro_y_data * T;      // integrate to obtain an angle

// complementary filter
pitch_angle = pitch_angle_gyro * 0.9996 + pitch_angle_acc * 0.0004;

pitch_angle_gyro = pitch_angle;           // corrects the pitch_angle_gyro drift
```