Getting Started with Arduino and Ginobot

# Overview

This guide will demonstrate how to connect an Arduino with the Ginobot and be able to command the Ginobot from the Arduino.

# Prerequisites

To be able to command the Ginobot with an Arduino, the following prerequisites are required:

1. Ginobot
2. Arduino Microcontroller
3. 4 Dupont Wires, two male-to-female and two male-to-male.
4. Ginobot PCB Adaptor

# Wiring the Arduino to Ginobot

The Ginobot is able to communicate with the Arduino via the I2C protocol (Inter-Integrated Circuit).

The library provided for the Arduino uses the SCL pin (Clock) and SDA pin (Data). In addition, the Vin pin and ground pin on the Arduino are required.

**IMPORTANT NOTE**: Only connect the 9V pin of the Ginobot to the Vin pin of the Arduino. Connecting the 9V pin to any other pin on the Arduino will irreversibly damage the board!

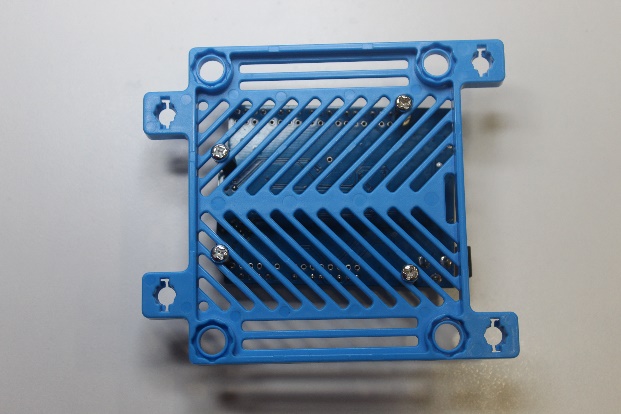
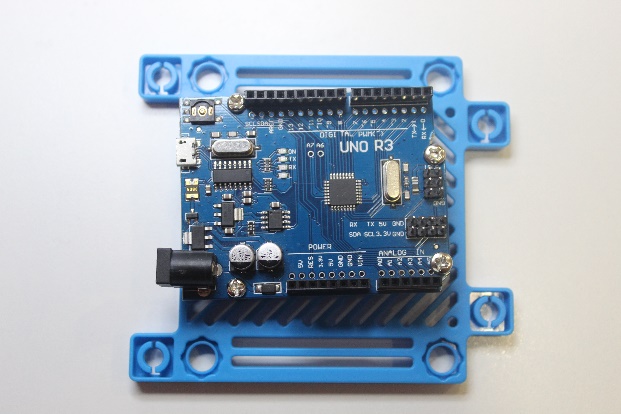
The Ginobot has a dedicated header for connecting external electronics. This is labeled as “HOST MCU”.

The Arduino must be connected in the following way:

|  |  |
| --- | --- |
| Arduino | Ginobot “HOST MCU” Header Pin |
| SCL | SCL |
| SDA | SDA |
| Vin | 9V |
| GND | GND |

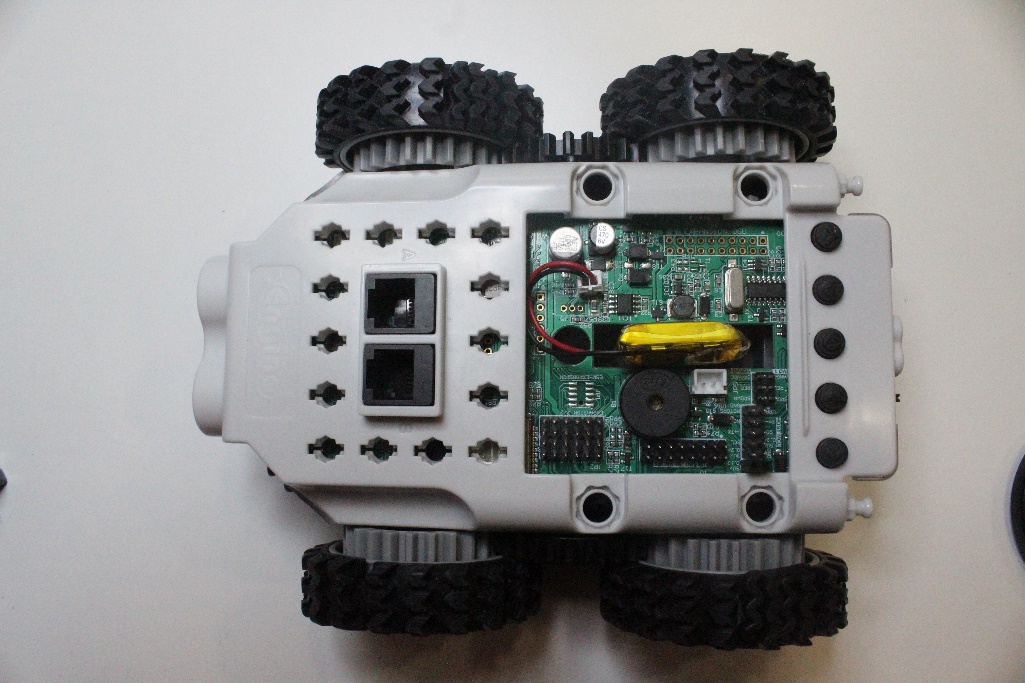
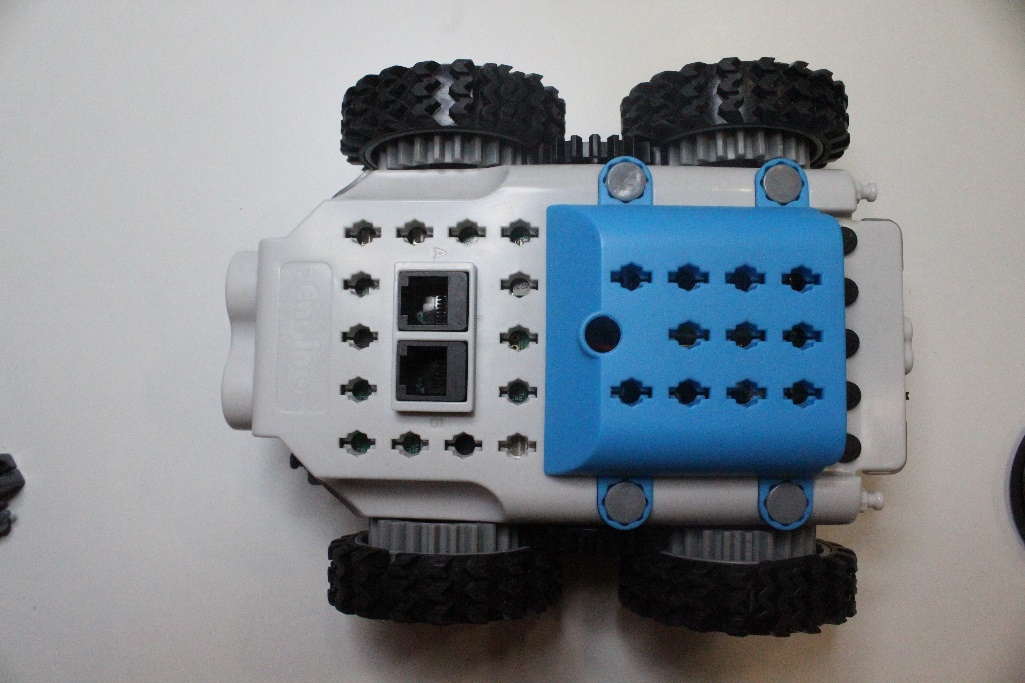
## STEP 1

Attach the Arduino to the Ginobot PCB Adaptor.



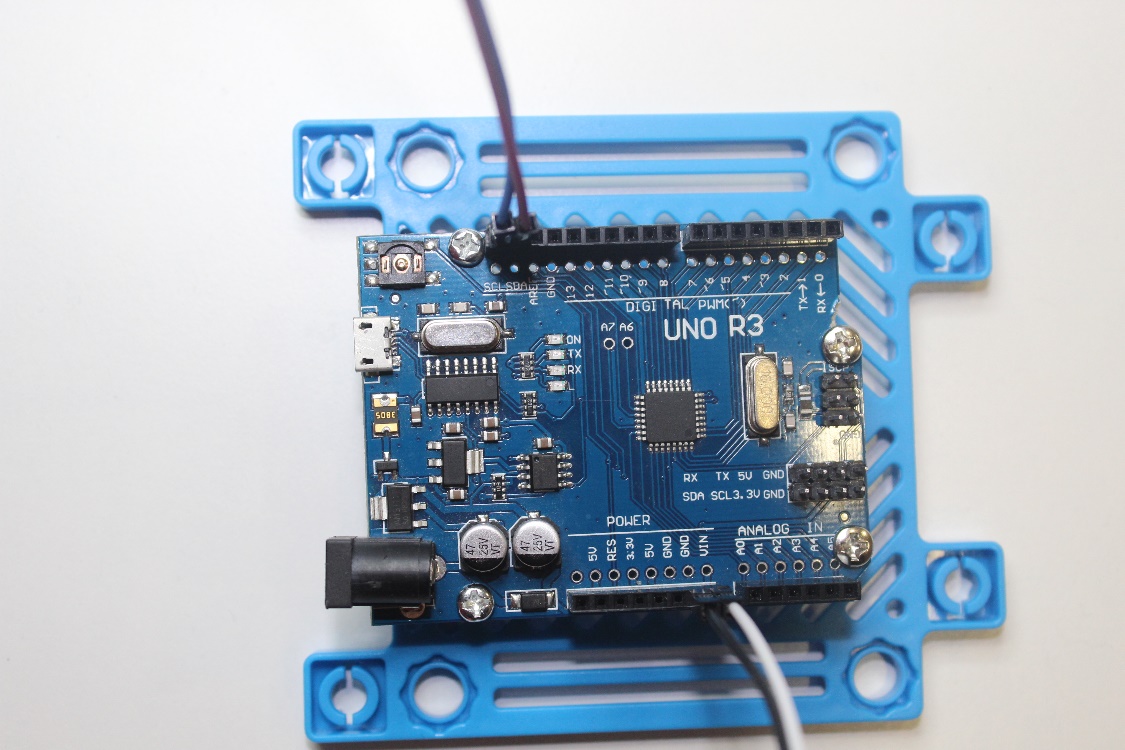
## STEP 2

Remove the Ginobot cover to expose the internal electronics.



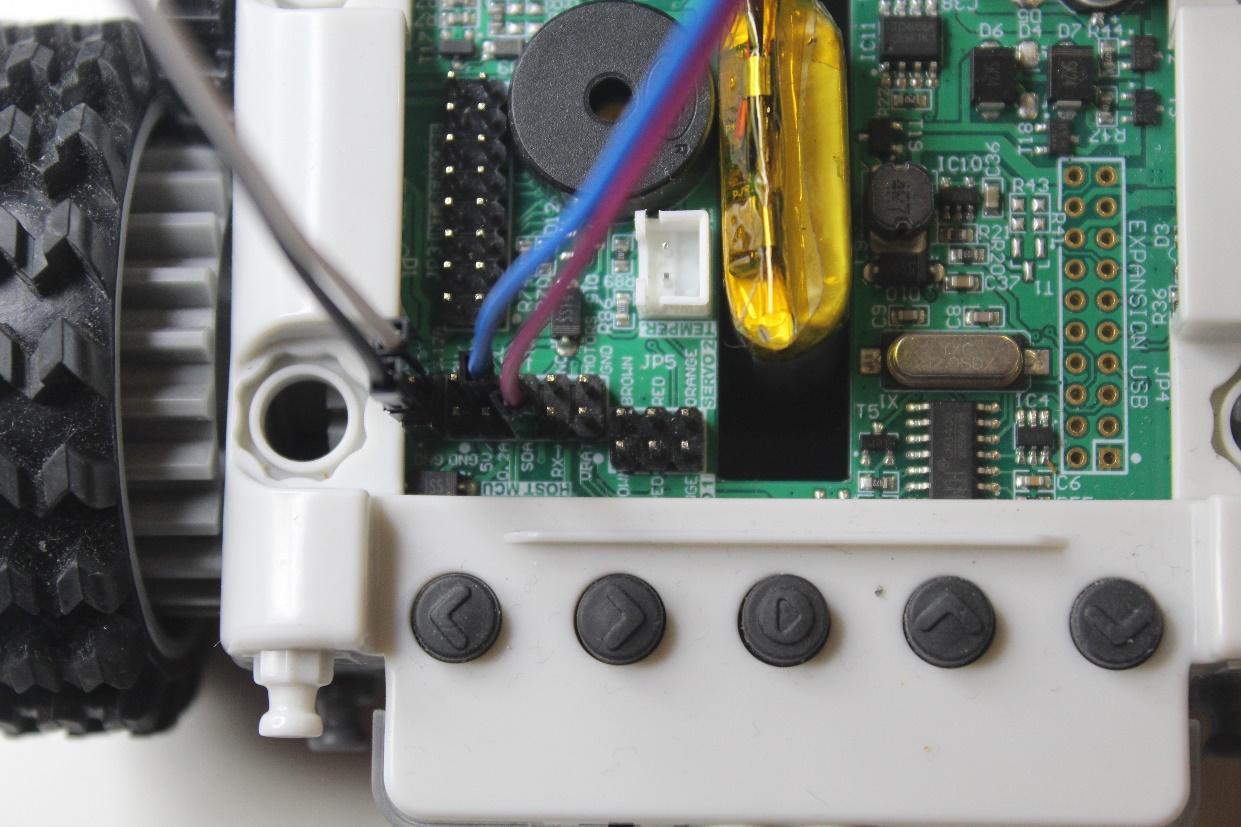
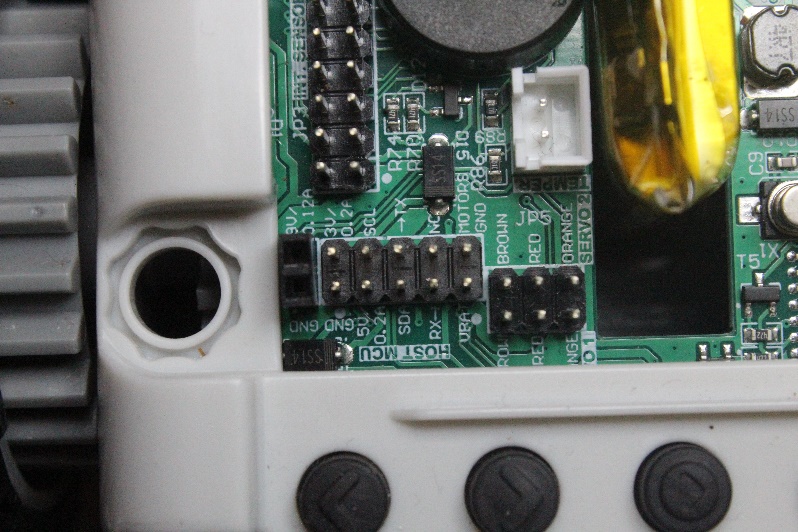
## STEP 3

Attach the Dupont wires to the Arduino. Please refer to the table in section 3 for wiring.



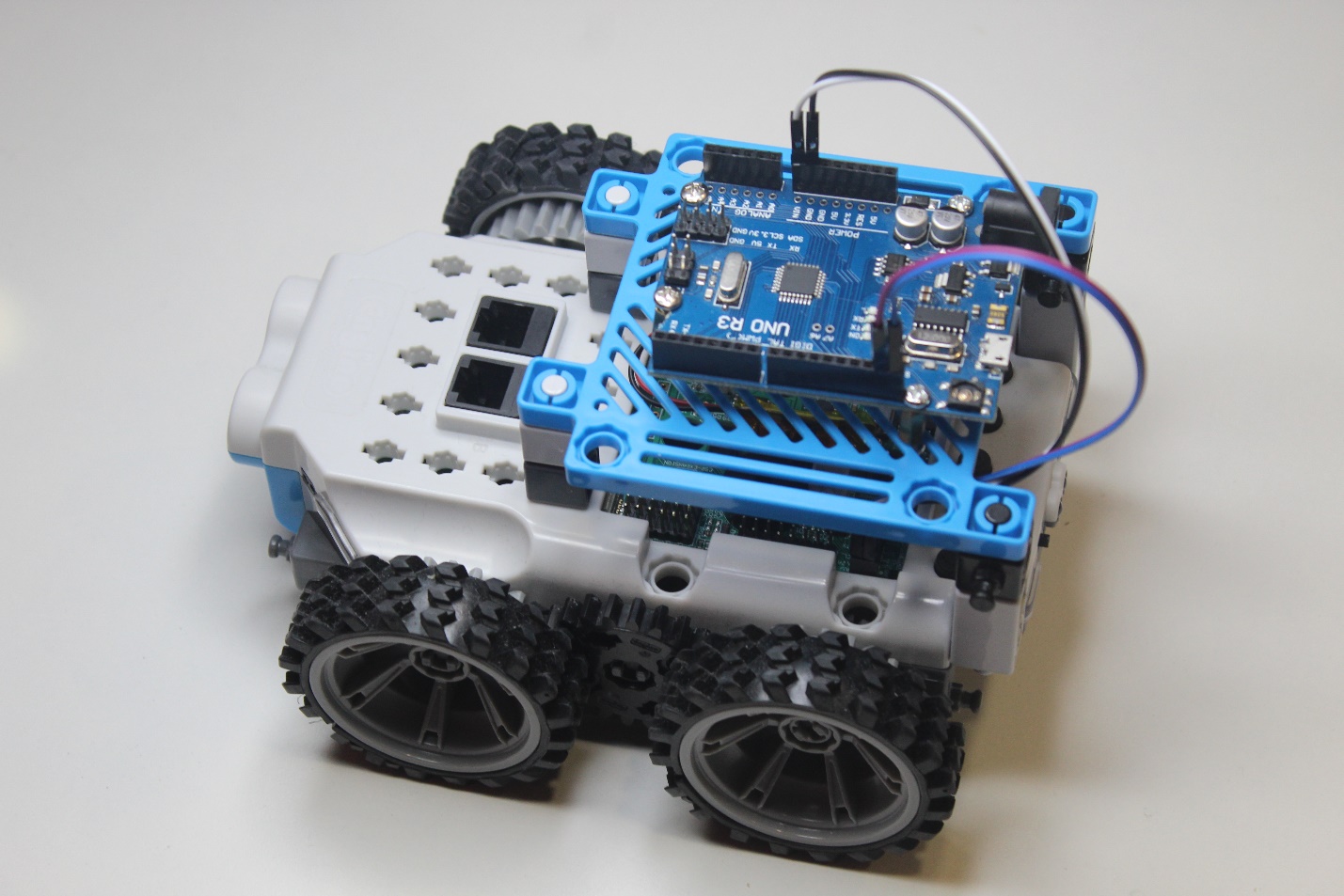
## STEP 4

Attach the Dupont wires to the Ginobot “HOST MCU” header. Please refer to the table in section 3 for wiring.



## STEP 5

Attach the Ginobot PCB Adaptor to the Ginobot.



You are now ready to command the Ginobot via the Arduino!

# Programming the Arduino

A library has been developed to allow the Arduino to communicate with the Ginobot. This can be downloaded from INSERT LINK HERE.

The tutorial assumes working knowledge of the Arduino platform and will not cover the basics of connecting and flashing the Arduino.

The Arduino typically programmed using the Arduino IDE which can be downloaded from <https://www.arduino.cc/en/software>

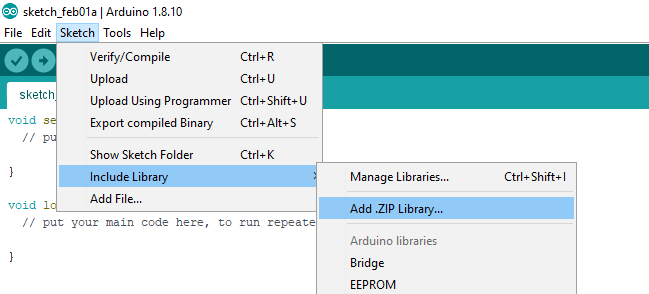
To get started, follow the procedure below.

## STEP 1

Add the Ginobot library to the Arduino IDE.

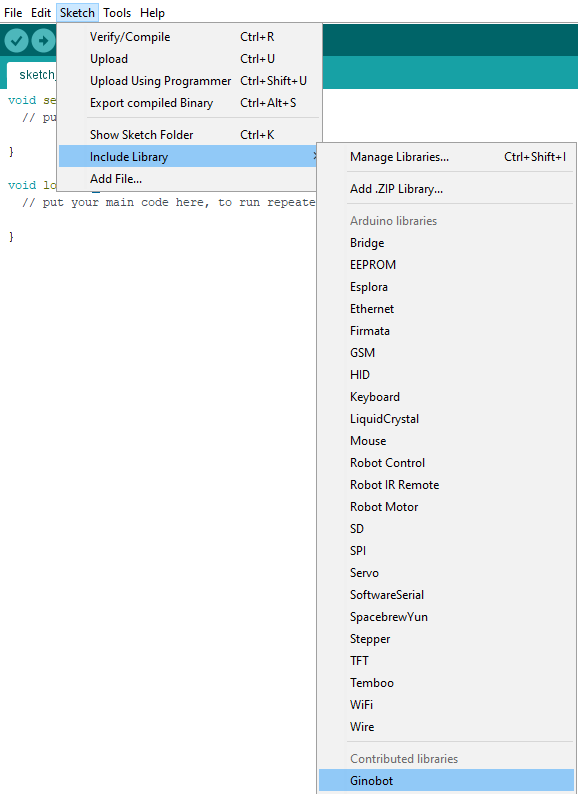
From the IDE, select “Sketch”, then “Include Library” and select “Add .ZIP Library…”.

Navigate to the downloaded Ginobot library for Arduino and select the zip file.



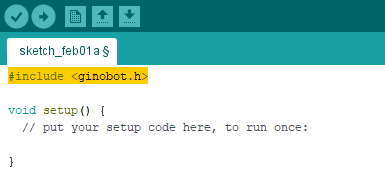
## STEP 2

To add the Ginobot library to your sketch, select “Sketch”, then “Include Library” and select “Ginobot” that will appear under the “Contributed Libraries” section.



## STEP 3

You will notice that a line of code was added to the top of your program. You are now ready to start programming the Arduino to control the Ginobot!



# Supported Commands

## List of Read Functions of the Ginobot class:

**get\_distance\_front\_mm()**

Returns the ultrasonic distance in mm. Requires ultrasound sensor present on Ginobot.

**get\_IR\_FL\_analog()**

Returns the Front Left IR sensor analog value (0-100).

**get\_IR\_FR\_analog()**

Returns the Front Right IR sensor analog value (0-100).

**get\_IR\_BACK\_analog()**

Returns the Back IR sensor analog value (0-100).

**get\_IR\_FL()**

Returns the Front Left IR sensor digital value (1 or 0).

**get\_IR\_FR()**

Returns the Front Right IR sensor digital value (1 or 0).

**get\_IR\_BACK()**

Returns the Back IR sensor digital value (1 or 0).

**get\_bottom\_left()**

Returns the Bottom Left IR sensor digital value (1 or 0).

**get\_bottom\_left\_white()**

Returns the Bottom Left color sensor white intensity (0-255).

**get\_bottom\_left\_red()**

Returns the Bottom Left color sensor red intensity (0-255).

**get\_bottom\_left\_green()**

Returns the Bottom Left color sensor green intensity (0-255).

**get\_bottom\_left blue()**

Returns the Bottom Left color sensor blue intensity (0-255).

**get\_bottom\_right()**

Returns the Bottom Right IR sensor digital value (1 or 0).

**get\_bottom\_right\_white()**

Returns the Bottom Right color sensor white intensity (0-255).

**get\_bottom\_right\_red()**

Returns the Bottom Right color sensor red intensity (0-255).

**get\_bottom\_right\_green()**

Returns the Bottom Right color sensor green intensity (0-255).

**get\_bottom\_right\_blue()**

Returns the Bottom Right color sensor blue intensity (0-255).

**get\_wheel\_right\_speed()**

Returns the current speed of the right wheel (0-100)

**get\_wheel\_right\_position()**

Returns the current position of the right wheel in clicks (0 – 65535)

**get\_wheel\_left\_speed()**

Returns the current speed of the left wheel (0-100)

**get\_wheel\_left\_position()**

Returns the current position of the left wheel in clicks (0 – 65535)

**get\_battery\_level()**

Returns the current battery level (0 – 4)

## List of Write Functions of the Ginobot class:

**move(x)**

Commands the Ginobot to move with speed x (-100 – 100). Moves forward if x is a positive value; moves backwards if x is a negative value.

**move\_forward(x)**

Commands the Ginobot to move forward with speed x (0 - 100)

**move\_backward(x)**

Commands the Ginobot to move backward with speed x (0 - 100)

**rotate(x)**

Commands the Ginobot to rotate with speed x (0-100). Rotates right if x is a positive value; rotates left if x is a negative value.

**rotate\_right(x)**

Commands the Ginobot to rotate right with speed x (0-100)

**rotate\_left(x)**

Commands the Ginobot to rotate left with speed x (0 -100)

**stop()**

Commands the Ginobot to stop moving.

**wheel\_left\_spin(x)**

Commands the Ginobot to turn the left wheel with speed x (0-100)

**wheel\_right\_spin(x)**

Commands the Ginobot to turn the right with speed x (0 -100)

**move\_distance\_mm(x, y, z)**

Commands the Ginobot to move a specific distance in mm, where x is the distance, y is the wheel diameter in mm and z is the speed (0 – 100). A positive x value will cause the Ginobot to move forward; a negative x value will cause the Ginobot to move backward.

**rotate\_clicks(x, y)**

Commands the Ginobot to rotate for a specific amount of clicks, where x is the number of clicks and y is the speed (0 – 100). A positive x value will cause the Ginobot to rotate right; a negative x value causes the Ginobot to rotate left.

**wheel\_left\_turn(x, y)**

Commands the Ginobot to rotate the left wheel for a specific number of clicks, where x is the number of clicks and y is the speed (0 – 100). A positive x value causes the wheel to rotate clockwise; a negative x value causes the wheel to spin anticlockwise.

**wheel\_right\_turn(x, y)**

Commands the Ginobot to rotate the right wheel for a specific number of clicks, where x is the number of clicks and y is the speed (0 – 100). A positive x value causes the wheel to rotate clockwise; a negative x value causes the wheel to spin anticlockwise.

**set\_IR\_FL\_threshold(x)**

Sets the threshold value of the Front Left IR sensor (0-100). When the sensor's analog value is below this then the sensor's digital value turns to 0, when it is higher than this, its digital value turns to 1.

**set\_IR\_FR\_threshold(x)**

Sets the threshold value of the Front Right IR sensor (0-100). When the sensor's analog value is below this then the sensor's digital value turns to 0, when it is higher than this, its digital value turns to 1.

**set\_IR\_Back\_threshold(x)**

Sets the threshold value of the back IR sensor (0-100). When the sensor's analog value is below this then the sensor's digital value turns to 0, when it is higher than this, its digital value turns to 1.

**set\_IR\_bot\_L\_threshold(x)**

Sets the threshold value of the bottom left IR sensor (0-100). When the sensor's analog value is below this then the sensor's digital value turns to 0, when it is higher than this, its digital value turns to 1.

**set\_IR\_bot\_R\_threshold(x)**

Sets the threshold value of the bottom right IR sensor (0-100). When the sensor's analog value is below this then the sensor's digital value turns to 0, when it is higher than this, its digital value turns to 1.

**set\_front\_lights(x,y,z)**

Sets the Ginobot front RGB red LED with intensity x (0-255), green LED with intensity y (0-255) and blue LED with intensity z (0-255)

**Set\_back\_lights(x,y,z)**

Sets the Ginobot back RGB red LED with intensity x (0-255), green LED with intensity y (0-255) and blue LED with intensity z (0-255)

**set\_buzzer\_frequency(x)**

Sets the Ginobot buzzer frequency in Hz to x (0-65535). Setting to 0 turns the buzzer off.

**servo1\_position(x)**

Sets the position for servo1 in degrees to x (0-180).

For more information, and help getting started, please consult the examples.