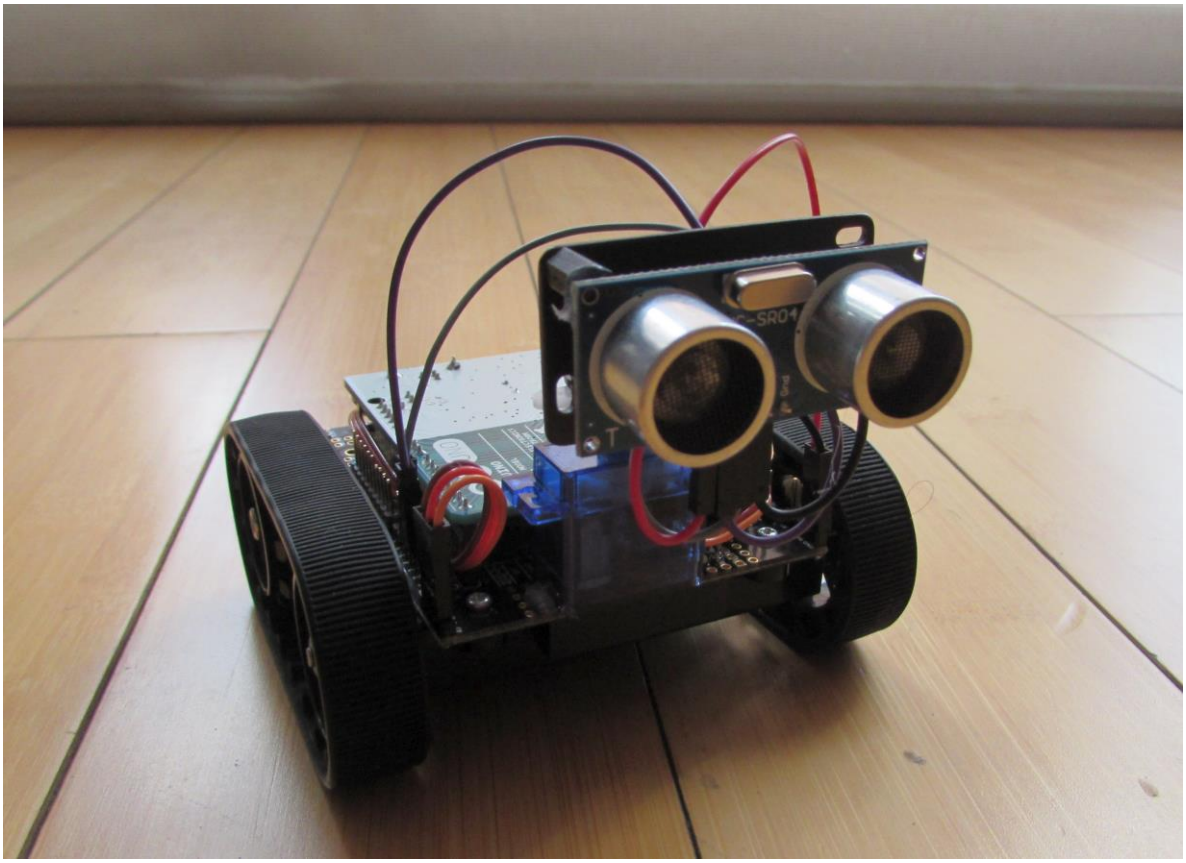
Sapphire Autonomous Vehicle V 1.0

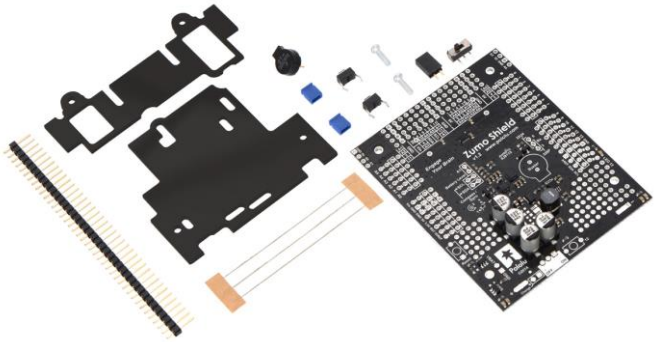


The Sapphire autonomous vehicle is a small basic robot with the capability of avoid obstacles, it is a system based on Arduino UNO board and Pololu Zumo Shield V 1.2

Required Hardward

Board Arduino UNO R3
https://www.arduino.cc/en/Main/ArduinoBoardUno

Pololu Zumo Shield kit for Arduino V 1.2
https://www.pololu.com/product/2508



2x Micro Metal Gearmotor HP 75:1
https://www.pololu.com/product/2361

Any Standard Servo Motor, this sample used this one
https://www.pololu.com/product/2820

Ultrasonic sensor HC-SR04
http://www.robotshop.com/en/hc-sr04-ultrasonic-range-finder.html

Multi-Purpose Sensor Housing
http://www.lynxmotion.com/p-397-multi-purpose-sensor-housing.aspx

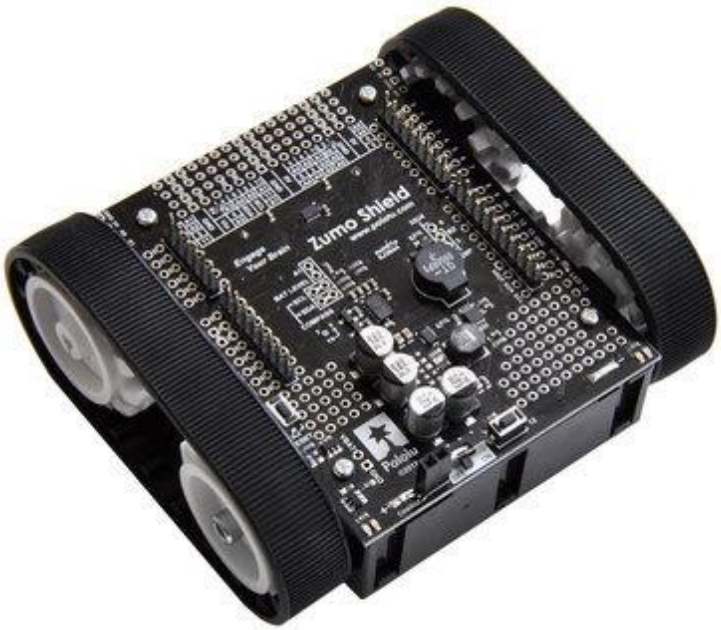
4x Female/Female jumper wires
http://www.robotshop.com/en/200mm-f-f-40-pin-jumper-wire.html

4x Rechargeable NiMH AA Battery: 1.2 V, 2200 mAh (4 batteries)
https://www.pololu.com/product/1003

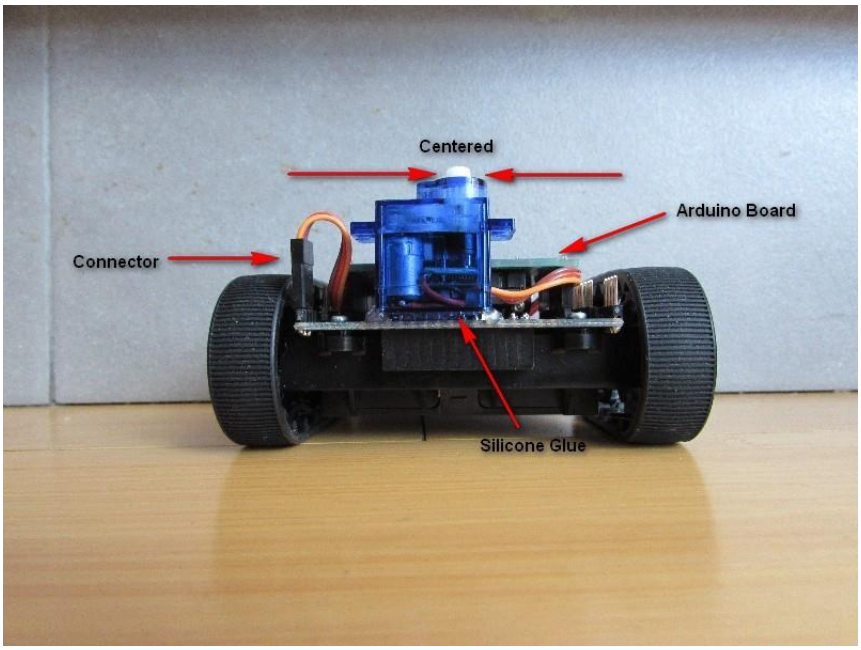
Assembling the Robot

The Zumo Shield for Arduino is the core of Sapphire since it provides the chassis, the shield board for Arduino which includes different interesting built-in devices like the accelerometer, magnetometer and the motor driver to control the motors and it provides also expansions to attach more devices. The official guide to assembly the parts is located in the manufacturer website, please read carefully the section 2 of this guide:
https://www.pololu.com/docs/0J57/2

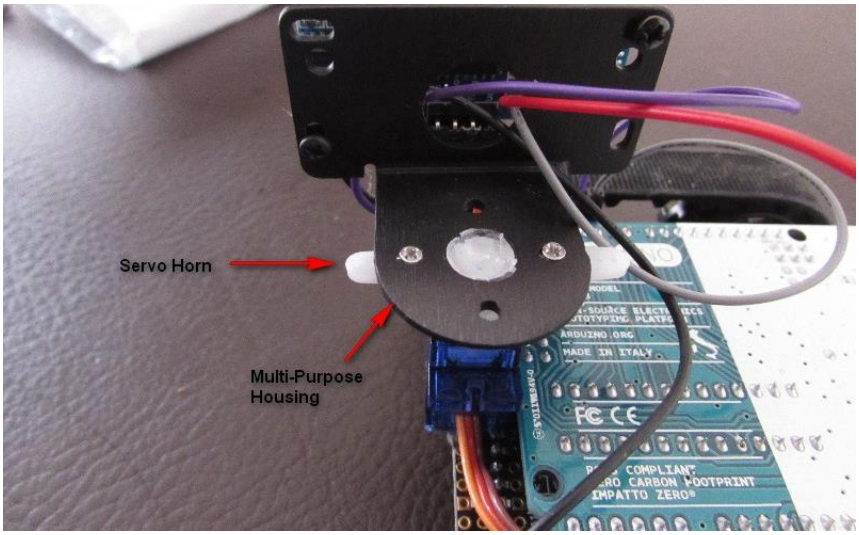
After the soldering the parts the Sapphire vehicle must look like this:



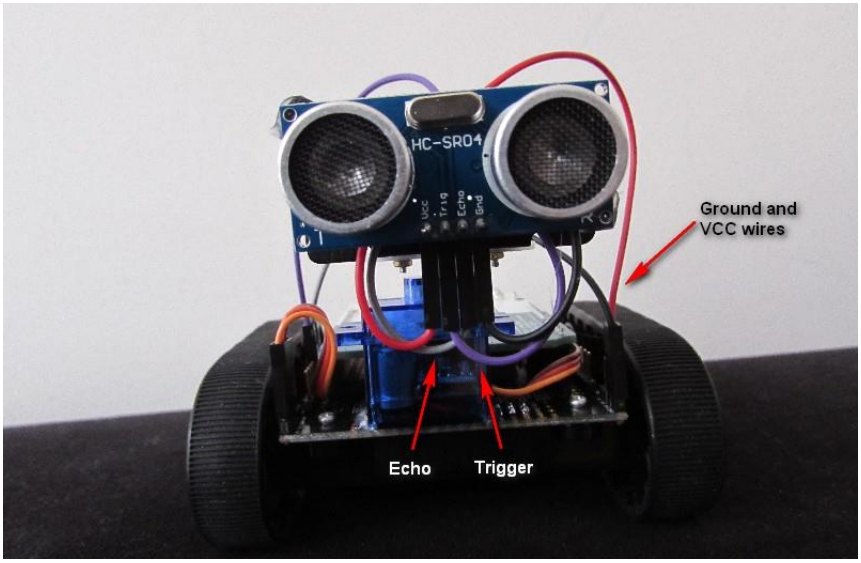
After the core chassis is assembled is time to attach the servo motor in the front of the shield, first attach the Arduino board in the top of the shield and using silicone glue to attach the servo motor, be careful to centering the white wheel with respect the frontal shield.



The Multi-purpose housing will support the ultrasonic sensor, but before to attach the sensor a servo horn must be plug into the housing, it will allow rotate the sensor.



With the servo horn attached to the housing now the ultrasonic sensor must be attached to the housing by following the small guide provided by the manufacturer: http://www.lynxmotion.com/images/html/mpsh-01.htm



Software

The software package or Sapphire provides everything needed to make the robot working properly, the package includes some third party libraries also used to manipulate the servo motor and the integrated compass of the Zumo shield, the compass is composed by one 3-axis accelerometer and 3-axis magnetometer, the software package contains the following libraries:

Wire: This library is standard of Arduino so it is included in every installation of the Arduino IDE.

LSM303: This is a third party library provided by the manufacturer of the Zumo Shield, the git hub repository is located here:
https://github.com/pololu/lsm303-arduino

NewPing: This library is provided in the Arduino website, it allows to control ultrasonic sensors using interruptions, this technic offers more speed, the library is detailed here:
http://playground.arduino.cc/Code/NewPing

SoftwareSerial: This library is also standard of Arduino so it is included in every installation of the Arduino IDE.

SapphireMotor: This is the controller for the DC motors.

SapphireServo: This is the controller for the servo motor, it uses a timer to control the rotations of the servo properly.

SapphireMotion: The controller of the motion of the Sapphire, it uses the SapphireMotor and the compass integrated in the Zumo shield to control the rotations.

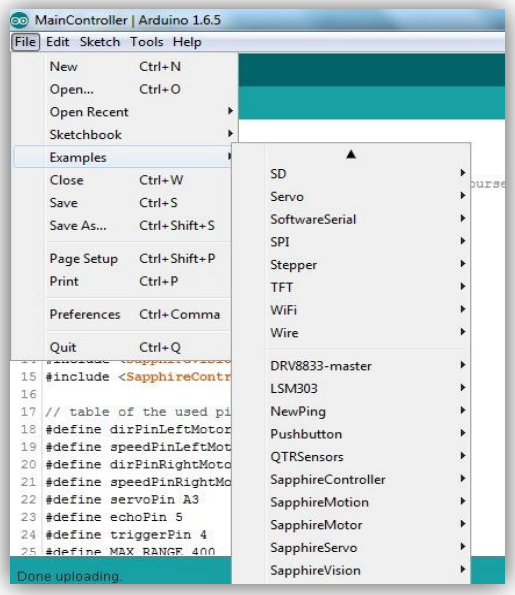
SapphireVision: The controller of the vision system for Sapphire, it controls the servo motor and the ultrasonic sensor.

SapphireController: This is the main controller of the system, in the examples folder is the main sketch to load in the Sapphire.

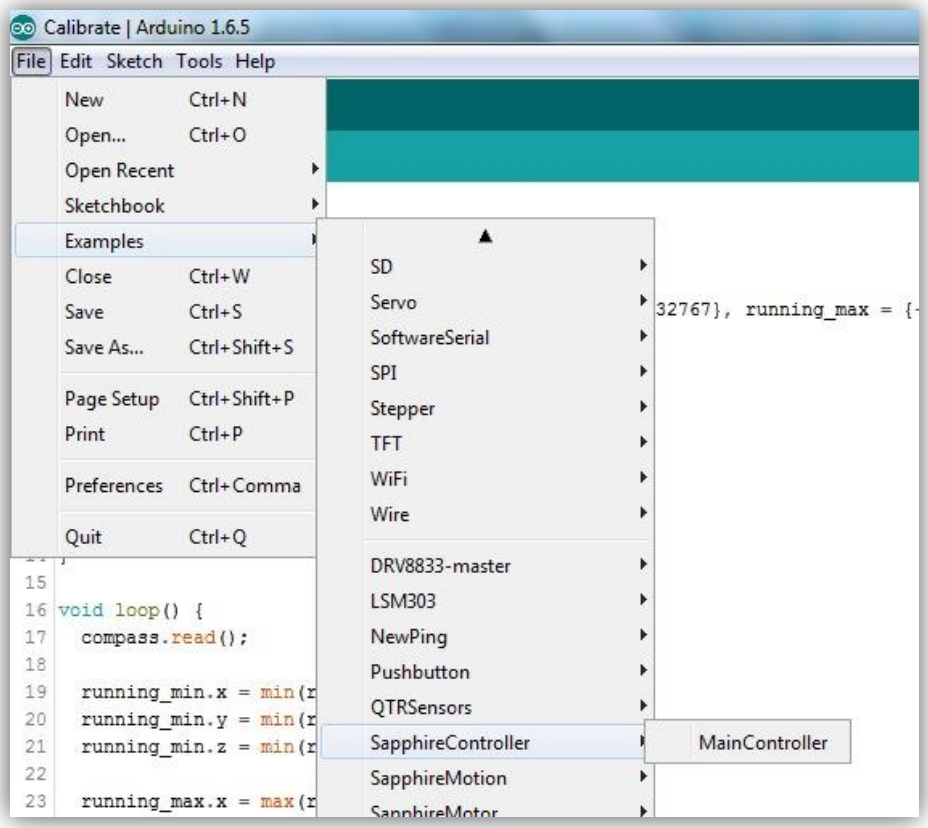
Installation

Close the Arduino IDE and put every every folder of the Sapphire software package in the main library folder of Arduino IDE, if working under Windows normally it is located in **Program Files (x86)\Arduino\libraries**

To check if the libraries are installed correctly open the Ardunino IDE and go to File, Examples, scroll down and check if the Sapphire libraries appears; everyone provides a sample sketch to test the library, just load any sample provided and follow the code, it is well commented line by line.

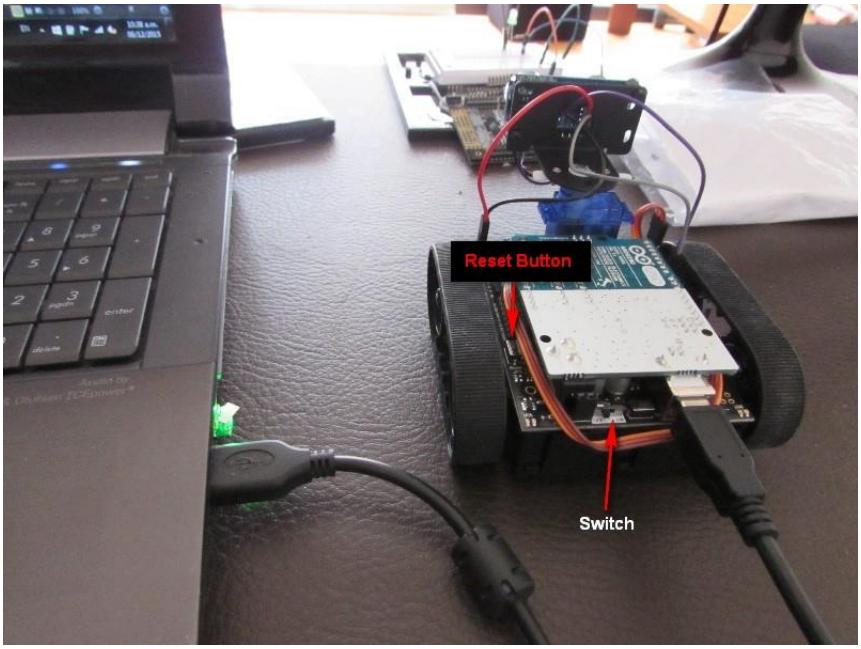


The MainController sketch of Sapphire vehicle is located in the SapphireController example, just fo to File, Examples, SapphireController and load the MainController sketch;



Now push the verify button and compiles the library, if everything is fine the following message must be shown in the main console.

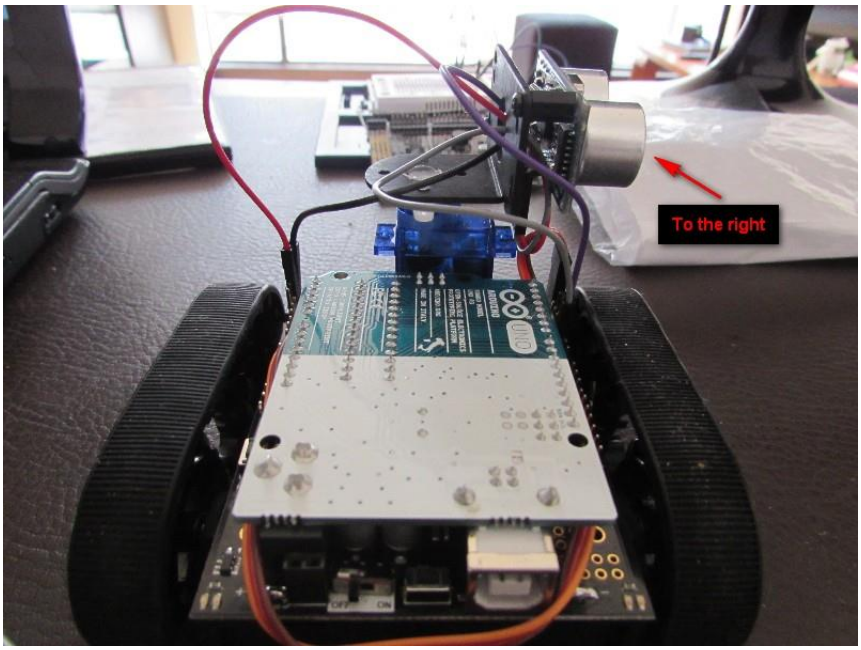
Now is time to connect the Sapphire vehicle to the USB port and upload the sketch with the upload button.



Be careful with the switch at the left of the USB connector, if it is turned off, the servo will rotate but the wheels will not move, if it is turned on, the servo will move first and after the wheels will move.

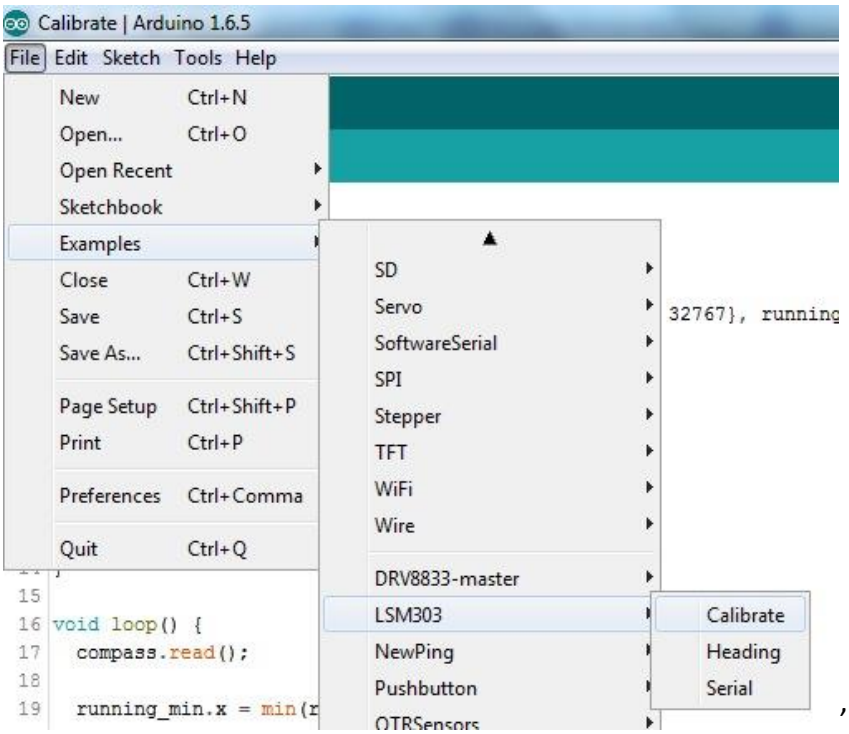
Calibration

Two calibration process need to be made, one is the calibration of the servo and the other is the calibration of the compass system (magnetometer and accelerometer). For Servo calibration unplug the USB and take off the multi-purpose housing with the ultrasonic sensor and push the reset button, turn on the Sapphire vehicle and you will hear two sounds coming from the servo motor, at the end of the second sound the servo is located to the right and it will begin to rotate grade by grade to the left, after the second sound, turn off the Sapphire vehicle and attach again the ultrasonic sensor but placed to the right.



The first thing the Sapphire vehicle will do is rotate from right to left the ultrasonic sensor and read all the distances grade by grade.

The second calibration step is regarding the compass system, to do that plug the USB and open Arduino IDE, go to File, Examples, LSM303 and load the sketch Calibrate:



Upload this sketch to the Sapphire vehicle and open the serial monitor, it will show the readings of the compass system, one vector for the minimum limit and another vector for the maximum limit, every vector showing the three values, one for every axis X, Y and Z. Now start rotating slowly the Sapphire vehicle in every axis until complete 360 degrees in every direction, you will note that the values will change during the rotation, after rotating enough the values in the serial monitor will keep stable, so take note of these values. Now open this file:

Program Files (x86)\Arduino\libraries\SapphireMotion\SapphireMotion.cpp

And go to the lines 30 and 31 and change the values of the vectors with the correspond values that you got previously. Compile and upload the MainController scketch, if everything is fine the Sapphire vehicle is ready.