

# Hummingbird Arduino API Description

---

The Hummingbird can be used with Arduino in two modes:

1. Use the Hummingbird Arduino library API to write programs that control the regular Hummingbird inputs and outputs (top-side ports). As the top-side ports use nearly all Arduino pins, it may be a challenge to use shields in addition to the Hummingbird library. Refer to the pin-mapping guide at the end of this document to see how Hummingbird I/O maps to the Arduino Pin mapping.
2. Use Hummingbird as an Arduino Leonardo with an integrated SPI-based Motor/Servo shield; all other inputs/outputs will need to be manually initialized by other libraries or the program.

## Initialization

### **void init()**

Fully initializes the Hummingbird. By default it will set as outputs all LED and Vibration motor pins (Arduino pins 2 to 12), it will also enable intensity control for LEDs (using Timer 3) and for vibration motors (using Timer 1).

### **void initOnlyMotorsAndServos()**

Initializes just the SPI communication with Hummingbird's integrated motor/servo controller. All regular Arduino pins except for digital I/O pin 8 are left alone. Pin 8 is the slave select line for SPI communication with the motor/servo controller.

### **void init(boolean turnOnLEDFading, boolean turnOnVibrationSpeed, boolean turnOnLEDandVibrationOutputs)**

Provides user control over specific initialization steps.

#### Parameters

*turnOnLEDFading*: Initializes software interrupt on Timer 3 that provides intensity control of the LEDs. If this is not initialized, the LED functions only turn LEDs on or off and Timer 3 is free for user control.

*turnOnVibrationSpeed*: Initialized speed control over the vibration motors, using Timer 1. If this is not initialized, the vibration motors can only be turned on or off and Timer 1 is free for user control.

*turnOnLEDandVibrationOutputs*: Sets Arduino pins 2-7 and 9-13 as outputs.

## LEDs

### **void setLED(byte port, byte intensity)**

Sets the intensity of the LED on port 1, 2, 3, or 4.

#### Parameters

*port*: The LED's port: 1, 2, 3, or 4

*intensity*: The intensity of light from the LED, 0 is off, 255 is fully on. If LED fading is turned off, all non-zero values will result in the LED being fully on.

### **void setStatusLED(byte intensity)**

Sets the intensity of the green status LED (also tied to 'L' LED on the Arduino side).

#### Parameters

*intensity*: The intensity of light from the LED, 0 is off, 255 is fully on. If LED fading is turned off, all non-zero values will result in the LED being fully on.

### **void setTriColorLED(byte port, byte red, byte green, byte blue)**

Sets the color and brightness of the tri-color LED on port 1 or 2

#### Parameters

*port*: The LED's port: 1 or 2

*red*: The intensity of red light from the LED, 0 is off, 255 is fully on. If LED fading is turned off, all non-zero values will result in the red LED being fully on.

*green*: The intensity of green light from the LED, 0 is off, 255 is fully on. If LED fading is turned off, all non-zero values will result in the green LED being fully on.

*blue*: The intensity of blue light from the LED, 0 is off, 255 is fully on. If LED fading is turned off, all non-zero values will result in the blue LED being fully on.

### **void turnOnLEDFade() and void turnOffLEDFade()**

Turns on or off fading for LEDs

## Motors and Servos

### **void setMotor(byte port, int velocity)**

Sets the velocity of the gear motor on port 1 or 2.

#### Parameters

*port*: The gear motor's port: 1 or 2

*velocity*: The power sent to the motor, the range is -255 to 255: 0 is off, 255 is full speed in one direction, -255 is full speed in the other direction. The motor has a dead zone where it will not spin for values between approximately -30 to 30.

### **void setServo(byte port, byte degrees)**

Sets the angle of the servo on ports 1, 2, 3 or 4.

#### Parameters

*port*: The servo's port: 1, 2, 3, or 4

*degrees*: The servo angle, specified in degrees, with a range of 0 to 180.

### **void setVibration(byte port, byte intensity)**

Sets the intensity of the vibration motor on port 1 or 2.

#### Parameters

*port*: The vibration motor's port: 1 or 2

*intensity*: The speed of the vibration motor, 0 is off, 255 is fully on. If vibration motor speed control is turned off, all non-zero values will result in full speed for the motor.

### **void turnOnVibrationMotorSpeed() and void turnOffVibrationMotorSpeed()**

Turns speed control on or off for vibration motors.

## Sensors

### **float readInputVoltage()**

Returns a reading in volts of the voltage of VIN. Voltages higher than 10V will read as 10V.

### **int readSensorValue(byte port)**

Reads the analog value at the port specified. Values can range from 0 to 1023, with 0 corresponding to 0V and 1023 corresponding to 5V.

#### Parameters

*port*: The sensor's port: 1, 2, 3, or 4

## Pin-mapping: Hummingbird Duo to Arduino Leonardo

### Digital I/O

Arduino Pin	Hummingbird Use
0	Not used
1	Not used
2	Controls LED port 2
3	Controls LED port 1
4	Controls RGB port 1, green element
5	Controls RGB port 2, red element
6	Controls RGB port 2, green element
7	Controls RGB port 1, red element
8	Controls SPI slave select for motor/servo controller.
9	Controls Vibration motor 1
10	Controls Vibration motor 2
11	Controls RGB port 2, blue element
12	Controls RGB port 1, blue element
13	Controls Status LED on top side, 'L' LED on bottom
SDA	Controls LED port 2
SCL	Controls LED port 1

### Analog In

Arduino Pin	Hummingbird Use
0	Controls LED port 4
1	Sensor 1 input
2	Sensor 2 input
3	Sensor 3 input
4	Sensor 4 input
5	Battery voltage monitor

### SPI

Arduino Pin	Hummingbird Use
MOSI	MOSI to motor/servo controller, can be used for additional modules
MISO	Not used
SCK	SCK to motor/servo controller, can be used for additional modules