

Description of Python Driver for Hummingbird Bit V0\_1

**Introduction:**

Hummingbird Bit is the next generation of the Hummingbird Robotics Kit. It can be interfaced with various software like Make Code , Snap , Scratch , Birdblox , Java and Python. This document contains a detailed description of how to use the python driver.

**Contents**

**1. Requirements**

**2. API**

**i. Class**

a. Micro:bit

b. Hummingbird Bit

**ii Outputs**

a. setLED

b. setTriLED

c. setPositionServo

d. setRotationServo

e. setDisplay

f. print

**iii. Inputs**

a. getAcceleration

b. getCompass

c. getMagnetometer

d. getButton

e. isShaking

f. getOrientation

g. getLight

h. getSound

i. getDistance

j. getDial

Requirements, API UM0100

**1.Requirements**

Python Version **3.2 or greater**

Bluebird Connector 1.0 or greater

Operating Systems tested on Windows 10

**2.API**

**i. Class**

With bluebird connector you can connect either a Micro:bit or Hummingbird bit. Once you have connected the device use the respective class.

**a. Micro:bit**

Micro bit class has all the methods to receive and set peripherals on the Micro:bit itself.

**b. Hummingbird Bit**

Includes all the methods of a Micro bit and others to control the outputs and inputs of the bit.

**ii. Outputs**

a. **setLED(port\_no,intensity)**

|  |  |
| --- | --- |
| First argument – port\_no | Select the port (1-3) of the LED to be changed. |
| Second argument -- intensity | Select the intensity value (0-100).  0 off  100 full intensity |
| Return | Integer 1/0. To Be Decided |
| Description | Change the LED intensity attached any three ports corresponding to LED. |
| Example | setLED(1,50) -- LED on 50% intensity  setLED(2,0) -- LED off  setLED(3,100) -- LED full intensity |

b. **setTriLED(port\_no,r\_intensity,g\_intensity,b\_intensity)**

|  |  |
| --- | --- |
| First argument – port\_no | Select the port (1-3) of the LED to be changed. |
| Second argument – r\_intensity  Third argument -- g\_intensity  Fourth argument -- b\_intensity | Select the intensity value (0-100).  0 off  100 full intensity |
| Return | Integer 1/0. To Be Decided |
| Description | Change the Tri Color LED intensity attached to any two ports corresponding to Tricolor LED. |
| Example | a.setTriLED(1,0,0,50)  Port --1  a. R Intensity – 0  b. G Intensity – 0  c. B Intensity – 50  a.setTriLED(2,0,0,0)  Port – 2, RGB LED off |

c. **setPositionServo(port\_no,angle)**

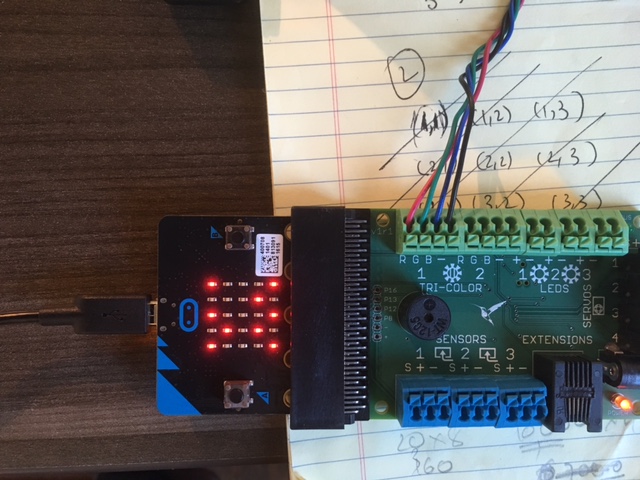
|  |  |
| --- | --- |
| First argument – port\_no | Select the port (1-4) of the servo to be changed. Connect the position servo to that port you desire. |
| Second argument -- Angle | Select the intensity value (0-180).  0 Position zero  180 |
| Return | Integer 1/0. To Be Decided |
| Description | Set the angle of the servo which is connected in the desired port |
| Example | setPositionServo(1,180) – End Point, Port 1  setPositionServo(2,0) – Start Point, Port2 |

d. **setRotationServo(port\_no,speed)**

|  |  |
| --- | --- |
| First argument – port\_no | Select the port (1-4) of the servo to be changed. Connect the rotation servo to the desired port |
| Second argument -- Angle | Select the intensity value (-100 - 100). |
| Return | Integer 1/0. To Be Decided |
| Description | Move the rotation servo connected in desired port in clock wise or anti-clock wise direction |
| Example | setRotationServo(1,0) – No Movement  setRotationServo(2,20) – Rotate clockwise  setRotationServo(2,-30)—Rotate Anti Clockwise |

e. **setDisplay(Display\_string)**

|  |  |
| --- | --- |
| First argument – Display\_string | String of 1’s and 0’s to control the corresponding LEDs on the LED matrix. |
| Return | Integer 1/0. To Be Decided |
| Description | Can control the 25 LEDs on the LED Matrix on the micro bit by appropriately choosing the values in the LED\_string |
| Example | setDisplay(“1010101000000000101010101”)  Image -1 |



**Image-1**

f. **print(print\_string)**

|  |  |
| --- | --- |
| First argument – print\_string | Maximum length of the string is 18. Print the desired string on the display. |
| Return | Integer 1/0. To Be Decided |
| Description | Flash a string on the LED display screen on the microbit |
| Example | Print(“TEST123”) |

**iii.Outputs**

**a. getAcceleartion()**

|  |  |
| --- | --- |
| Return | (float X, float Y, float Z) 🡪 m/sec^2 |
| Description | Get the acceleration values of X,Y,Z in m/sec^2 from the micro::bit connected. |
| Example | (X,Y,Z) = getAcceleration() |

**b. getCompass()**

|  |  |
| --- | --- |
| Return | Float value 🡪 degree(0-360) |
| Description | **0 should roughly correspond to North.** |
| Example | degree = getCompass() |

**c. getMagnetometer()**

|  |  |
| --- | --- |
| Return | (float X, float Y, float Z) 🡪 uT |
| Description | Get the Magnetometer values of (X,Y,Z) in uT from the micro::bit connected. |
| Example | (X,Y,Z) = getMangetometer() |

**d. getButtons(button\_name)**

|  |  |
| --- | --- |
| First argument – button\_name | ‘A’ or ‘B’ |
| Return | “true”/”false” |
| Description | Get the state of the button on the micro:bit. True when the button is pressed and false when the button is not pressed. |
| Example | Button\_state = getButtons(‘A’) |

**e. isShaking()**

|  |  |
| --- | --- |
| Return | “true”/”false” |
| Description | True if the device is shaken. |
| Example | shake = isShaken() |

**f. getOrientation()**

|  |  |
| --- | --- |
| Return | “ScreenUp”/”ScreenDown”/”TiltLeft”/”TiltRight” “LogoUp”/”LogoDown” |
| Description | Returns the value of Orientation in which micro::bit is aligned. |
| Example | orientation = getOrientation() |

**g. getLight(Port)**

|  |  |
| --- | --- |
| First argument – button\_name | 1-3 |
| Return | 0-100 |
| Description | 0 corresponds to if there is no light , and 100 if there is light at the maximum level. |
| Example | Light =getLight(1) , port 1 |

**g. getSound(Port)**

|  |  |
| --- | --- |
| First argument – button\_name | 1-3 |
| Return | 0-100 |
| Description | 0 corresponds to if there is no sound , and 100 if there is sound at the maximum level. |
| Example | Light =getSound(1) , port 1 |

**h. getDistance(Port)**

|  |  |
| --- | --- |
| First argument – button\_name | 1-3 |
| Return | 0-200 in cms |
| Description | Ultrasound range is from 3cm – 2meters. |
| Example | Distance = getDistance(1) , port 1 |

**h. getDial(Port)**

|  |  |
| --- | --- |
| First argument – button\_name | 1-3 |
| Return | 0-100 |
| Description | One of the extreme point is 0 and other is 100 |
| Example | Dial = getDial(1) , port 1 |