

Microbit Class Methods

Constructors

Method Signature: `Microbit()`

Description: Constructor that creates an object corresponding to a micro:bit. This method assume that the micro:bit is the only device connected in the BlueBird Connector.

Example: `myBit = Microbit()`

Method Signature: `Microbit(device)`

Description: Constructor that creates an object corresponding to a micro:bit. This method requires a string equal to 'A', 'B', or 'C' that specifies the letter of the device in the BlueBird Connector.

Example: `myBit = Microbit('B')`

Output Methods

Method Signature: `setDisplay(LEDlist)`

Description: Sets the LED array of the micro:bit to display a pattern defined by a list of length 25. Each value in the list must be 0 (off) or 1 (on). The first five values in the array correspond to the five LEDs in the first row, the next five values to the second row, etc.

Example: `myBit.setDisplay([1,1,1,1,1,0,0,0,0,0,0,1,1,1,1,1,0,0,0,0,0,1,1,1,1,1])`

Method Signature: `setPoint(row, column, value)`

Description: Turn on or off a single LED on the micro:bit display. The position of the LED is given by the row and column parameters, which should both be between 1 and 5. The value of the LED must be 0 (off) or 1 (on).

Example: `myBit.setPoint(3,3,1)`

Method Signature: `print(message)`

Description: Print a string on the micro:bit LED array. The string must have 15 or fewer characters and should contain only digits and English letters (upper or lower case).

Example: `myBit.print("hello")`

Input Methods

Method Signature: `getButton(button)`

Description: Takes a string corresponding to a micro:bit button ("A" or "B") and returns a boolean value that is true if the button is

being pressed, and false otherwise.

Example: `print("Button A is pressed: ", myBit.getButton('A'))`

Method Signature: `isShaking()`

Description: Returns True if the micro:bit is shaking and False otherwise.

Example: `print("Shake Status: ", myBit.isShaking())`

Method Signature: `getOrientation()`

Description: Returns a string that represents the orientation of the micro:bit. The possible values are “Screen up”, “Screen down”, “Tilt left”, “Tilt right”, “Logo up”, “Logo down”, and “In between”.

Example: `print("Orientation: " + myBit.getOrientation())`

Method Signature: `getAcceleration()`

Description: Returns a list that contains the acceleration in m/s² in the x, y, and z directions.

Example: `print("Acceleration: ", myBit.getAcceleration())`

Method Signature: `getCompass()`

Description: Returns the direction of the micro:bit in degrees from magnetic north (0°-359°). The compass should be calibrated in the BlueBird Connector before using this method.

Example: `print("Compass Heading: ", myBit.getCompass())`

Method Signature: `getMagnetometer()`

Description: Returns a list that contains the value of the magnetic field in µT in the x, y, and z directions. The compass should be calibrated in the BlueBird Connector before using this method.

Example: `print("Magnetic Field: ", myBit.getMagnetometer())`

Method Signature: `public void stopAll()`

Description: Stops all outputs. This includes the LED display for the micro:bit and all lights and motors for the Hummingbird Bit.

Example: `myBit.stopAll()`

Hummingbird Class Methods

Hummingbird is a subclass of Microbit

Constructors

Method Signature: `Hummingbird()`

Description: Constructor that creates an object corresponding to a Hummingbird Bit. This method assume that the Hummingbird Bit is the only device connected in the BlueBird Connector.

Example: `myBit = Hummingbird()`

Method Signature: `Hummingbird(device)`

Description: Constructor that creates an object corresponding to a Hummingbird Bit. This method requires a string equal to "A", "B", or "C" that specifies the letter of the device in the BlueBird Connector.

Example: `myBit = Hummingbird('B')`

Output Methods

Method Signature: `setLED(port, intensity)`

Description: Sets an LED to a given intensity value. The method requires the port number of the LED (1-3) and an intensity value from 0-100. An intensity value of 0 turns the LED off.

Example: `myBit.setLED(1,100)`

Method Signature: `setTriLED(port, redIntensity, greenIntensity, blueIntensity)`

Description: Sets a tri-color LED to a given color by setting the intensities of the red, green, and blue elements inside it . The method requires the port number of the tri-color LED (1-2) and three intensity values from 0-100. Setting all three intensity values to 0 turns the LED off.

Example: `myBit.setTriLED(1,75,0,75)`

Method Signature: `setPositionServo(port, angle)`

Description: Sets a position servo to a given angle. The method requires the port number of the servo (1-4) and an angle from 0°-180°.

Example: `myBit.setPositionServo(1,90)`

Method Signature: `setRotationServo(port, speed)`

Description: Sets a rotation servo to spin at a given speed. The method requires the port number of the servo (1-4) and a speed between -100 and 100. A speed of 0 turns the motor off.

Example: `myBit.setRotationServo(1,100)`

Method Signature: `playNote(note, beats)`

Description: Plays a note using the buzzer on the Hummingbird Bit. The method requires a integer representing the note (32-135) and a number giving the number of beats (0-16). The number of beats can be a decimal number.

Example: `myBit.playNote(60,0.5)`

Input Methods

Method Signature: `getLight(port)`

Description: Takes an integer corresponding to the sensor port (1-3) and returns the intensity measured by the light sensor. The light measurement is a an integer between 0 and 100 (arbitrary units).

Example: `print("Light Sensor: ", myBit.getLight(1))`

Method Signature: `getDial(port)`

Description: Takes an integer corresponding to the sensor port (1-3) and returns the position of the dial sensor. The dial position is a an integer between 0 and 100 (arbitrary units).

Example: `print("Dial Sensor: ", myBit.getDial(1))`

Method Signature: `getDistance(port)`

Description: Takes an integer corresponding to the sensor port (1-3) and returns the distance in centimeters measured by the distance sensor. The sensor detects distance most accurately in the range of 2-200 cm.

Example: `print("Distance Sensor: ", myBit.getDistance(1))`

Method Signature: `getSound(port)`

Description: Takes an integer corresponding to the sensor port (1-3) and returns the intensity measured by the sound sensor. The sound measurement is a an integer between 0 and 100 (arbitrary units).

Example: `print("Sound Sensor: ", myBit.getSound(1))`

Method Signature: `public double getOther(int port)`

Description: Takes an integer corresponding to the sensor port (1-3) and returns the voltage measured by any sensor. The voltage measurement is a double between 0 and 3.3 V.

Example: `print("Sensor Voltage: ", myBit.getVoltage(1))`