Internet of Things

Home

Getting Started

Tutorials

How Tos

IoT in Action

Devices

BluetoothLE

Arduino 101

BBC micro:bit

Accelerometer

Button

IO Pin

# BluetoothLE

Bluetooth Low Energy, also referred to as Bluetooth LE or simply BLE, is a new communication protocol similar to classic Bluetooth except that it is designed to consume less power while maintaining comparable functionality. For this reason, Bluetooth LE is the preferred choice of communication with IoT devices that have limited power resources. Starting with Android 4.3, Google introduced built-in support for Bluetooth Low Energy. The BluetoothLE extension requires Android 5.0 or higher to avoid known issues with Google's Bluetooth LE support prior to Android 5.0.

Download the BluetoothLE extension (version 20181124)

## Version History

Build 20181124

- Handle dangerous permissions required on Android 6.0 Marshmallow and higher

Build 20171109

- Convert strings to numbers in WriteBytes, WriteShorts, WriteIntegers, WriteFloats

Home
Getting Started
Tutorials
How Tos
IoT in Action
Devices
BluetoothLE
Arduino 101
BBC micro:bit
Accelerometer
Button
IO Pin

Build 20171108

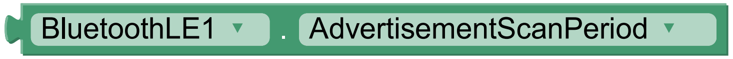
- Fix bug that prevented extensions built on top of BluetoothLE from working correctly when writing values

Build 20171107

- Typecheck inputs and cast where appropriate from list inputs
- Fix NullPointerException during scanning on some phones
- Allow 16-bit UUIDs to be used
- Fix link error with Lcom/google/common/collect/Lists;

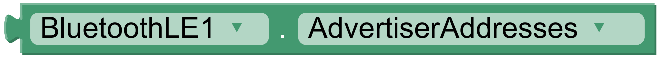
## Properties

- `AdvertisementScanPeriod` – Returns the value of ScanPeriod.



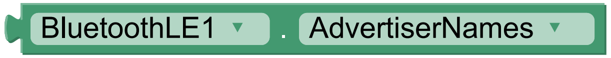
BluetoothLE1 ▾ . AdvertisementScanPeriod ▾

- `AdvertiserAddresses` – Returns a list of the addresses of devices found during Advertisement scanning.



BluetoothLE1 ▾ . AdvertiserAddresses ▾

- `AdvertiserNames` – Returns a list of the names of the devices found during Advertisment scanning.



Home

Getting Started

Tutorials

How Tos

IoT in Action

Devices

BluetoothLE

Arduino 101

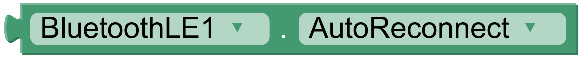
BBC micro:bit

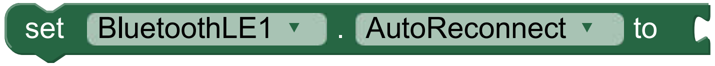
Accelerometer

Button

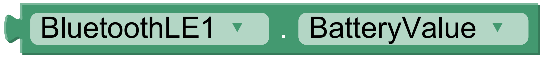
IO Pin

- `AutoReconnect` – If true, the application will attempt to reestablish a lost connection to a device due to link loss (e.g., moving out of range). This will not apply to connections that are disconnected by a call to the `Disconnect` method. Such disconnects will need to be reconnected via a call to `Connect` or `ConnectWithAddress` .

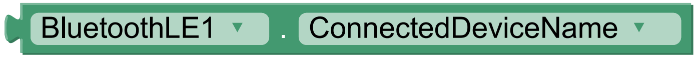




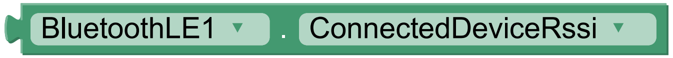
- `BatteryValue` – Returns the battery level.



- `ConnectedDeviceName` – The advertised name of the connected device. If no device is connected or Bluetooth low energy is not supported, this will return the empty string.



- `ConnectedDeviceRssi` – Returns the RSSI (Received Signal Strength Indicator) of connected device.



Home

Getting Started

Tutorials

How Tos

IoT in Action

Devices

BluetoothLE

Arduino 101

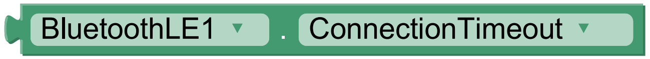
BBC micro:bit

Accelerometer

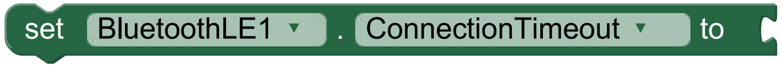
Button

IO Pin

- `ConnectionTimeout` – The amount of time, in seconds, that the BluetoothLE component will wait for a connection to be established with a device after a call to `Connect` or `ConnectWithAddress`. If a connection is not established in the given amount of time, the attempt will be aborted and the `ConnectionFailed` event will be run.

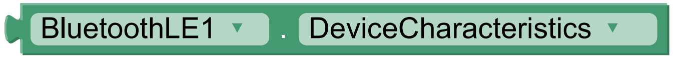


```
BluetoothLE1 . ConnectionTimeout
```



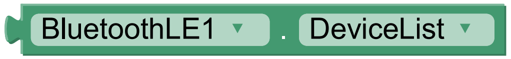
```
set BluetoothLE1 . ConnectionTimeout to
```

- `DeviceCharacteristics` – A list of triples, one for each characteristic advertised by the connected device, containing the service UUID, characteristic UUID, and the characteristic's name, if known. The format of the list will be ((service1 characteristic1 name1) (service2 characteristic2 name2) ...). If no device is connected or Bluetooth low energy is not supported, then an empty list will be returned.



```
BluetoothLE1 . DeviceCharacteristics
```

- `DeviceList` – Returns a sorted list of BluetoothLE devices as a String.



```
BluetoothLE1 . DeviceList
```

- `DeviceServices` – A list of pairs, one for each advertised service, indicating the service's UUID and its name, if known. The format of the list will be ((uuid1 name1)

Home

Getting Started

Tutorials

How Tos

IoT in Action

Devices

BluetoothLE

Arduino 101

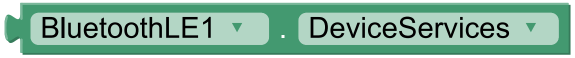
BBC micro:bit

Accelerometer

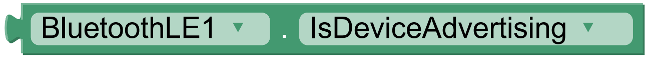
Button

IO Pin

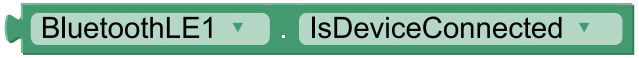
(uuid2 name2) ...). If no device is connected or Bluetooth low energy is not supported, then an empty list will be returned.



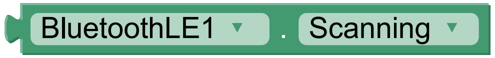
- `IsDeviceAdvertising` – Returns true if the device is currently advertising, false otherwise.



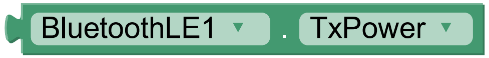
- `IsDeviceConnected` – Returns true if a BluetoothLE device is connected; Otherwise, returns false.



- `Scanning` – The scanning state of the Bluetooth low energy component.



- `TxPower` – Returns the transmission power.



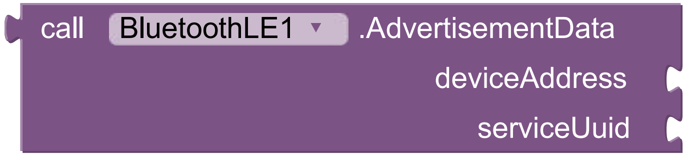
Internet of Things

## Methods

- AdvertisementData – Returns the advertisement data associated with the specified device address.

  **Parameters**:

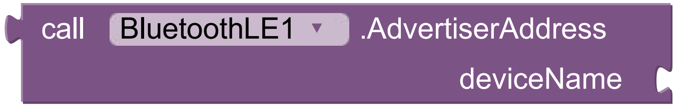
  - deviceAddress (*text*) — The Media Access Control (MAC) address of the target Bluetooth low energy device.
  - serviceUuid (*text*) — The unique identifier of the advertised service.



- AdvertiserAddress – Returns the address of the device with the name specified.

  **Parameters**:

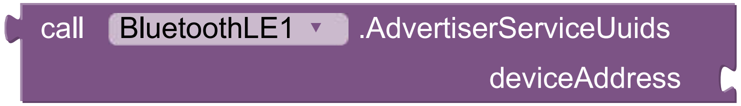
  - deviceName (*text*) — The advertised name of the target Bluetooth low energy device.



- AdvertiserServiceUuids – Returns the list of services available on the advertising device.

**Parameters**:

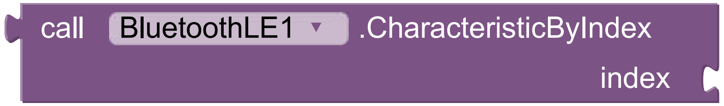
- `deviceAddress` (*text*) — The Media Access Control (MAC) address of the target Bluetooth low energy device.



- `CharacteristicByIndex` – Returns Unique ID of selected characteristic with index. Index specified by list of supported characteristics for a connected device, starting from 1.

  **Parameters**:

  - `index` (*number*) — The index of the desired characteristic, which must be between 1 and the length of the characteristic list.

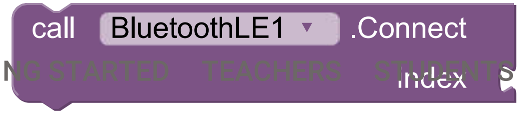


- `Connect` – Use the `Connect` method to connect to a Bluetooth low energy device at the given index in the device list.

  **Parameters**:

  - `index` (*number*) — The index of the target device, which must be between 1 and the length of the list.

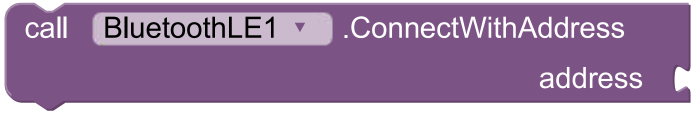
Home
Getting Started
Tutorials
How Tos
IoT in Action
Devices
BluetoothLE
Arduino 101
BBC micro:bit
Accelerometer
Button
IO Pin



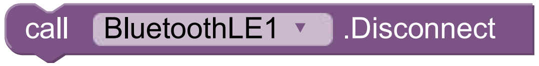
- `ConnectWithAddress` – Use the `ConnectWithAddress` method to connect to a specific Bluetooth low energy device if its Media Access Control (MAC) address is known. If none of the devices in the device list matches the given address, the `ConnectionFailed` event will be run. Otherwise, if a connection is successful the `Connected` event will be run.

  **Parameters**:

  - `address` (*text*) — The MAC address of the target device, of the form "12:34:56:78:90:ab"



- `Disconnect` – Disconnects from the currently connected BluetoothLE device if a device is connected.



- `DisconnectWithAddress` – Disconnects from a connected BluetoothLE device with the given address.

  **Parameters**:

Home

Getting Started

Tutorials

How Tos

IoT in Action

Devices

BluetoothLE

Arduino 101

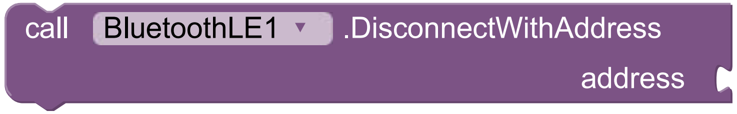
BBC micro:bit

Accelerometer

Button

IO Pin

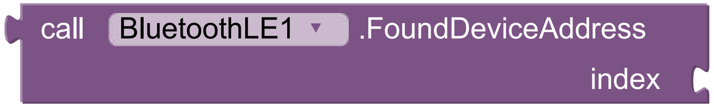
  - `address` (*text*) — The Media Access Control (MAC) address of the device to disconnect, of the form "12:34:56:78:90:ab"



- `FoundDeviceAddress` – Gets the Media Access Control (MAC) address of the found device at the given index in the device list. Index specifies the position in the BluetoothLE device list, starting from 1.

  **Parameter**:

  - `index` (*number*) — The index of the desired device, which must be between 1 and the length of the device list.



- `FoundDeviceName` – Gets the name of the found device at the given index in the device list.

  **Parameter**:

  - `index` (*number*) — The index of the desired device, which must be between 1 and the length of the device list.

Home

Getting Started

Tutorials

How Tos

IoT in Action

Devices

BluetoothLE

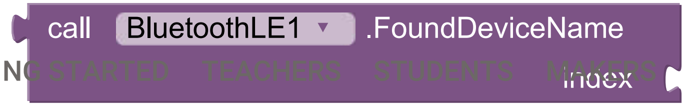
Arduino 101

BBC micro:bit

Accelerometer

Button

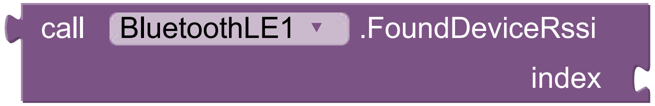
IO Pin



- `FoundDeviceRssi` – Gets the Received Signal Strength Indicator (RSSI) of the found device at the given index. The returned value will be between -100 and 0 indicating the strength of the connection.

  **Parameter**:

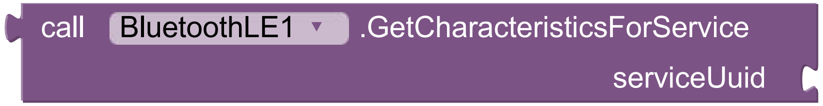
  - `index` (*number*) — The index of the desired device, which must be between 1 and the length of the device list.



- `GetCharacteristicsForService` – Returns the list of supported characteristics for the given service. The list will contain (UUID, name) pairs for each characteristic provided by the service UUID.

  **Parameters**:

  - `serviceUuid` (*text*) — The unique identifier of the service passed in the read or register call.



Home

Getting Started

Tutorials

How Tos

IoT in Action

Devices

BluetoothLE

Arduino 101

BBC micro:bit

Accelerometer

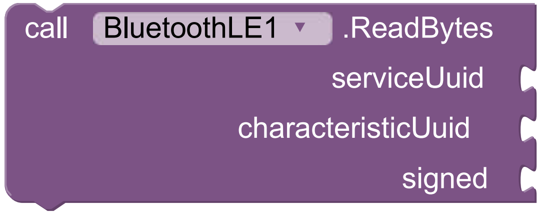
Button

IO Pin

- `ReadBytes` – Reads one or more 8-bit integer values from a connected BluetoothLE device. Service Unique ID and Characteristic UniqueID are required. The `signed` parameter indicates whether the bytes should be interpreted as signed values or not when being converted into App Inventor numbers. After the bytes are read, the `BytesReceived` event will be run.

  **Parameters:**

  - `serviceUuid` (*text*) — The unique identifier of the service passed in the read or register call.
  - `characteristicUuid` (*text*) — The unique identifier of the characteristic in the read or register call.
  - `signed` (*boolean*) Interpret the bytes as signed (true) or unsigned (false).



- `ReadFloats` – Reads one or more IEEE 754 floating point numbers from a connected BluetoothLE device. Service Unique ID and Characteristic UniqueID are required. The `shortFloat` parameter indicates whether the floats are either 16-bit half-precision floating point or 32-bit single precision floating point numbers. After the floats are read, the `FloatsReceived` event will be run.

  **Parameters:**

Home

Getting Started

Tutorials

How Tos

IoT in Action

Devices

BluetoothLE

Arduino 101

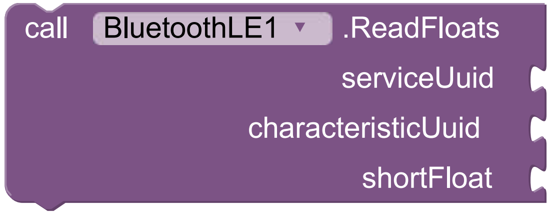
BBC micro:bit

Accelerometer

Button

IO Pin

- `serviceUuid` (*text*) — The unique identifier of the service passed in the read or register call.
- `characteristicUuid` (*text*) — The unique identifier of the characteristic in the read or register call.
- `shortFloat` (*boolean*) Interpret the floats as 16-bit half-precision (true) or 32-bit single-precision (false).

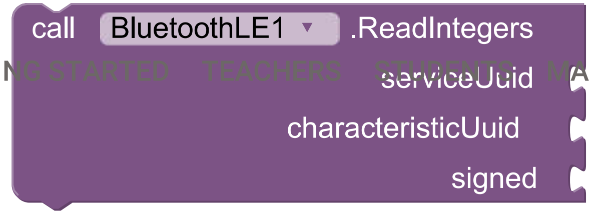


- `ReadIntegers` – Reads one or more 32-bit integer values from a connected BluetoothLE device. Service Unique ID and Characteristic UniqueID are required. The `signed` parameter indicates whether the integers should be interpreted as signed values or not when being converted into App Inventor numbers. After the integers are read, the `IntegersReceived` event will be run.

  **Parameters**:

  - `serviceUuid` (*text*) — The unique identifier of the service passed in the read or register call.
  - `characteristicUuid` (*text*) — The unique identifier of the characteristic in the read or register call.
  - `signed` (*boolean*) Interpret the integers as signed (true) or unsigned (false).

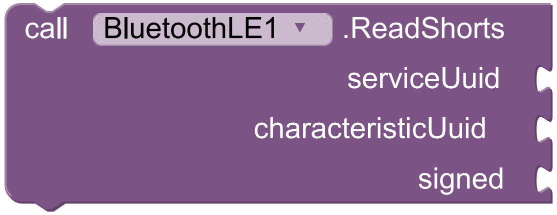
Home
Getting Started
Tutorials
How Tos
IoT in Action
Devices
BluetoothLE
Arduino 101
BBC micro:bit
Accelerometer
Button
IO Pin



- `ReadShorts` – Reads one or more 16-bit integer values from a connected BluetoothLE device. Service Unique ID and Characteristic UniqueID are required. The `signed` parameter indicates whether the shorts should be interpreted as signed values or not when being converted into App Inventor numbers. After the shorts are read, the `ShortsReceived` event will be run.

  **Parameters**:

  - `serviceUuid` (*text*) — The unique identifier of the service passed in the read or register call.
  - `characteristicUuid` (*text*) — The unique identifier of the characteristic in the read or register call.
  - `signed` (*boolean*) Interpret the shorts as signed (true) or unsigned (false).



- `ReadStrings` – Reads one or more null-terminated strings from a connected BluetoothLE device. Service Unique ID and Characteristic Unique ID are required. The

Home

Getting Started

Tutorials

How Tos

IoT in Action

Devices

BluetoothLE

Arduino 101

BBC micro:bit

Accelerometer

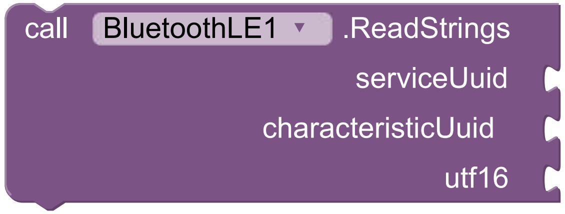
Button

IO Pin

`utf16` parameter indicates whether the content should be decoded as UTF-16 (true) or UTF-8 (false) code points when converting to App Inventor strings. After the strings are read, the `StringsReceived` event will be run.

**Parameters**:

- `serviceUuid` (*text*) — The unique identifier of the service passed in the read or register call.
- `characteristicUuid` (*text*) — The unique identifier of the characteristic in the read or register call.
- `utf16` (*boolean*) Interpret the string content as UTF-16 (true) or UTF-8 (false) code points.



- `RegisterForBytes` – Registers to receive updates when one or more 8-bit integer values from a connected BluetoothLE device are changed. Service Unique ID and Characteristic Unique ID are required. The `signed` parameter indicates whether the bytes should be interpreted as signed values or not when being converted into App Inventor numbers. Whenever a change is received, the `BytesReceived` event will be run.

**Parameters**:

Home

Getting Started

Tutorials

How Tos

IoT in Action

Devices

BluetoothLE

Arduino 101

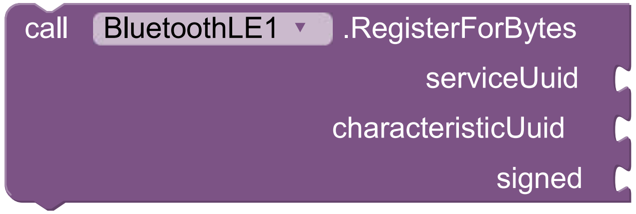
BBC micro:bit

Accelerometer

Button

IO Pin

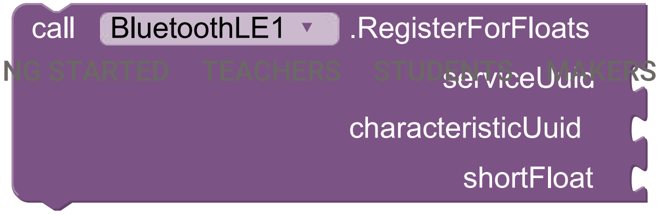
- `serviceUuid` (*text*) — The unique identifier of the service passed in the read or register call.
- `characteristicUuid` (*text*) — The unique identifier of the characteristic in the read or register call.
- `signed` (*boolean*) Interpret the bytes as signed (true) or unsigned (false).



- `RegisterForFloats` – Registers to receive updates when one or more IEEE 754 floating point numbers from a connected BluetoothLE device are changed. Service Unique ID and Characteristic Unique ID are required. The `shortFloat` parameter indicates whether the floats are either 16-bit half-precision floating point or 32-bit single precision floating point numbers. Whenever a change is received, the `FloatsReceived` event will be run.

  **Parameters**:

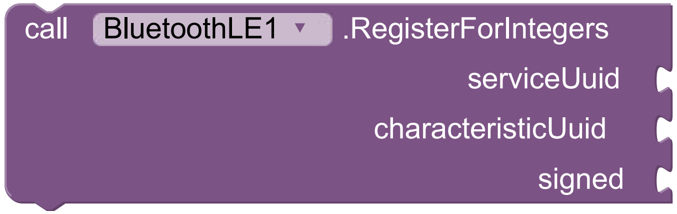
  - `serviceUuid` (*text*) — The unique identifier of the service passed in the read or register call.
  - `characteristicUuid` (*text*) — The unique identifier of the characteristic in the read or register call.
  - `shortFloat` (*boolean*) Interpret the floats as 16-bit half-precision (true) or 32-bit single-precision (false).



- `RegisterForIntegers` – Registers to receive updates when one or more 32-bit integer values from a connected BluetoothLE device are changed. Service Unique ID and Characteristic Unique ID are required. The `signed` parameter indicates whether the integers should be interpreted as signed values or not when being converted into App Inventor numbers. Whenever a change is received, the `IntegersReceived` event will be run.

  **Parameters**:

  - `serviceUuid` (*text*) — The unique identifier of the service passed in the read or register call.
  - `characteristicUuid` (*text*) — The unique identifier of the characteristic in the read or register call.
  - `signed` (*boolean*) Interpret the integers as signed (true) or unsigned (false).



Home

Getting Started

Tutorials

How Tos

IoT in Action

Devices

BluetoothLE

Arduino 101

BBC micro:bit

Accelerometer

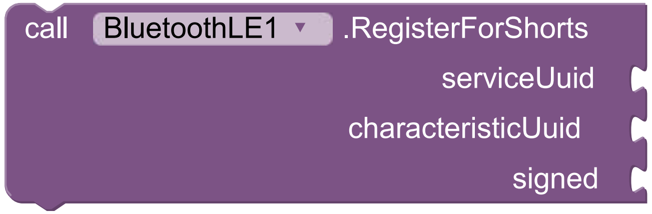
Button

IO Pin

- `RegisterForShorts` – Registers to receive updates when one or more 16-bit integer values from a connected BluetoothLE device are changed. Service Unique ID and Characteristic Unique ID are required. The `signed` parameter indicates whether the shorts should be interpreted as signed values or not when being converted into App Inventor numbers. Whenever a change is received, the `ShortsReceived` event will be run.

  **Parameters**:

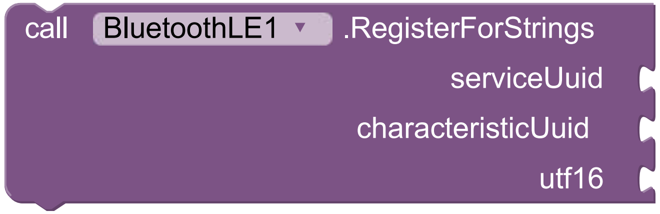
  - `serviceUuid` (*text*) — The unique identifier of the service passed in the read or register call.
  - `characteristicUuid` (*text*) — The unique identifier of the characteristic in the read or register call.
  - `signed` (*boolean*) Interpret the shorts as signed (true) or unsigned (false).



- `RegisterForStrings` – Registers to receive updates when one or more null-terminated strings from a connected BluetoothLE device are changed. Service Unique ID and Characteristic Unique ID are required. The `utf16` parameter indicates whether the content should be decoded as UTF-16 (true) or UTF-8 (false) code points when converting to App Inventor strings. Whenever a change is received, the `StringsReceived` event will be run.

**Parameters**:

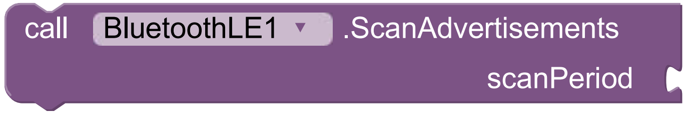
- `serviceUuid` (*text*) — The unique identifier of the service passed in the read or register call.
- `characteristicUuid` (*text*) — The unique identifier of the characteristic in the read or register call.
- `utf16` (*boolean*) Interpret the string content as UTF-16 (true) or UTF-8 (false) code points.



- `ScanAdvertisements` – Scans for advertising Bluetooth low energy devices.

**Parameter**:

- `scanPeriod` (*number*) — The amount of time to spend scanning, in milliseconds.



- `ServiceByIndex` – Returns the Unique ID of the service at the given index in the service list.

**Parameters**:

Home

Getting Started

Tutorials

How Tos

IoT in Action

Devices

BluetoothLE

Arduino 101

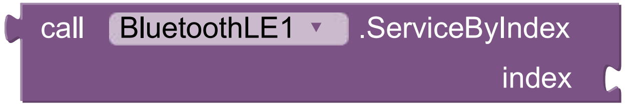
BBC micro:bit

Accelerometer

Button

IO Pin

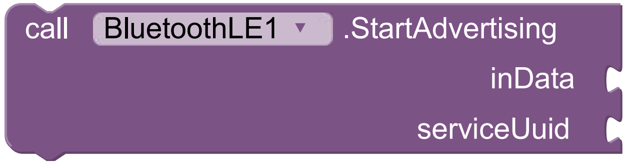
  - `index` (*number*) — The index of the desired service, which must be between 1 and the length of the service list.



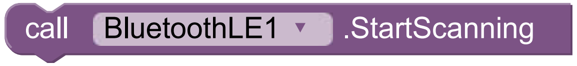
- `StartAdvertising` – Creates and publishes a Bluetooth low energy advertisement.

  **Parameters**:

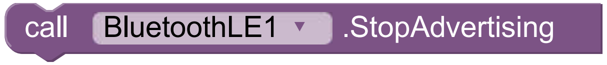
  - `inData` (*text*) — The data to be included in the service advertisement.
  - `serviceUuid` (*text*) — The unique identifier of the service passed in the read or register call.



- `StartScanning` – Starts scanning for Bluetooth low energy devices.



- `StopAdvertising` – Stops Bluetooth low energy advertisement from a previous call to `StartAdvertising`.



Home

Getting Started

Tutorials

How Tos

IoT in Action

Devices

BluetoothLE

Arduino 101

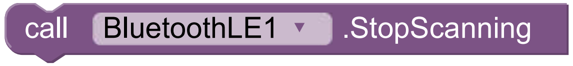
BBC micro:bit

Accelerometer

Button

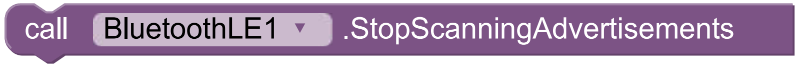
IO Pin

- `StopScanning` – Stops scanning for Bluetooth low energy devices.



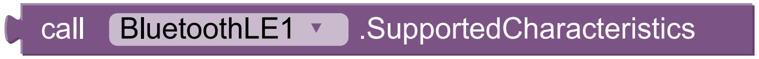
call BluetoothLE1 .StopScanning

- `StopScanningAdvertisements` – Stops scanning for Bluetooth low energy advertisements.



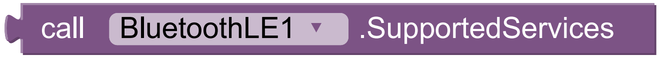
call BluetoothLE1 .StopScanningAdvertisements

- `SupportedCharacteristics` – Returns a list of supported characteristic for the connected device as a string.



call BluetoothLE1 .SupportedCharacteristics

- `SupportedServices` – Returns the list of supported service for the connected device as a string.



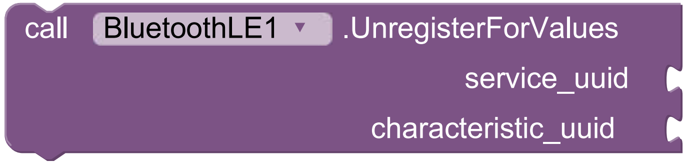
call BluetoothLE1 .SupportedServices

- `UnregisterForValues` – Unregisters for updates from the given service and characteristic.

  **Parameters**:

  - `serviceUuid` (*text*) — The unique identifier of the service passed in the read or register call.

  - `characteristicUuid` (*text*) — The unique identifier of the characteristic in the read or register call.



- `WriteBytes` – Writes one or more 8-bit integer values to a connected BluetoothLE device. Service Unique ID and Characteristic Unique ID are required. The values parameter can either be a single numeric value or a list of values. If `signed` is true, the acceptable values are between -128 and 127. If `signed` is false, the acceptable values are between 0 and 255.

  **Parameters**:

  - `serviceUuid` (*text*) — The unique identifier of the service passed in the read or register call.
  - `characteristicUuid` (*text*) — The unique identifier of the characteristic in the read or register call.
  - `signed` (*boolean*) Interpret the bytes as signed (true) or unsigned (false).
  - `values` (*list*) — A list of values to write to the device.

Internet of Things

Home

Getting Started

Tutorials

How Tos

IoT in Action

Devices

BluetoothLE

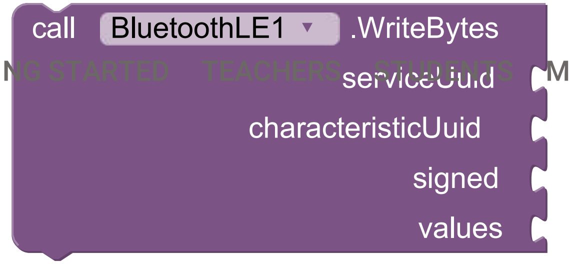
Arduino 101

BBC micro:bit

Accelerometer

Button

IO Pin

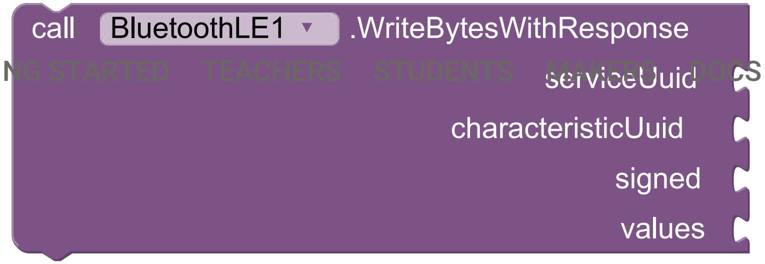


- `WriteBytesWithResponse` – Writes one or more 8-bit integer values to a connected BluetoothLE device and waits for an acknowledgement via the `BytesWritten` event. Service Unique ID and Characteristic Unique ID are required. The values parameter can either be a single numeric value or a list of values. If `signed` is true, the acceptable values are between -128 and 127. If `signed` is false, the acceptable values are between 0 and 255.

  **Parameters**:

  - `serviceUuid` (*text*) — The unique identifier of the service passed in the read or register call.
  - `characteristicUuid` (*text*) — The unique identifier of the characteristic in the read or register call.
  - `signed` (*boolean*) Interpret the bytes as signed (true) or unsigned (false).
  - `values` (*list*) — A list of values to write to the device.

Home
Getting Started
Tutorials
How Tos
IoT in Action
Devices
BluetoothLE
Arduino 101
BBC micro:bit
Accelerometer
Button
IO Pin

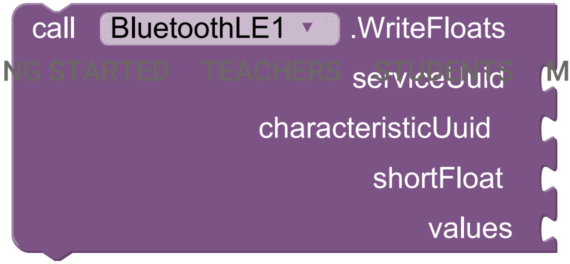


- `WriteFloats` – Writes one or more IEEE 754 floating point numbers to a connected BluetoothLE device. Service Unique ID and Characteristic Unique ID are required. The values parameter can either be a single numeric value or a list of values. If `shortFloat` is true, then each numeric value will be compressed to fit into a 16-bit half-precision floating point value. If `shortFloat` is false, then each numeric value will be sent as a 32-bit single precision floating point value.

  **Parameters**:

  - `serviceUuid` (*text*) — The unique identifier of the service passed in the read or register call.
  - `characteristicUuid` (*text*) — The unique identifier of the characteristic in the read or register call.
  - `shortFloat` (*boolean*) Interpret the floats as 16-bit half-precision (true) or 32-bit single-precision (false).
  - `values` (*list*) — A list of values to write to the device.

Home
Getting Started
Tutorials
How Tos
IoT in Action
Devices
BluetoothLE
Arduino 101
BBC micro:bit
Accelerometer
Button
IO Pin



- `WriteFloatsWithResponse` – Writes one or more IEEE 754 floating point values to a connected BluetoothLE device and waits for an acknowledgement via the `FloatsWritten` event. Service Unique ID and Characteristic Unique ID are required. The values parameter can either be a single numeric value or a list of values. If `shortFloat` is false, then each numeric value will be sent as a 32-bit single precision floating point value.

  **Parameters**:

  - `serviceUuid` (*text*) — The unique identifier of the service passed in the read or register call.
  - `characteristicUuid` (*text*) — The unique identifier of the characteristic in the read or register call.
  - `shortFloat` (*boolean*) Interpret the floats as 16-bit half-precision (true) or 32-bit single-precision (false).
  - `values` (*list*) — A list of values to write to the device.

Home

Getting Started

Tutorials

How Tos

IoT in Action

Devices

BluetoothLE

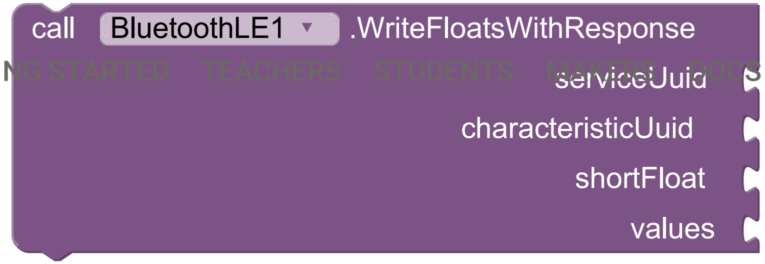
Arduino 101

BBC micro:bit

Accelerometer

Button

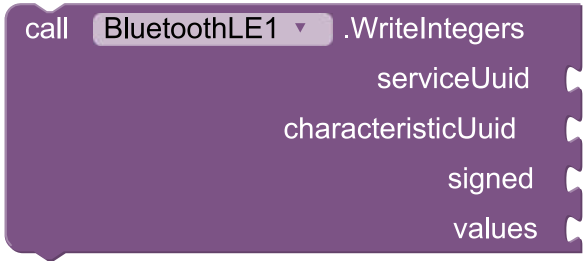
IO Pin



- `WriteIntegers` – Writes one or more 32-bit integer values to a connected BluetoothLE device. Service Unique ID and Characteristic Unique ID are required. The values parameter can either be a single numeric value or a list of values. If `signed` is true, the acceptable values are between -2147483648 and 2147483647. If `signed` is false, the acceptable values are between 0 and 4294967295.

  **Parameters**:

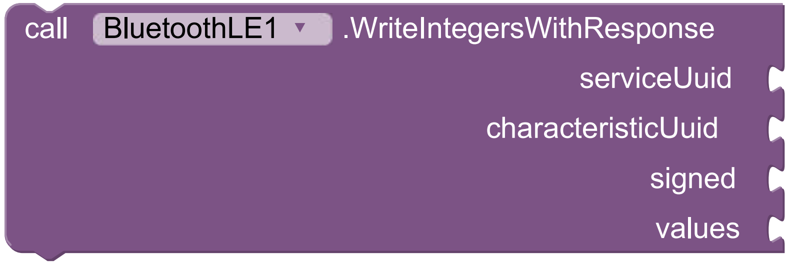
  - `serviceUuid` (*text*) — The unique identifier of the service passed in the read or register call.
  - `characteristicUuid` (*text*) — The unique identifier of the characteristic in the read or register call.
  - `signed` (*boolean*) Interpret the integers as signed (true) or unsigned (false).
  - `values` (*list*) — A list of values to write to the device.



- `WriteIntegersWithResponse` – Writes one or more 32-bit integer values to a connected BluetoothLE device and waits for an acknowledgement via the `IntegersWritten` event. Service Unique ID and Characteristic Unique ID are required. The values parameter can either be a single numeric value or a list of values. If `signed` is true, the acceptable values are between -2147483648 and 2147483647. If `signed` is false, the acceptable values are between 0 and 4294967295.

  **Parameters**:

  - `serviceUuid` (*text*) — The unique identifier of the service passed in the read or register call.
  - `characteristicUuid` (*text*) — The unique identifier of the characteristic in the read or register call.
  - `signed` (*boolean*) Interpret the integers as signed (true) or unsigned (false).
  - `values` (*list*) — A list of values to write to the device.



- `WriteShorts` – Writes one or more 16-bit integer values to a connected BluetoothLE device. Service Unique ID and Characteristic Unique ID are required. The values parameter can either be a single numeric value or a list of values. If `signed` is true, the

Home

Getting Started

Tutorials

How Tos

IoT in Action

Devices

BluetoothLE

Arduino 101

BBC micro:bit

Accelerometer

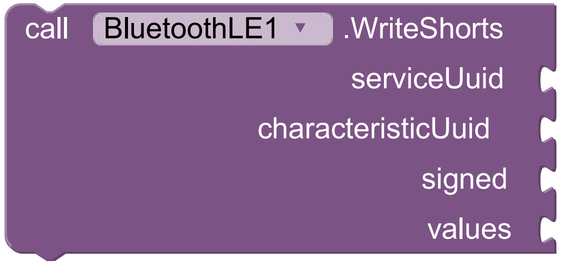
Button

IO Pin

acceptable values are between -32768 and 32767. If `signed` is false, the acceptable values are between 0 and 65535.

**Parameters**:

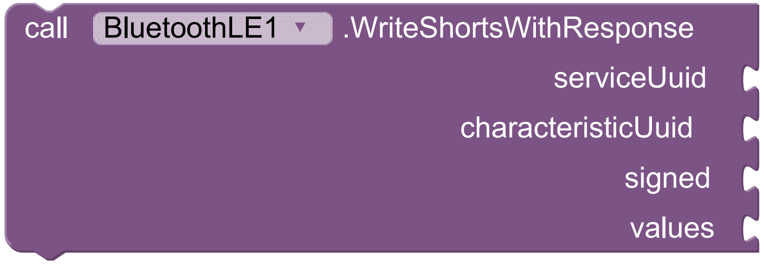
- `serviceUuid` (*text*) — The unique identifier of the service passed in the read or register call.
- `characteristicUuid` (*text*) — The unique identifier of the characteristic in the read or register call.
- `signed` (*boolean*) Interpret the shorts as signed (true) or unsigned (false).
- `values` (*list*) — A list of values to write to the device.



- `WriteShortsWithResponse` – Writes one or more 16-bit integer values to a connected BluetoothLE device and waits for an acknowledgement via the `ShortsWritten` event. Service Unique ID and Characteristic Unique ID are required. The values parameter can either be a single numeric value or a list of values. If `signed` is true, the acceptable values are between -32768 and 32767. If `signed` is false, the acceptable values are between 0 and 65535.

**Parameters**:

- `serviceUuid` (*text*) — The unique identifier of the service passed in the read or register call.
- `characteristicUuid` (*text*) — The unique identifier of the characteristic in the read or register call.
- `signed` (*boolean*) Interpret the shorts as signed (true) or unsigned (false).
- `values` (*list*) — A list of values to write to the device.

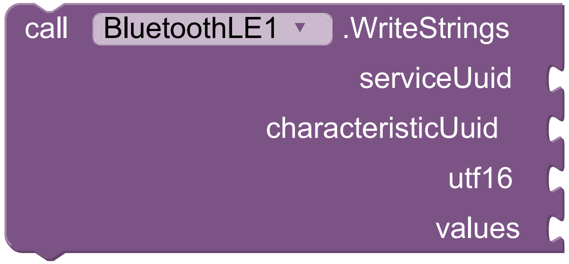


- `WriteStrings` – Writes one or more strings to a connected BluetoothLE device. Service Unique ID and Characteristic Unique ID are required. The values parameter can either be a single string or a list of strings. If `utf16` is true, the string(s) will be sent using UTF-16 little endian encoding. If `utf16` is false, the string(s) will be sent using UTF-8 encoding.

  **Parameters**:

  - `serviceUuid` (*text*) — The unique identifier of the service passed in the read or register call.
  - `characteristicUuid` (*text*) — The unique identifier of the characteristic in the read or register call.

- `utf16` (*boolean*) Send the string encoded as UTF-16 little endian (true) or UTF-8 (false) code points.
- `values` (*list*) — A list of values to write to the device.



- `WriteStringsWithResponse` – Writes one or more strings to a connected BluetoothLE device and waits for an acknowledgement via the `StringsWritten` event. Service Unique ID and Characteristic Unique ID are required. The values parameter can either be a single string or a list of values. If `utf16` is true, the string(s) will be sent using UTF-16 little endian encoding. If `utf16` is false, the string(s) will be sent using UTF-8 encoding.

  **Parameters**:

  - `serviceUuid` (*text*) — The unique identifier of the service passed in the read or register call.
  - `characteristicUuid` (*text*) — The unique identifier of the characteristic in the read or register call.
  - `utf16` (*boolean*) Send the string encoded as UTF-16 little endian (true) or UTF-8 (false) code points.
  - `values` (*list*) — A list of values to write to the device.

Home

Getting Started

Tutorials

How Tos

IoT in Action

Devices

BluetoothLE

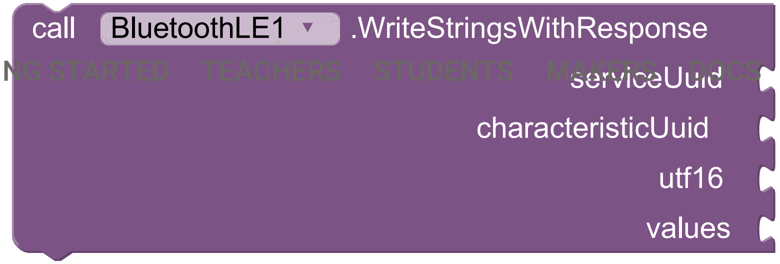
Arduino 101

BBC micro:bit

Accelerometer

Button

IO Pin

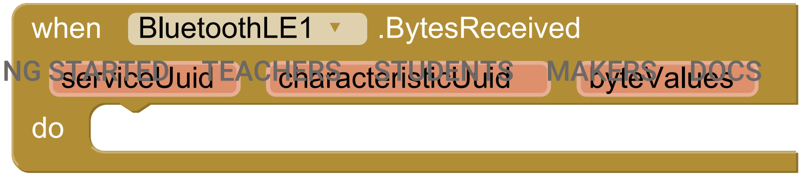


## Events

- `BytesReceived` – The `BytesReceived` event is run when one or more byte values are received from a connected Bluetooth device. Depending on the `sign` parameter of the last call to `ReadBytes` or `RegisterForBytes` for the given `serviceUuid` and `characteristicUuid`, the `byteValues` list will contain numbers ranging from -128 to 127 ( `signed = true` ) or 0 to 255 ( `signed = false` ).

  **Parameters**:

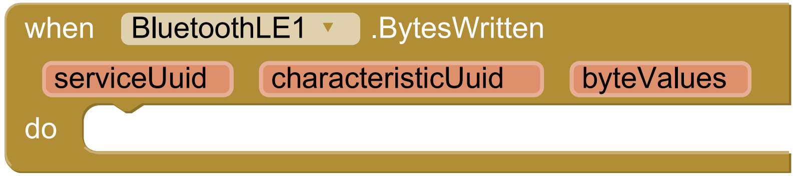
  - `serviceUuid` (*text*) — The unique identifier of the service passed in the read or register call.
  - `characteristicUuid` (*text*) — The unique identifier of the characteristic in the read or register call.
  - `byteValues` (*list*) — A list of values read from the device. The range of each value will depend on the `sign` flag previously specified in the call to read or register.



- `BytesWritten` – The `BytesWritten` event is run when one or more byte values are written to a connected Bluetooth device. `byteValues` will be a list of values actually written to the device. This may be different if the original input was too long to fit into a single transmission unit (typically 23 bytes).

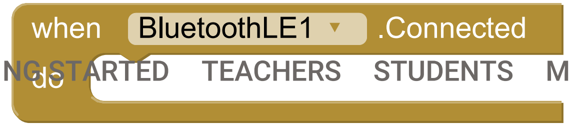
  **Parameters**:

  - `serviceUuid` (*text*) — The unique identifier of the service passed in the read or register call.
  - `characteristicUuid` (*text*) — The unique identifier of the characteristic in the read or register call.
  - `byteValues` (*list*) — A list of values written to the device.



- `Connected` – The `Connected` event is run after the application successfully connects to a Bluetooth low energy device. This can be the result of a call to `Connect` or `ConnectWithAddress`, or as a result of an automatic reconnect if the `AutoReconnect` property was true at the time a connection was requested.

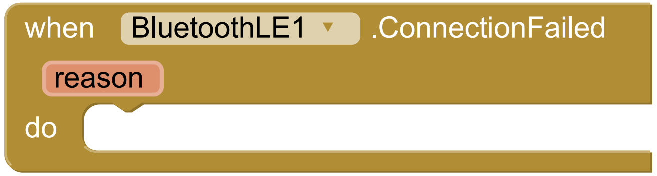
Home
Getting Started
Tutorials
How Tos
IoT in Action
Devices
BluetoothLE
Arduino 101
BBC micro:bit
Accelerometer
Button
IO Pin



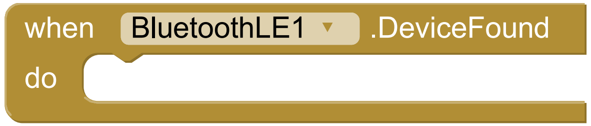
- `ConnectionFailed` – The `ConnectionFailed` event is run when an attempt to connect to a device does not succeed. If a reason is provided by the Bluetooth low energy stack it will be reported via the `reason` parameter.

  **Parameters**:

  - `reason` (*text*) — The reason the connection failed, if known.



- `DeviceFound` – The `DeviceFound` event is run when a new Bluetooth low energy device is found.



- `Disconnected` – The `Disconnected` event is run when a Bluetooth low energy device is disconnected. This can be caused by a call to `Disconnect` or `DisconnectWithAddress` , or after a device is moved away or reset such that a loss of connection occurs.

Home

Getting Started

Tutorials

How Tos

IoT in Action

Devices

BluetoothLE

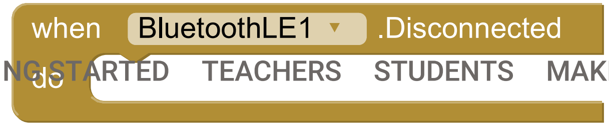
Arduino 101

BBC micro:bit

Accelerometer

Button

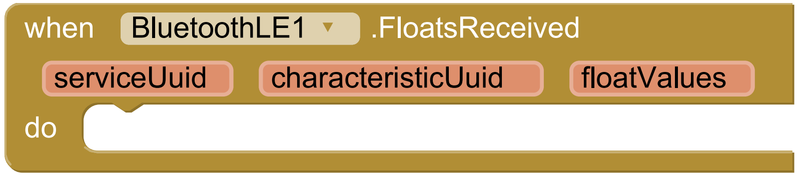
IO Pin



- `FloatsReceived` – The `FloatsReceived` event is run when one or more IEEE 754 floating point values are received from a connected Bluetooth device. Depending on the `shortFloat` parameter of the last call to `ReadFloats` or `RegisterForFloats` for the given `serviceUuid` and `characteristicUuid`, the `floatValues` list will contain numbers ranging from -65504.0 to 65504.0 ( `shortFloat = true` ) or -3.402823466E38 to 3.402823466E38 ( `shortFloat = false` ).

  **Parameters**:

  - `serviceUuid` (*text*) — The unique identifier of the service passed in the read or register call.
  - `characteristicUuid` (*text*) — The unique identifier of the characteristic in the read or register call.
  - `floatValues` (*list*) — A list of values read from the device. The range of each value will depend on the `sign` flag previously specified in the call to read or register.



Home

Getting Started

Tutorials

How Tos

IoT in Action

Devices

BluetoothLE

Arduino 101

BBC micro:bit

Accelerometer

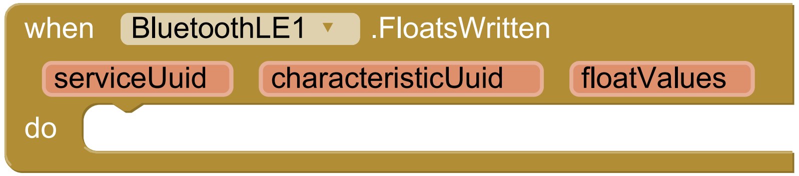
Button

IO Pin

- `FloatsWritten` – The `FloatsWritten` event is run when one or more IEEE 754 floating point values are written to a connected Bluetooth device. `floatValues` will be a list of values actually written to the device. This may be different if the original input was too long to fit into a single transmission unit (typically 11 short floats or 5 regular floats).

  **Parameters:**

  - `serviceUuid` (*text*) — The unique identifier of the service passed in the read or register call.
  - `characteristicUuid` (*text*) — The unique identifier of the characteristic in the read or register call.
  - `floatValues` (*list*) — A list of values written to the device.



- `IntegersReceived` – The `IntegersReceived` event is run when one or more 32-bit integer values are received from a connected Bluetooth device. Depending on the `sign` parameter of the last call to `ReadIntegers` or `RegisterForIntegers` for the given `serviceUuid` and `characteristicUuid`, the `intValues` list will contain numbers ranging from -2147483648 to 2147483647 ( `signed = true` ) or 0 to 4294967296 ( `signed = false` ).

  **Parameters:**

Home

Getting Started

Tutorials

How Tos

IoT in Action

Devices

BluetoothLE

Arduino 101

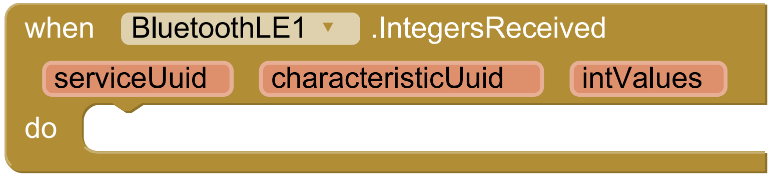
BBC micro:bit

Accelerometer

Button

IO Pin

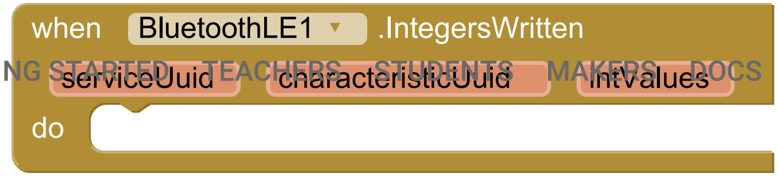
- `serviceUuid` (*text*) — The unique identifier of the service passed in the read or register call.
- `characteristicUuid` (*text*) — The unique identifier of the characteristic in the read or register call.
- `intValues` (*list*) — A list of values read from the device. The range of each value will depend on the `sign` flag previously specified in the call to read or register.



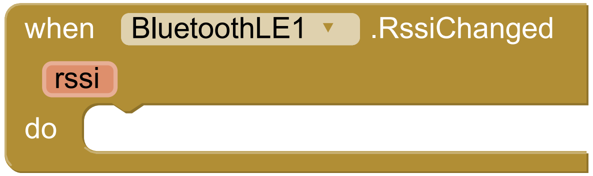
- `IntegersWritten` – The `IntegersWritten` event is run when one or more 32-bit integers values are written to a connected Bluetooth device. `intValues` will be a list of values actually written to the device. This may be different if the original input was too long to fit into a single transmission unit (typically 5 integers).

  **Parameters**:

  - `serviceUuid` (*text*) — The unique identifier of the service passed in the read or register call.
  - `characteristicUuid` (*text*) — The unique identifier of the characteristic in the read or register call.
  - `intValues` (*list*) — A list of values written to the device.



- `RssiChanged` – Trigger event when RSSI (Received Signal Strength Indicator) of found BluetoothLE device changes



- `ShortsReceived` – The `ShortsReceived` event is run when one or more short integer values are received from a connected Bluetooth device. Depending on the `sign` parameter of the last call to `ReadShorts` or `RegisterForShorts` for the given `serviceUuid` and `characteristicUuid`, the `shortValues` list will contain numbers ranging from -32768 to 32767 ( `signed = true` ) or 0 to 65535 ( `signed = false` ).

  **Parameters**:

  - `serviceUuid` (*text*) — The unique identifier of the service passed in the read or register call.
  - `characteristicUuid` (*text*) — The unique identifier of the characteristic in the read or register call.
  - `shortValues` (*list*) — A list of values read from the device. The range of each value will depend on the `sign` flag previously specified in the call to read or register.

Home

Getting Started

Tutorials

How Tos

IoT in Action

Devices

BluetoothLE

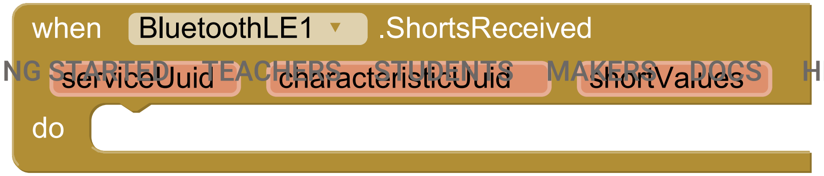
Arduino 101

BBC micro:bit

Accelerometer

Button

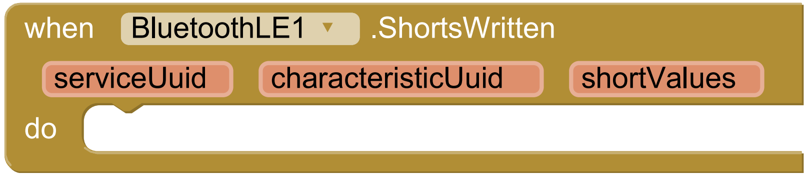
IO Pin



- `ShortsWritten` – The `ShortsWritten` event is run when one or more short integers values are written to a connected Bluetooth device. `shortValues` will be a list of values actually written to the device. This may be different if the original input was too long to fit into a single transmission unit (typically 11 shorts).

  **Parameters**:

  - `serviceUuid` (*text*) — The unique identifier of the service passed in the read or register call.
  - `characteristicUuid` (*text*) — The unique identifier of the characteristic in the read or register call.
  - `shortValues` (*list*) — A list of values written to the device.



- `StringsReceived` – The `StringsReceived` event is run when one or more strings are received from a connected Bluetooth device. Depending on the `utf16` parameter of the last call to `ReadStrings` or `RegisterForStrings` for the given `serviceUuid` and `characteristicUuid`, the `stringValues` list will contain either a UTF-16 little endian decoded ( `utf16 = true` ) or UTF-8 decoded ( `utf16 = false` ) strings. The string length

Home

Getting Started

Tutorials

How Tos

IoT in Action

Devices

BluetoothLE

Arduino 101

BBC micro:bit

Accelerometer

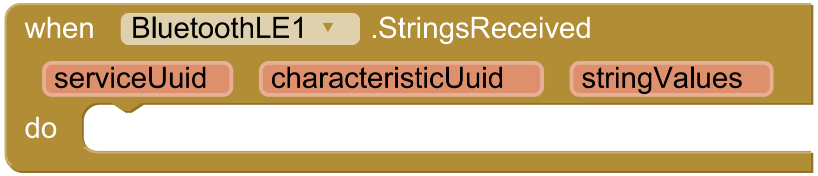
Button

IO Pin

is limited by the maximum transmission unit (MTU) of the Bluetooth device, which is typically 23 bytes.

**Parameters**:

- `serviceUuid` (*text*) — The unique identifier of the service passed in the read or register call.
- `characteristicUuid` (*text*) — The unique identifier of the characteristic in the read or register call.
- `stringValues` (*list*) — A list of values read from the device. The strings will be decoded as UTF-16 or UTF-8 based on the `utf16` flag previously specified in the call to read or register.



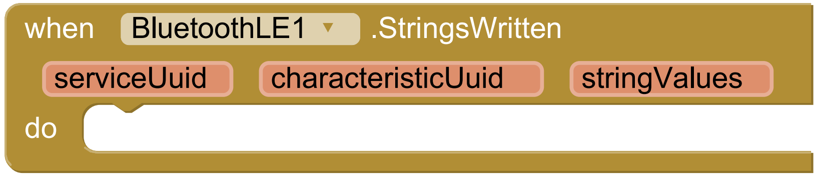
- `StringsWritten` – The `StringsWritten` event is run when one or more strings are written to a connected Bluetooth device. `stringValues` will be a list of values actually written to the device. This may be different if the original input was too long to fit into a single transmission unit (typically 22 bytes).

**Parameters**:

- `serviceUuid` (*text*) — The unique identifier of the service passed in the read or register call.

Home
Getting Started
Tutorials
How Tos
IoT in Action
Devices
BluetoothLE
Arduino 101
BBC micro:bit
Accelerometer
Button
IO Pin

- `characteristicUuid` (*text*) — The unique identifier of the characteristic in the read or register call.
- `stringValues` (*list*) — A list of values written to the device.



Contact Us