# App Inventor 2 Classic Bluetooth Extension

**DESCRIPTION**

This document describes the use of an App Inventor 2 (AI2) extension to be able to communicate via Classic Bluetooth an App developed with AI2 with any device. Associated with this extension, an Arduino code is included to be able to encode and decode the Bluetooth telegrams (see Protocol section).

The extension includes AI2 components to access different types of generic electronics such as digital, analog, and PWM inputs and outputs, but also for fairly common Arduino devices such as the sound buzzer, temperature and humidity sensors, infrared, etc… each These devices have an AI2 component with specific functionalities.

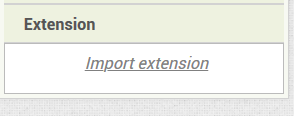
In addition, there are also components for access to variables, not necessarily associated with a certain type of electronics, on which we can share information between the App and the device. These variables can be of the type boolean (bool), integer (int), decimal (float) and text strings (String).

**DOWNLOAD AND IMPORT**

**The latest version of the extension can be downloaded from:**

<https://roboticafacil.es/facilino/ai2/es.roboticafacil.facilino.runtime.bluetooth.aix>

**Once downloaded, you can import the extension to your project from the designer view:**



**USER MANUAL**

**Components**

The extension has the following components, which are classified according to the colour code indicated in the following figure. To add the components, drag them to the interface in the designer view and they will appear as non-visible components.

The main component is the “FacilinoBluetoothClient” component that allows communication with the Bluetooth device. This component allows you to connect / disconnect with the device, as well as different utilities to reconnect with the last device and know if it is paired. It has a timer that allows the regularity with which the availability of new data is checked (reception).

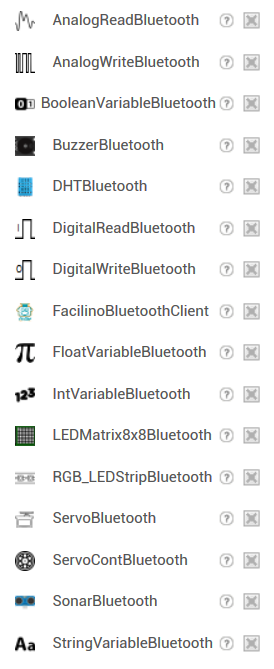
On the other hand, the components of digital, analog signals, sensors and actuators and variables all work in a very similar way. Those components that read information have the “Request” and “Update” methods, whose functionality is similar, unlike “Request” is non-blocking, while “Update” is blocking. After making a call to any of the methods, the component will report with the information received through the “Received” event. In addition, if for any reason the information is not received, this will be notified through the "Timeout" event.

Bluetooth Client

Analog/Digital signals

Sensors/Actuators

Variables



**FacilinoBluetoothClient**

**Properties**

**AdressesAndNames *list read-only***

**List with addresses and names of paired Bluetooth devices.**

**Available *boolean read-only***

**Returns true if Bluetooth is available on the device.**

**Enabled *boolean read-only***

**Returns true if Bluetooth is enabled on the device.**

**Connected *boolean read-only***

**Returns true if connected to a Bluetooth device.**

**LastConnectedAddress *text read-only***

**Gets the address and name of the last connected (and saved) device.**

**TimerEnabled *boolean***

**Gets or sets the timer to notify the reception of new data.**

**TimerInterval *number***

**Gets or sets the interval in milliseconds of the timer to notify the reception of new data.**

**FacilinoBluetoothClient *component***

**A FacilinoBluetoothClient component.**

**Methods**

***boolean* Connect (*text* address)**

**To connect to a device, we will use the “Connect” method that requires us to pass it a MAC address of the bluetooth device provided by the “AddressesAndNames” property. Returns true or false depending on whether the connection was established.**

**Disconnect**

**Disconnect the current connection.**

**ForgetLastConnection**

**Allows you to forget the last connection established (and saved).**

***boolean* IsDevicePaired (*text* address)**

**Returns true if the device indicated by the MAC address is paired. We can only connect to paired devices before trying to reconnect. Therefore, we should check that the last address with which the connection was established is still paired.**

**Reconnect**

**Allows to reconnect with the last connection established (and saved).**

**SaveConnection**

**Saves address and name of the last connection established, to be able to reconnect easily.**

**Events**

**TelegramError Notifies that an error has been generated in the telegram.**

**error *text* Telegram error message.**

**TelegramReceived Notifica que se ha recibido un telegrama.**

**cmd *number* Command of the received telegram.**

**length *number* Number of bytes received in the data.**

**data *list*  List with telegram data.**

**DigitalReadBluetooth**

**Properties**

**FacilinoBluetoothClient *component***

**Gets or sets the component that manages Bluetooth communication. Must be a component of type “FacilinoBluetoothClient”.**

**Pin *number***

**Gets or sets the digital input pin number.**

**UpdateTimeout *number***

**Gets or sets the maximum time that must elapse to consider that an error has occurred in the reception of the telegram.**

**Value *boolean***

**Gets the value of the digital input value (last received value).**

**DigitalReadBluetooth *component***

**The DigitalReadBluetooth component.**

**Methods**

**Request**

**Request the reading of the digital input with the indicated pin number (sends a reading telegram with the pin number). This feature is non-blocking and the reception of the telegram is expected to be notified with the response with “Received” or “Changed” events.**

**Update**

**Request the reading of the pin number (sends a read telegram with the pin number and waits to receive the telegram with the information of the variable with the same pin number). This is a blocking method. “Timeout” property indicates the maximum time this function is blocking before “Timeout” event is generated.**

**Events**

**Changed**

**Notifies that the digital input value has changed (a telegram has been received indicating that change).**

**value *boolean***

**Value of the digital input.**

**Received**

**Notifies that a telegram has been received with information on the value of the digital input.**

**value *boolean***

**Value of the digital input.**

**Timeout**

**Notifies that a read timer overflow has occurred.**

**error *text***

**Text with the timeout information.**

**DigitalWriteBluetooth**

**Properties**

**FacilinoBluetoothClient *component***

**Gets or sets the component that manages Bluetooth communication. Must be a component of type “FacilinoBluetoothClient”.**

**Pin *number***

**Gets or sets the digital output pin number.**

**Value *boolean***

**Gets the value of the last value set.**

**DigitalWriteBluetooth *component***

**The DigitalWriteBluetooth component.**

**Methods**

**Set (*boolean* value)**

**Sets the value of the digital output (sends a write telegram with the pin number).**

**Toggle**

**Toggles the value of the digital output (sends a write telegram with the opposite value to the last value stored with the pin number).**

**Events**

**None**

**AnalogReadBluetooth**

**Properties**

**FacilinoBluetoothClient *component***

**Gets or sets the component that manages Bluetooth communication. Must be a component of type “FacilinoBluetoothClient”.**

**Pin *number***

**Gets or sets the analog input pin number.**

**UpdateTimeout *number***

**Gets or sets the maximum time that must elapse to consider that an error has occurred in the reception of the telegram.**

**Value *number***

**Gets the value of the analog input value (last received value).**

**AnalogReadBluetooth *component***

**The AnalogReadBluetooth component.**

**Methods**

**Request**

**Request the reading of the analog input with the indicated analog pin number (sends a reading telegram with the analog pin number). This feature is non-blocking and the reception of the telegram is expected to be notified with the response with “Received” event.**

**Update**

**Request the reading of the analog pin number (sends a read telegram with the pin number and waits to receive the telegram with the information of the variable with the same pin number). This is a blocking method. “Timeout” property indicates the maximum time this function is blocking before “Timeout” event is generated.**

**Events**

**Received**

**Notifies that a telegram has been received with information on the value of the analog input.**

**value *number***

**Value of the analog input.**

**Timeout**

**Notifies that a read timer overflow has occurred.**

**error *text***

**Text with the timeout information.**

**AnalogWriteBluetooth**

**Properties**

**FacilinoBluetoothClient *component***

**Gets or sets the component that manages Bluetooth communication. Must be a component of type “FacilinoBluetoothClient”.**

**Pin *number***

**Gets or sets the PWM digital output pin number.**

**Value *number***

**Gets the value of the last value set.**

**DigitalWriteBluetooth *component***

**The AnalogWriteBluetooth component.**

**Methods**

**Set (*number* value)**

**Sets the value of the PWM digital output (sends a write telegram with the pin number). A value between 0 and 65535 (16 bit number).**

**Events**

**None**

**BooleanVariableBluetooth**

**Properties**

**FacilinoBluetoothClient *component***

**Gets or sets the component that manages Bluetooth communication. Must be a component of type “FacilinoBluetoothClient”.**

**Index *number***

**Gets or sets the index of the boolean variable position within the list of boolean variables that the device with which we communicate will maintain.**

**UpdateTimeout *number***

**Gets or sets the maximum time that must elapse to consider that an error has occurred in the reception of the telegram.**

**Value *boolean***

**Gets the value of the boolean variable (last received value).**

**BooleanVariableBluetooth *component***

**The BooleanVariableBluetooth component.**

**Methods**

**Request**

**Request the reading of the boolean variable with the indicated index (sends a reading telegram with the variable index). This feature is non-blocking and the reception of the telegram is expected to be notified with the response with “Received” or “Changed” events.**

**Set (*boolean* value)**

**Sets the value of the boolean variable (sends a write telegram with the variable index).**

**Toggle**

**Toggles the value of the Boolean variable (sends a write telegram with the opposite value to the last value stored with the variable index).**

**Update**

**Request the reading of the boolean variable (sends a read telegram with the variable index and waits to receive the telegram with the information of the variable with the same index). This is a blocking method. “Timeout” property indicates the maximum time this function is blocking before “Timeout” event is generated.**

**Events**

**Changed**

**Notifies that the boolean variable has changed (a telegram has been received indicating that change).**

**value *boolean***

**Value of the boolean variable.**

**Received**

**Notifies that a telegram has been received with information on the value of the boolean variable.**

**value *boolean***

**Value of the boolean variable.**

**Timeout**

**Notifies that a read timer overflow has occurred.**

**error *text***

**Text with the timeout information.**

**IntVariableBluetooth**

**Properties**

**FacilinoBluetoothClient *component***

**Gets or sets the component that manages Bluetooth communication. Must be a component of type “FacilinoBluetoothClient”.**

**Index *number***

**Gets or sets the index of the integer variable position within the list of integer variables that the device with which we communicate will maintain.**

**UpdateTimeout *number***

**Gets or sets the maximum time that must elapse to consider that an error has occurred in the reception of the telegram.**

**Value *number***

**Gets the value of the integer variable (last received value). The number is between 0 and 65535 (a 16-bit number).**

**IntVariableBluetooth *component***

**The IntVariableBluetooth component.**

**Methods**

**Request**

**Request the reading of the integer variable with the indicated index (send a reading telegram with the variable index). This feature is non-blocking and is expected to be notified the reception of the telegram with the response with “Received” event.**

**Set (*number* value)**

**Sets the value of the integer variable (sends a write telegram with the variable index). The value must be between 0 and 65535 (16-bit number).**

**Update**

**Request the reading of the integer variable (sends a read telegram with the variable index and waits to receive the telegram with the information of the variable with the same index). This is a blocking method. “Timeout” property indicates the maximum time this function is blocking before “Timeout” event is generated.**

**Events**

**Received**

**Notifies that a telegram has been received with information on the value of the integer variable.**

**value *number***

**Value with the integer variable. A number between 0 and 65535 (a 16-bit number).**

**Timeout**

**Notifies that a read timer overflow has occurred.**

**error *text***

**Text with the timeout information.**

**FloatVariableBluetooth**

**Properties**

**FacilinoBluetoothClient *component***

**Gets or sets the component that manages Bluetooth communication. Must be a component of type “FacilinoBluetoothClient”.**

**Index *number***

**Gets or sets the index of the float variable position within the list of float variables that the device with which we communicate will maintain.**

**UpdateTimeout *number***

**Gets or sets the maximum time that must elapse to consider that an error has occurred in the reception of the telegram.**

**Value *number***

**Gets the value of the float variable (last received value). The number is 32-bit precision.**

**FloatVariableBluetooth *component***

**The FloatVariableBluetooth component.**

**Methods**

**Request**

**Request the reading of the float variable with the indicated index (send a reading telegram with the variable index). This feature is non-blocking and the reception of the telegram is expected to be notified with the response with “Received” event.**

**Set (*number* value)**

**Sets the value of the float variable (sends a write telegram with the variable index). The number sent is 32-bit precision.**

**Update**

**Request the reading of the float variable (sends a read telegram with the variable index and waits to receive the telegram with the information of the variable with the same index). This is a blocking method. “Timeout” property indicates the maximum time this function is blocking before “Timeout” event is generated.**

**Events**

**Received**

**Notifies that a telegram has been received with information on the value of the float variable.**

**value *number***

**Value with the float variable. The number received is 32-bit precision.**

**Timeout**

**Notifies that a read timer overflow has occurred.**

**error *text***

**Text with the timeout information.**

**StringVariableBluetooth**

**Properties**

**FacilinoBluetoothClient *component***

**Gets or sets the component that manages Bluetooth communication. Must be a component of type “FacilinoBluetoothClient”.**

**Index *number***

**Gets or sets the index of the String variable position within the list of String variables that the device with which we communicate will maintain.**

**UpdateTimeout *number***

**Gets or sets the maximum time that must elapse to consider that an error has occurred in the reception of the telegram.**

**Value *text***

**Gets the value of the String variable (last received value). Maximum amount of string characters is 252.**

**StringVariableBluetooth *component***

**The StringVariableBluetooth component.**

**Methods**

**Request**

**Request the reading of the String variable with the indicated index (send a reading telegram with the variable index). This feature is non-blocking and the reception of the telegram is expected to be notified with the response with “Received” event.**

**Set (*text* value)**

**Sets the value of the String variable (sends a write telegram with the variable index). The maximum number of string characters is 252.**

**Update**

**Request the reading of the String variable (sends a read telegram with the variable index and waits to receive the telegram with the information of the variable with the same index). This is a blocking method. “Timeout” property indicates the maximum time this function is blocking before “Timeout” event is generated.**

**Events**

**Received**

**Notifies that a telegram has been received with information on the value of the float variable.**

**value *text***

**Value with the String variable. The maximum number of received characters is 252.**

**Timeout**

**Notifies that a read timer overflow has occurred.**

**error *text***

**Text with the timeout information.**

**PROTOCOL**

**ARDUINO CODE**

**EXTENSION DEVELOPER**