

Documentación del Proyecto: Aplicación Bluetooth para Lectura de Datos

1. Introducción

El presente documento describe el desarrollo de una aplicación móvil sencilla creada mediante App Inventor, cuyo objetivo principal es establecer una conexión Bluetooth con un dispositivo externo (como un microcontrolador PIC) para recibir y mostrar información, como coordenadas u otros datos enviados.

La aplicación fue diseñada para facilitar el proceso de conexión, lectura y visualización de datos transmitidos desde el dispositivo remoto.

2. Objetivo

Crear una aplicación que permita:

- Escanear dispositivos Bluetooth cercanos.
 - Conectarse a un dispositivo seleccionado.
 - Leer datos enviados vía Bluetooth.
 - Mostrar los datos de manera clara al usuario.
 - Informar visualmente el estado de la conexión.
-

3. Componentes Utilizados

Componente	Nombre (ID)	Función Principal
ListPicker	ListPickerDispositivos	Mostrar lista de dispositivos Bluetooth disponibles
Button	BotonLeer	Leer datos enviados por el dispositivo
Label	LabelEstadoConexion	Mostrar el estado de conexión (conectado o no)
Label o TextBox	LabelCoordenadas	Mostrar datos recibidos (coordenadas)
BluetoothClient (no visible)	BluetoothClient1	Gestionar la conexión Bluetooth

4. Programación de la Aplicación (Bloques)

4.1 Inicialización

- Se inicializa una variable global llamada data con un valor vacío "" para almacenar la información recibida.

4.2 Conexión a Dispositivo Bluetooth

- **Antes de elegir un dispositivo** (ListPicker1.BeforePicking):
Se carga en el selector la lista de dispositivos Bluetooth disponibles (AddressesAndNames).
- **Después de seleccionar un dispositivo** (ListPicker1.AfterPicking):
 - Se intenta conectar al dispositivo seleccionado.
 - Si la conexión es exitosa:
 - Se actualiza LabelEstadoConexion a "Connected".
 - Se habilita el botón BotonLeer.
 - Si falla:
 - Se muestra "Failing Connection".
 - Se desactiva el botón de lectura.

4.3 Recepción de Datos

- **Al hacer clic en el botón de lectura** (Button1.Click):
 - Se verifica si hay una conexión activa.
 - Se consulta si existen datos disponibles para recibir.
 - Si hay datos:
 - Se reciben los datos.
 - Se almacenan en la variable global data.
 - Se muestran en LabelCoordenadas.
 - Si no hay conexión:
 - Se actualiza LabelEstadoConexion a "Connection Lost".
 - Se deshabilita BotonLeer.

5. Funcionamiento General

1. El usuario abre la aplicación y presiona el ListPickerDispositivos para ver los dispositivos disponibles.
2. Elige un dispositivo para conectarse vía Bluetooth.
3. La aplicación establece la conexión y muestra el estado ("Connected" o "Failing Connection").
4. Una vez conectado, el usuario presiona BotonLeer para recibir los datos enviados.
5. Los datos recibidos se muestran en LabelCoordenadas.
6. En caso de perder la conexión, la aplicación informa al usuario y desactiva la lectura de datos.

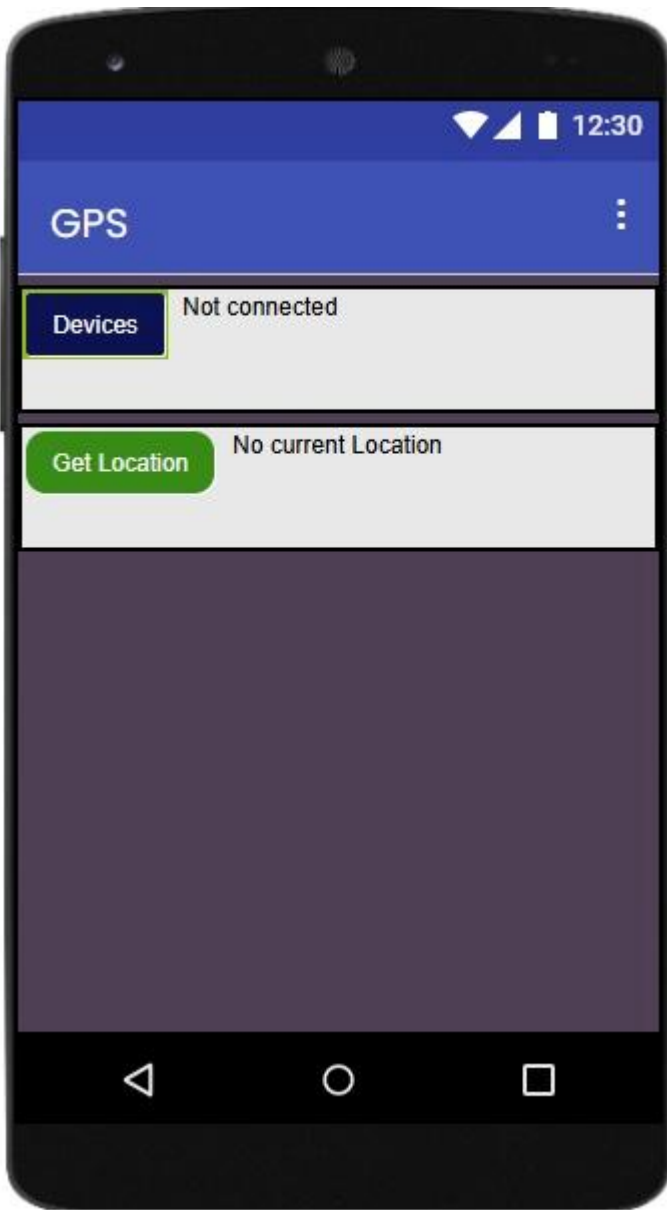
6. Conclusiones

La aplicación cumple con su objetivo de establecer comunicación Bluetooth de manera sencilla y efectiva. Gracias a su estructura basada en eventos, garantiza una interacción fluida y directa para el usuario, minimizando errores de conexión y facilitando la lectura de datos transmitidos.

El proyecto puede ser fácilmente ampliado para incluir funcionalidades adicionales, como:

- Reconexión automática.
- Almacenamiento de datos en un historial.
- Visualización de los datos en gráficos en tiempo real.

Esta solución es ideal para proyectos académicos o prototipos de interacción Bluetooth con microcontroladores.



```
initialize global data to ""  
when ListPicker1 .BeforePicking  
do set ListPicker1 .Elements to BluetoothClient1 .AddressesAndNames
```

```
when ListPicker1 .AfterPicking  
do if call BluetoothClient1 .Connect  
    address ListPicker1 .Selection  
    then set Label1 .Text to "Connected"  
        set Button1 .Enabled to true  
    else set Label1 .Text to "Failing Connection"  
        set Button1 .Enabled to false
```

```
when Button1 .Click  
do if BluetoothClient1 .IsConnected  
    then if call BluetoothClient1 .BytesAvailableToReceive > 0  
        then set global data to call BluetoothClient1 .ReceiveText  
            numberOfBytes -1  
        set Label2 .Text to get global data  
    else set Label1 .Text to "Connection Lost"  
        set Button1 .Enabled to false
```