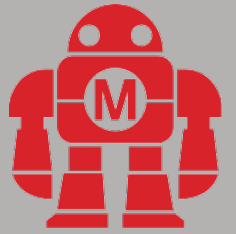


Maker Faire® CDMX

Un evento para toda la familia que reúne a la comunidad de Makers del país.

12 & 13 MAYO 2023



CENART
Centro Nacional de las Artes

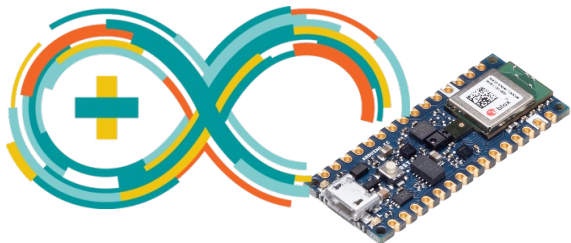


MakerFaire.mx



¡BLE para todos!

Por: Gustavo Reynaga



Quién soy.

- GUSTAVO SALVADOR REYNAGA AGUILAR.
- MAESTRÍA EN EDUCACIÓN.
- INSTRUCTOR DE OFIMÁTICA EN CECATI 132.
- 27 AÑOS DE EXPERIENCIA COMO DOCENTE.
- REDES, SOLARIS, LINUX, SERVIDORES.
- MAZATLÁN, SINALOA.

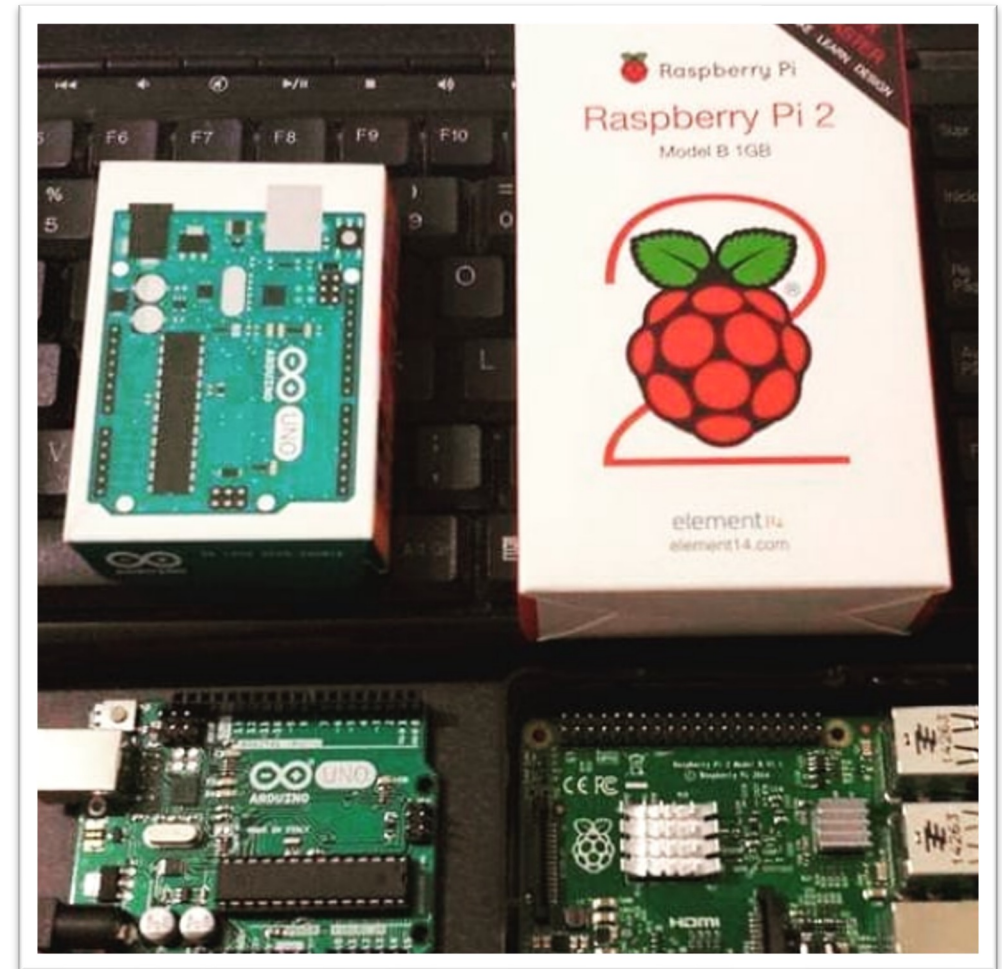


El inicio.

SEMANA SANTA DE 2015
30 MARZO - 10 ABRIL

- ARDUINO

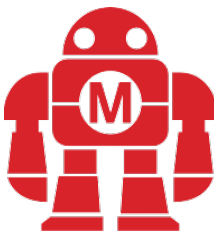
- RASPBERRY PI



Resumen



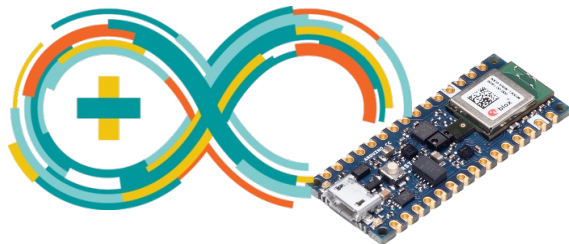
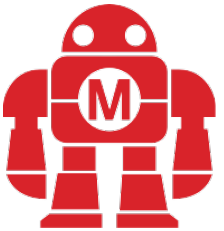
Hoy en día las aplicaciones basadas en el protocolo “WiFi”, son numerosas y bastantes difundidas, por lo tanto, existe mucha información de cómo crear una utilizando el ecosistema Arduino o similar, pero existe otro protocolo, el Bluetooth Low Energy o BLE, el cual como su nombre lo indica, está diseñado para ser de bajo consumo y resuelve problemas donde “WiFi” puede no hacerlo, a diferencia de este último, BLE no está tan difundido y por ende, hay poca información, de cómo implementarlo.



Objetivo

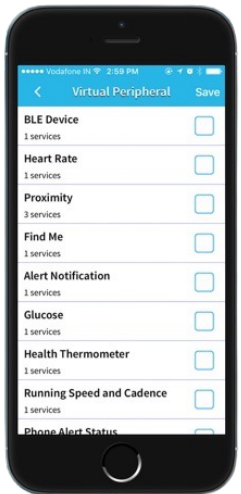
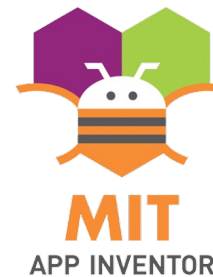


Crear un prototipo rápido de una Aplicación móvil usando AppInventor, que muestre los datos obtenidos mediante una solución BLE (Bluetooth Low Energy) basada en Arduino.

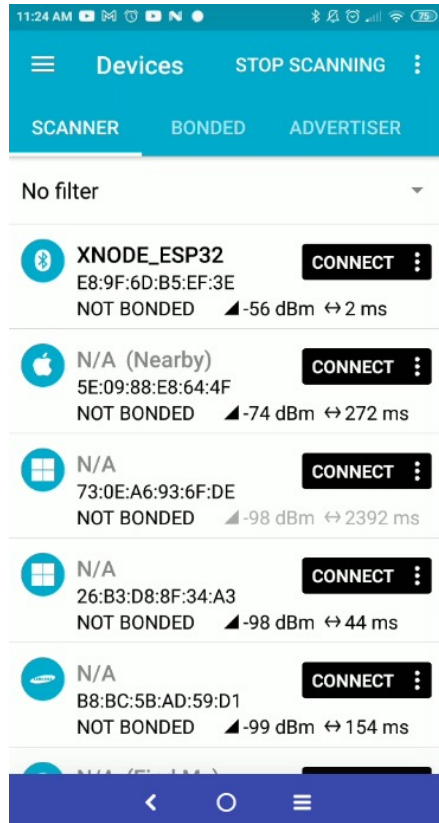


```
BLE_notify | Arduino 1.8.19
File Edit Sketch Tools Help

BLE_notify
22 #include <BLEDevice.h>
23 #include <BLEServer.h>
24 #include <BLEUtils.h>
25 #include <BLE2902.h>
26
27 BLEServer* pServer = NULL;
28 BLECharacteristic* pCharacteristic = NULL;
29 bool deviceConnected = false;
30 bool oldDeviceConnected = false;
31 uint32_t value = 0;
32
```



Roles



Central /Server
Scanning information



Periférico/Client
Advertising/broadcasting
information

Perfiles

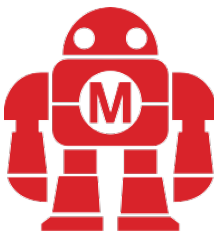


GAP

Es el acrónimo para el Generic Access Profile, y se encarga de controlar las conexiones y los anuncios en BLE. GAP es lo que permite que tu dispositivo sea público hacia el exterior y determina como dos dispositivos pueden (o no) interactuar entre ellos.

GATT

Es el acrónimo de Generic Attribute Profile, y define la manera en que dos dispositivos BLE pueden comunicarse usando los Servicios y Características



Servicios y Características



```
BLE_notify | Arduino 1.8.19
File Edit Sketch Tools Help

BLE_notify

32
33 // See the following for generating UUIDs:
34 // https://www.uuidgenerator.net/
35
36 #define SERVICE_UUID          "4fafc201-1fb5-459e-8fcc-c5c9c331914b"
37 #define CHARACTERISTIC_UUID   "beb5483e-36e1-4688-b7f5-ea07361b26a8"
38
```

Profile

Service

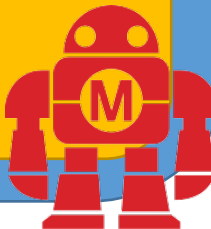
Characteristic

Characteristic

Service

Characteristic

Characteristic



UUID Universally Unique ID

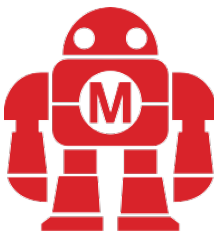


Son las siglas en inglés del Identificador Universalmente Único. El UUID es el equivalente a Folio Fiscal, está compuesta por 32 dígitos hexadecimales, mostrados en 5 grupos separados por guiones.

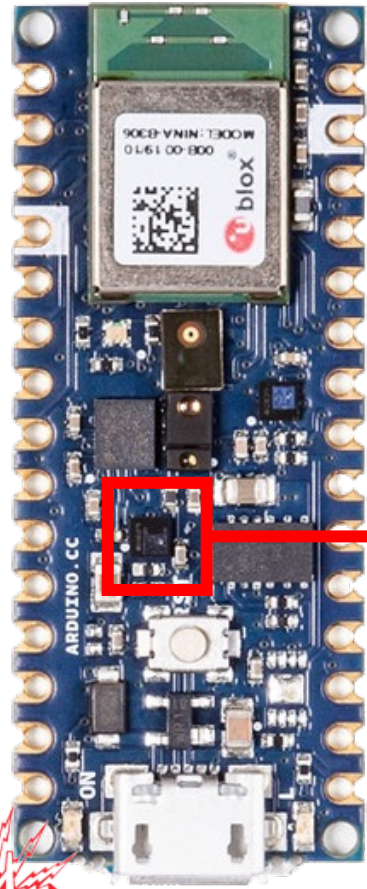
Por ejemplo, "**0x180F**" o "**6E400001-B5A3-F393-E0A9-E50E24DCCA9E**".

Para ser un poco más prácticos, veamos, por ejemplo, el servicio oficial de nivel de batería **Battery Service**.

Este servicio tiene un UUID de **0x180F** y una característica llamada Nivel de batería de UUID **0x2A19**.

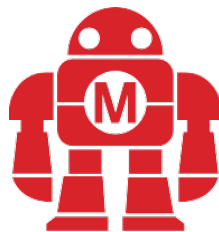
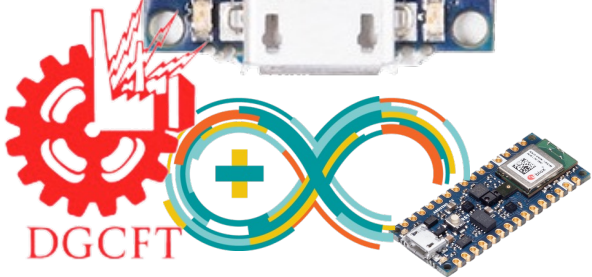


Hardware opción 1.



Nano BLE Sense

HTS221

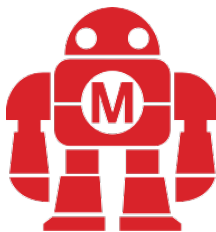
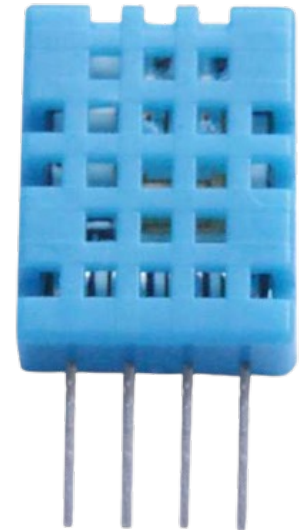
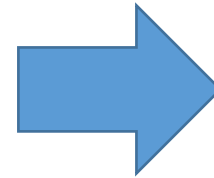


Hardware opción 2.



X-NODE MCU WIFI BLE
(ESP32-WROOM)

DHT11

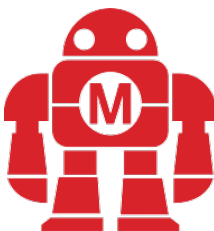


Código Fuente



Mostrar en
el IDE de
Arduino

```
BLE_notify | Arduino 1.8.19
File Edit Sketch Tools Help
BLE_notify
32
33 // See the following for generating UUIDs:
34 // https://www.uuidgenerator.net/
35
36 #define SERVICE_UUID          "4fafc201-1fb5-459e-8fcc-c5c9c331914b"
37 #define CHARACTERISTIC_UUID "beb5483e-36e1-4688-b7f5-ea07361b26a8"
38
ESP32 Dev Module, Disabled, Default 4MB with spiiffs (1.2MB APP/1.5MB SPIFFS), 240MHz (WiFi/BT), QIO, 80MHz, 4MB (32Mb), 921600, Core 1, Core 1, None on COM8
```



Crear aplicación BLE

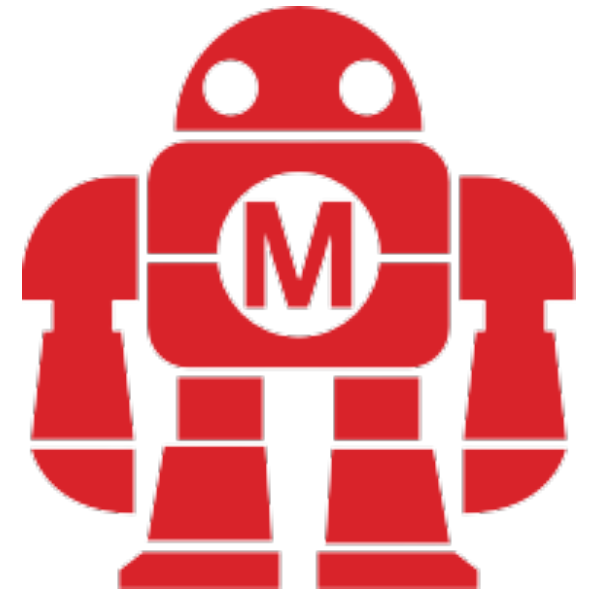


Mostrar en el
entorno de
App Inventor

The screenshot shows the MIT App Inventor web interface. At the top, the MIT App Inventor logo is on the left, and navigation links (Projects, Connect, Build, Settings, Help) and user information (My Projects, View Trash, Guide, Report an Issue, English, gustavo.reynaga@gmail.com) are on the right. Below the navigation bar, the project name "BLE_05" is displayed. The interface is divided into four main panels: 1. Palette: A sidebar on the left containing a search bar and categories like "User Interface" (with components like Button, CheckBox, DatePicker, Image, Label, ListPicker, ListView, Notifier, PasswordTextBox, Slider, Spinner, Switch, TextBox, TimePicker, WebViewer) and "Layout", "Media", and "Drawing and Animation". 2. Viewer: The central area showing a mobile phone emulator with a screen titled "Screen1". Above the emulator, there are checkboxes for "Display hidden components in Viewer" and a dropdown for "Phone size (505,320)". 3. Components: A panel on the right showing a tree view of the app's components, currently showing "Screen1". 4. Properties: A panel on the far right showing the properties for the selected component, "Screen1". Properties include "AboutScreen" (a text input), "AccentColor" (set to "Default"), "AlignHorizontal" (set to "Left"), "AlignVertical" (set to "Top"), "AppName" (set to "BLE_05"), "BackgroundColor" (set to "Default"), "BackgroundImage" (set to "None..."), "BigDefaultText" (a checkbox), "BlocksToolkit" (set to "All"), "CloseScreenAnimation" (set to "Default"), "DefaultFile" (set to "App"), "HighContrast" (a checkbox), and "Icon" (set to "None..."). There are also buttons for "Rename" and "Delete" in the Components panel, and an "Upload File ..." button in the Media section.



Gracias!!!



Gustavo Salvador Reynaga Aguilar

Twitter: @gsreynaga