Table of Contents

1.	Foreword	2
2.	Hardware used	2
3.	Development environment	2
4.	Operating instructions	2
5.	INO code explanation	5
6.	AI2 code explanation	7
7.	EXTENSIONS used	9

1. Foreword

This BLE example is intended to show how an Adroid device running an app made by MIT Al2 can exchange data (characters) with an ESP32 board by means of BLE (Bluetooth Low Energy) communication.

2. Hardware used

- An Android device: LENOVO pad TB-8505XS (8" pad) featuring Android 9
- An ESP32 board Dev Module featuring the BLE communication capability

3. Development environment

- DELL E7270 notebook c/w Windows 10 OS
- Brave browser
- Arduino 2.3.2 IDE

4. Operating instructions

- 1) Download on your PC the following files:
 - The AI2 app: BLE_Test.aia
 - The ESP32 code: BLE_TestAl2.ino
- 2) Compile the .ino file with the Arduino IDE (be sure to have all the ESP32 relevant libraries installed).

Most probably in your IDE preferences menu tab you should have (green highlighted rows):

Prefe	renze
	Impostazioni
Perce	uso della cartella deoli sketch:
C:	URL aggiuntive per il Gestore schede
8	one aggianate per a destore solicide
Din	Aggiungi degli URLs aggiuntivi, uno per ogni riga
Sci	
Ter	https://arduino.esp8266.com/stable/package_esp8266com_index.json
Lin	https://dl.espressif.com/dl/package_esp32_index.json
Mo	https://raw.githubusercontent.com/espressif/arduino-esp32/gh-pages/package_esp32_index.json
Wa	
wa	
	Clicca per ottenere la lista di collegamenti per le schede di terze parti, non schede ufficiali.
	(ANNULLA) (OK
UR	

Be sure to have selected your relevant ESP32 board. This example is based on:



The example .ino file is intended to be connected to the Serial Monitor at a baudrate of 115200. Please be sure to have this setting in your IDE.



NOTE: (COM11 is the serial line to which my board was connected: please verify and set your true one !

BLETestAl2_uart.ino

Download the code on the ESP32 board:

Once downloaded, press the reset button: on the serial Monitor it should appear:

```
11:43:43.715 -> ets Jul 29 2019 12:21:46
11:43:43.715 ->
11:43:43.715 -> rst:0x1 (POWERON_RESET),boot:0x13 (SPI_FAST_FLASH_BOOT)
11:43:43.715 -> configsip: 0, SPIWP:0xee
11:43:43.715 -> clk_drv:0x00,q_drv:0x00,d_drv:0x00,cs0_drv:0x00,hd_drv:0x00,wp_drv:0x00
11:43:43.715 -> clk_drv:0x00,q_lre:0x00,d_drv:0x00,cs0_drv:0x00,hd_drv:0x00,wp_drv:0x00
11:43:43.715 -> load:0x3fff0030,len:1344
11:43:43.715 -> load:0x40078000,len:13964
11:43:43.715 -> load:0x40080400,len:3600
11:43:43.715 -> load:0x400805f0
11:43:44.773 -> Waiting a client connection to notify...
```

The ESP32 is then waiting for a BLE connection toward a client (your AI2 app).

On your PC open AI2 and load the BLE_Test.aia	3)	On your PC open	AI2 and load t	he BLE_T	est.aia.
-----------------------------------------------------------------	----	-----------------	----------------	----------	----------

BLE_Test	My projects
Palette	Start new project
	Import project (.aia) from my computer
Search Components	Import project (.aia) from a repository
User Interface	Move To Trash

4) Create the .apk and download it on your Adroid device (be sure that it is capable of BLE)

5) Install the .apk; the following icon should appear on the screen of your device (if not on home page, please search in app's general directory/screen page):



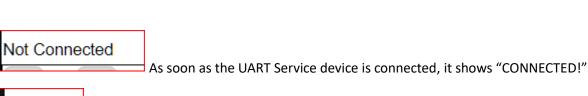
- 6) In your Android Settings tab: Bluetooth Devices, do a search for new devices and look for "UART Service". When found, do pair it.
- 7) When paired, you can tap on the icon, it should appear the following page:

Scan&Connect Disconnected A B C RECEIVED: Received	BLE Simple				
Not Connected	Scan	scionnect			
A B C RECEIVED: Received	Dis	connect			
A B C RECEIVED: Received			1		
	0.000 720	90.65			
	ALC: NOT THE OWNER OF	_	(lizzen)	RECEIVED: Received	
	ALC: NOT THE OWNER OF	_	c	RECEIVED: Received	
	A	В	C	RECEIVED: Received	

Where:

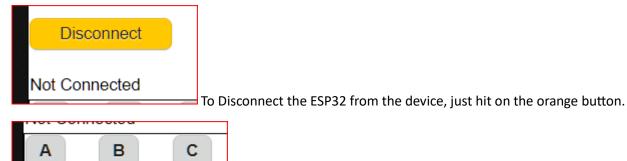


by tapping on it a search of BLE reachable devices starts. As soon as the UART Service is found, the search stops and the device is connected automatically.



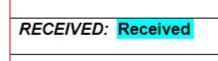


Shows the IP of the BLE device just connected



By hitting any of them the relevant character is sent to the ESP32

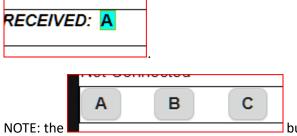




If the BLE loop = transmission from APP to the ESP and loopback of the same character from ESP to APP has worked fine, it shows the same character just sent ('A', 'B', 'C').

For example by hitting the

And on the app screen you should see:



buttons are enabled only when the ESP32 is connected,

otherwise they are "greyed" and disabled.

5. INO code explanation

Many comments are available, row-by-row, in the code, nevertheless some details are written here.



#define SERVICE_UUID "6E400001-B5A3-F393-E0A9-E50E24DCCA9E" // UART service UUID
#define CHARACTERISTIC_UUID_RX "6E400002-B5A3-F393-E0A9-E50E24DCCA9E" // its characteristics
#define CHARACTERISTIC_UUID_TX "6E400003-B5A3-F393-E0A9-E50E24DCCA9E"

Service and Characteristics creation (one for reading and one for writing data)

std::string myStringForUnit8((char*)&rxValue[0], 1); // trick to allow sending back one single character to the client

Mandatory to allow a single character to be sent back to the app.

for (int i = 0; i < cxValue.length()-1; i++) // AT2 sends the linefeed to close the string: it shall be discorded

Since the app sends a string terminated by a linefeed character (0x0A), it shall be discarded, this explains why .length()-1



that does not stop the CPU.

Pag. 7

6. AI2 code explanation

Condensed image:

	when Clock_StarForDevices
when Screent Initialize d 🗧	when Clock Register Shing
when ListPicker1 AtterPic	
when B_ScanForDevices Cil	when BinetochiLE2 Connect
when B_Disconnect_Click d	when DivefoothEE2 Discom
when B_A Click do call B	when Bluetholthi, E2 Shings > millaizs global (VUID_Rs to 6620000385533#363E0A9E502240CCA9E)
when B_B.Click do call B_	to Disable Buls do set B A.
when B_C Click do call B.	to Ensble_Buts do set B_A 👗
when B_EXIT Click do call.	

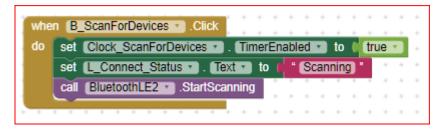
The following UUID's do replicate the same that are set in the ESP32 code

initi	alize	glol	bal	UL	JID_	Srv	/C	to	C	* (6	E40	000)1-B	5A3-	-F39	93-E	0A	9-E	50	E24	4D(cc	A98	9,	1	
initi	alize	glol	bal	ÜL	JID_	Tx) to	-C	۳ (6E4	1000	002	-B5/	\3-F	393	-E0/	A9-	E5(DE2	24D	000	CAS	E)	•	+	
initi	alize	glol	bal	UL	JID_	Rx) to		*	6E4	400	003	-B5/	\3-F	393	-E0	A9-	E5	0E2	240	bC	CAS	9E)		÷ +	
				· · · · · · · · · · · · · · · · · · ·																						
÷						+				+ +	+	+	+		+	+	+	÷	+	+	+	+	+	+	+	
+	· ·	+	•	•	+ +	+ +	•			+ +	*	++++	+ +	· ·	+++++++++++++++++++++++++++++++++++++++	+ +	+	+	+	+	+	+	+ +	+	+	

When you hit the



The scanning clock is enabled:



This clock has a period of 500 milliseconds: i.e. each 500 ms it performs a new scan, looking for new devices presenting themselves on the BLE radio.

The search (scan) continues until the	UART Service device	e is found. See below:
------------------	-----------------------	----------------------------	------------------------

when Clobe SceniforDevices Timer
the set plotaling Devices Found at the Contra text Buelook 5240 (Device Letter)
TentBox1 * Tent * Bluetoot*LE2 * DeviceList *
O * I length of the link pet (blocal Lies Devices Found and CIC)
then for each Lacindae from D
to I longith of his Hud Cost groop List Devices Found
de A Comment Marine La Les des Auf herr find get global List DevicesFound 33 index get Listinda(30)
piece "UART Depuce"
time ou electrosectere energies in faire
set @local@dayAntinane of Select and how list sold at spaces (select list new list or (CoostList Devices Journals) index or (Listindex Sal
pet (Connect Status M . East M to I select ist tem lat get Const Lat Devices Found mits get Listindex
sut EUColossia . ECISIA to 1000 SEE center 1
call Electrocal Electrocal Stop Scanning
call EldersonitERER ConnectWithAddress
uddress Vort global BLE scollcant
GibbalList_DevicesFoundid M () creade emply int
· · · · · · · · · · · · · · · · · · ·
A B C
Once connected the server, the three buttons
scan clock is deactivated.

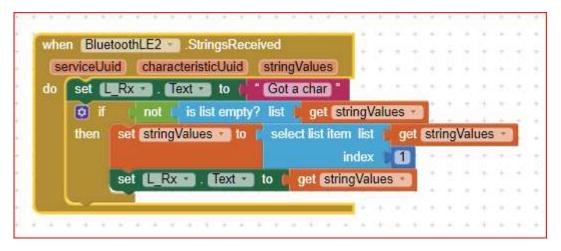
NOTE: it is not used, for the time being, the BLE device to be

connected is fixed (UART_Service).

A second clock is used to periodically register the client to be allowed to read strings, used also to show the connection status updated in "real time" (i.e. 500 ms).

io	D if	BluetoothLE2 IsDeviceConnected	ŧ
	then	call BluetoothLE2	ł
		serviceUuid get global UUID_Srvc	•
		characteristicUuid get global UUID_Rx •	
		utf16 false	1
		set L_Connect_Status . Text . to Connected!	1
	else	set L_Connect_Status . Text . to Connect	
		set LRX . Text > to (i.
	S		

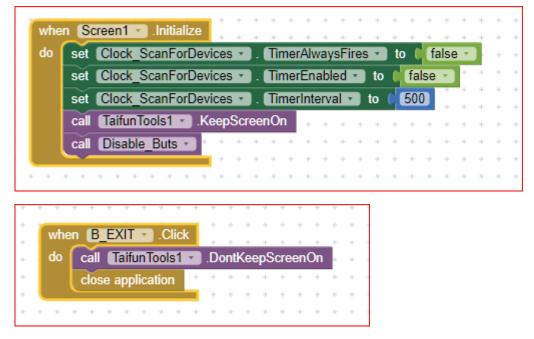
The following event is raised whenever a new characteristic is available at ESP32 side. In this case the ESP echoes the received character, therefore this event is raised "immediately" after a button ('A' 'B' 'C') is pressed, and its character is sent to the ESP.



The received data is in a list form, therefore to show the received character, the first element of such list shall be extracted. In case the received data is not recognized, a simple warning "Got a char" is shown instead.

7. EXTENSIONS used

1) The app uses the "TaifunTools" extension to keep the screen on until the EXIT pushbutton is hit.



Mamy thanks to Taifun. You can find many Extensions, Snippets and ready made code on their web site:

https://puravidaapps.com/

2) BluetoothLE, from ewPatton release 20230728