Project number: 2023-1-PL01-KA220-SCH-000154043

IoT4Schools
"Bringing the Internet of Things in school education as a tool to address 21st century challenges"

MIT App Inventor application design

Teachers' guidelines

Authors: C. Papasarantou, R. Alimisi

Organization: EDUMOTIVA

Co-funded by
the European Union

Table of contents

1 Creating the interface of the application

1.1 Designing the application

Design is a rather free process, and is usually based on the creator's aesthetics. The following instructions are indicative and present a rather simplified version of the appearance of the interface that our application can have.

Before starting to design the application, it is important to be aware of the components that need to be included. Figure 1 presents a preview of the interface based on the needs of the project.

Figure 1: A preview of the interface

In particular, the application should have the following components:

Connect: A component that scans for all available Bluetooth Low Energy devices in the area, and open a list of results. From this list, the user will select the micro:bit's Bluetooth address. The connection is then established automatically. To enable this feature, we will use a "***ListPicker***" button.

Disconnect: A button that, when pressed, terminates the connection between the micro:bit and the user's smart device.

Status: A label indicating connectivity status.

Steps …: A label indicating the counted steps

Image: An image to graphically enhance the interface design

1.2 Design menu

Open the MIT App Inventor software (https://appinventor.mit.edu/) and create a new project from the "Projects" menu. Give the project a name of your choice (e.g., Pedometer project).

To develop an application, you must first design the interface of the application (add all the components needed to make your application work). Therefore, we will begin by working on the Designer menu (**1**) (Figure 2).

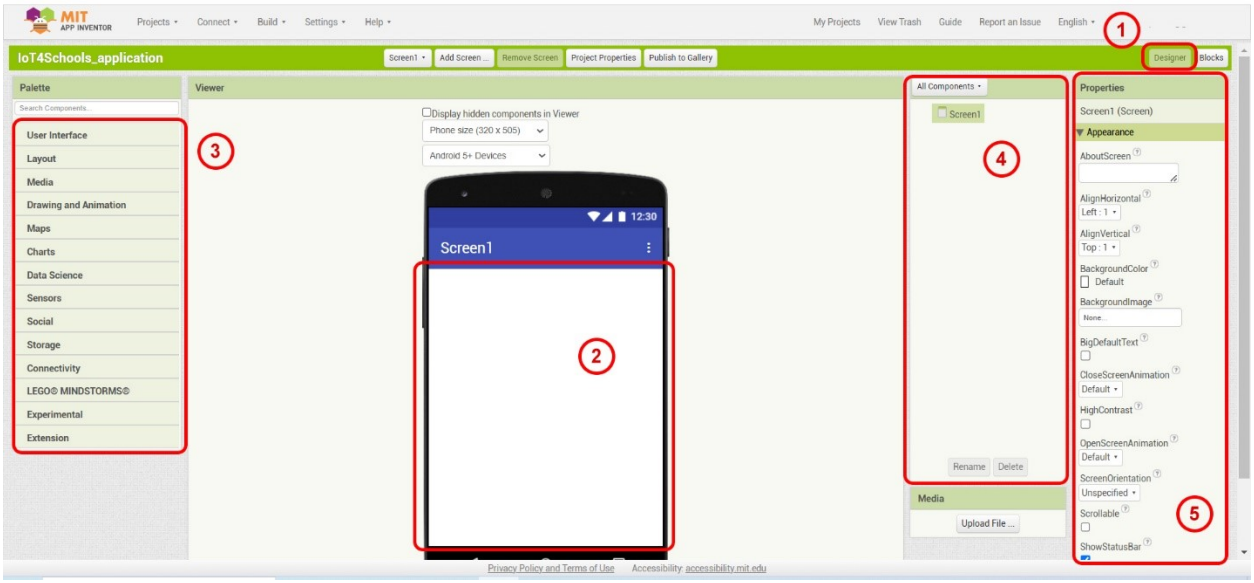


Figure 2: The Designer menu and the included tabs

In the view tab, and in the preview of the screen **(2)** you can add all the needed components by dragging and dropping them. The needed components can be found in the Palette tab **(3)**. Some components are non-visible. To make sure that you have added a component, check the "All components" tab **(4)**, and see if the added component appears on the list. To change the properties of an added component, use the Properties tab **(5)**.

1.3 Adding layouts

You can use layouts to organize the components you want to add. Layouts are located on the Palette tab.

First, drag and drop two Horizontal Arrangement layouts, and place them one after the other. From the Properties tab, set their Heigh to Automatic, and their Width to "Fill parent". Then place a Vertical Arrangement inside the second Horizontal Arrangement layout. When you have added all the layouts, the screen preview will look like Figure 3.

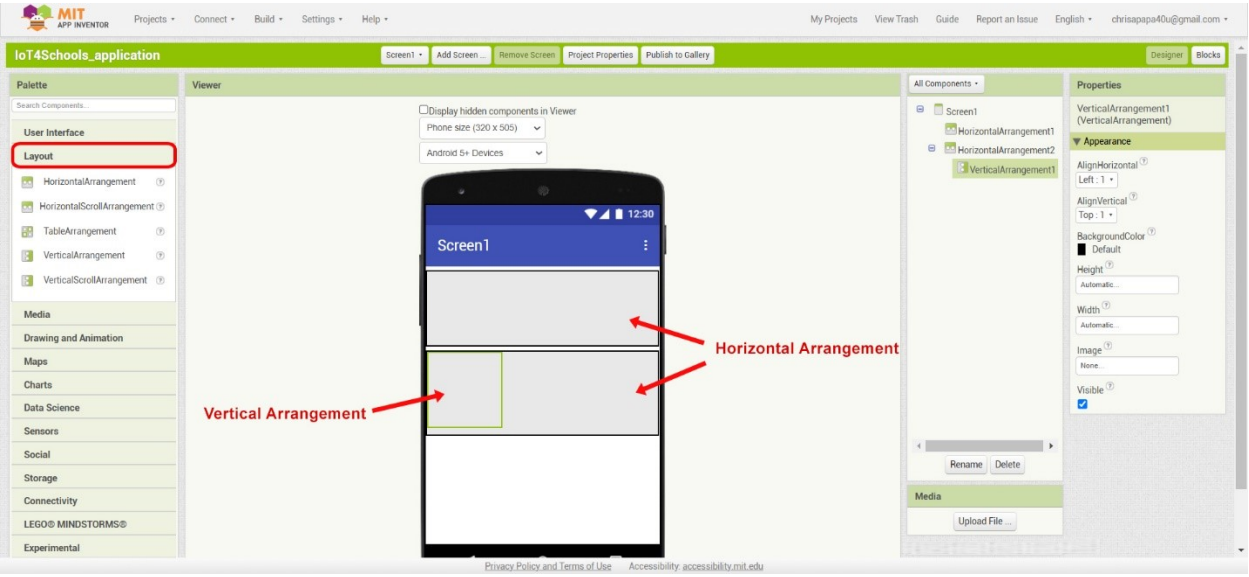


Figure 3: Adding two Horizontal and one vertical arrangement layouts

In the first Horizontal Arrangement layout you will place the component needed to connect the application to the micro:bit, and in the second Horizontal Arrangement Layout (which also contains a Vertical Horizontal Arrangement layout) you will add the components needed to receive and display the counted steps.

Tip: The height and width of a layout can be changed at any time. This gives you the freedom to make any adjustment you want once you have added the components (i.e., buttons, labels etc.) to the layouts.

1.4 Adding a ListPicker and a button component

Inside the first Horizontal Arrangement layout, add a ListPIcker (1) and a Button (2) component (Figure 4). These will be the Connect button and the Disconnect button, respectively.

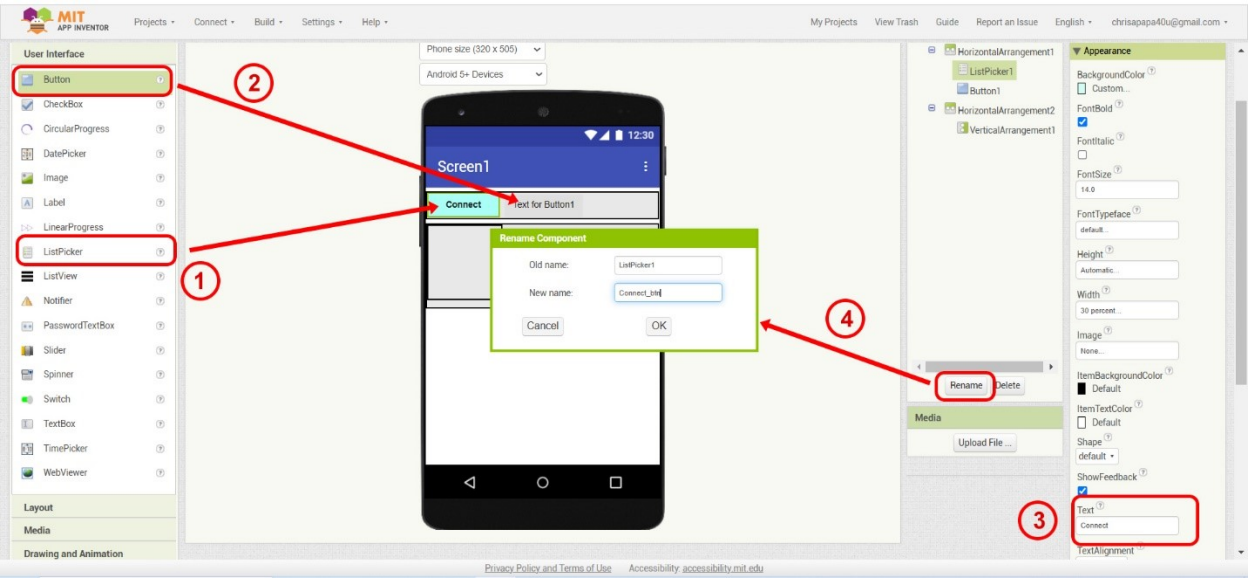


Figure 4: Adding and modifying the ListPicker and Button components

The Properties tab allows you to make several modifications to each component such as changing the background and/or the text colour, the Font size and the text written on the component (e.g. from List Picker 1 to Connect, and from Button 1 to Disconnect) **(3)**. The All components tab also allows you to change the name of each component, by selecting a component and clicking on the “Rename” button **(4)**. It is highly recommended that you change the name of the components to something meaningful (e.g. change ListPicker 1 to Connect_btn and Button1 to Dis_btn) as this will help you to easier identify the components when programming the application.

Important note: Do not use the same word for the text name and the button name as this will cause App Inventor to malfunction and not be able to build the application.

1.5 Adding labels

The next step is to add three labels (Figure 5). Add one inside the first Horizontal Arrangement layout, next to the Disconnect button, and the other two inside the Vertical Arrangement layout, located at the second Horizontal Arrangement layout. Change the text of label 1 to “status”, the text of label 2 to “Steps”, and the text of label 3 to “…”. Rename also each one of these components through the “All components” tab (e.g. Label1 to Label_Connectivity, Label2 to Steps_label, Label3 to Counter). Feel free to do any modifications on the size, the color and the style of the Fonts.

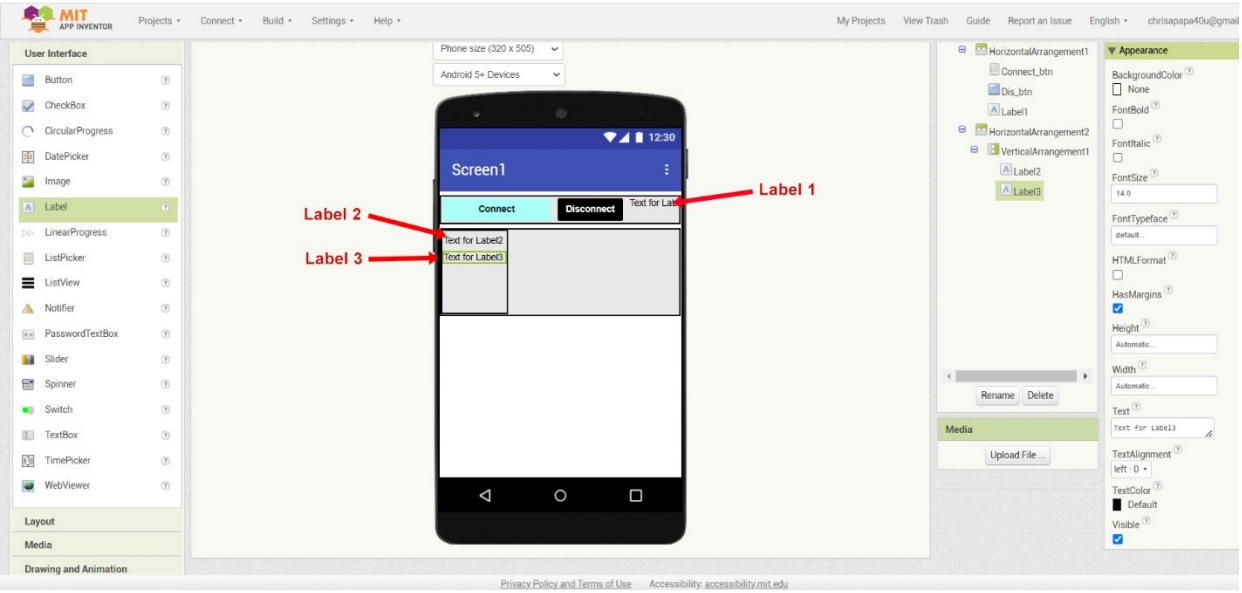


Figure 5: Adding the three labels

1.6 Adding an image

The last step is optional, as it involves adding an image to graphically enhance the interface.

From the Palette tab, drag and drop an “Image” component **(1)** on the second Horizontal Arrangement layout, next to the Vertical Arrangement layout (Figure 6). A small icon will appear on the screen. To add an image from your computer, click on the Picture field in the Properties menu, and select “Upload File” from the drop-down menu **(2)**. Find the image you want to add and press ok. The selected image will appear in the drop-down menu **(2)**. Select the image and press ok. Then make any necessary changes to the height and width of the image to make it the size you prefer.

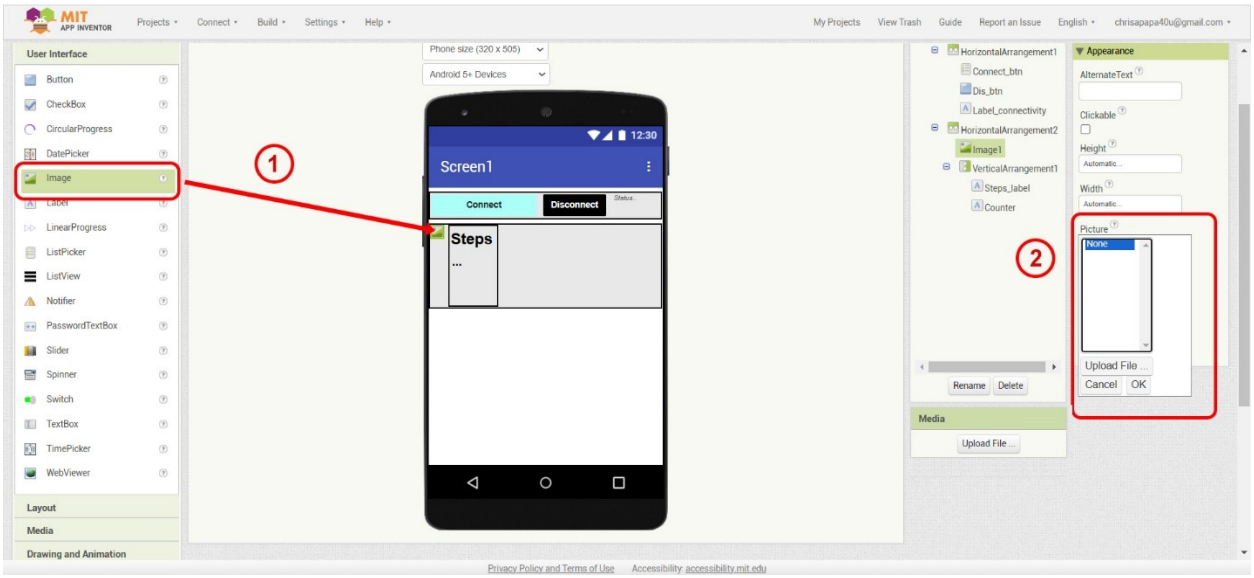
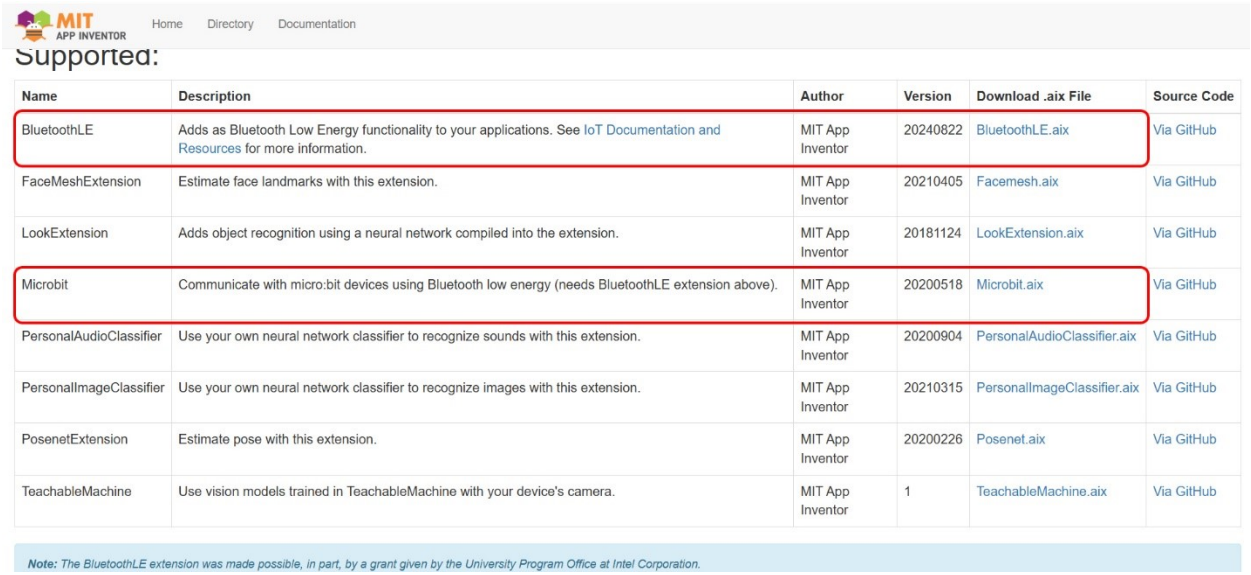


Figure 6: Adding an image component

1.7 Importing extensions

The next step is to add some components that will allow the connection between the application and the pedometer device. Specifically, we need to use the Bluetooth LE extension and the Microbit_Uart_Simple extension. The former allows us to establish the Bluetooth connection between our smart device and the micro:bit, while the latter allows us to exchange data once the connection has been established.

To be able to use these extensions, you need to download them locally to your computer. To do this, click here https://mit-cml.github.io/extensions/ and download to your computer the BluetoothLE.aix file and the Microbit.aix file (Figure 7).



MIT APP INVENTOR Home Directory Documentation

Supported:

Name	Description	Author	Version	Download .aix File	Source Code
BluetoothLE	Adds as Bluetooth Low Energy functionality to your applications. See IoT Documentation and Resources for more information.	MIT App Inventor	20240822	BluetoothLE.aix	Via GitHub
FaceMeshExtension	Estimate face landmarks with this extension.	MIT App Inventor	20210405	Facemesh.aix	Via GitHub
LookExtension	Adds object recognition using a neural network compiled into the extension.	MIT App Inventor	20181124	LookExtension.aix	Via GitHub
Microbit	Communicate with micro:bit devices using Bluetooth low energy (needs BluetoothLE extension above).	MIT App Inventor	20200518	Microbit.aix	Via GitHub
PersonalAudioClassifier	Use your own neural network classifier to recognize sounds with this extension.	MIT App Inventor	20200904	PersonalAudioClassifier.aix	Via GitHub
PersonalImageClassifier	Use your own neural network classifier to recognize images with this extension.	MIT App Inventor	20210315	PersonalImageClassifier.aix	Via GitHub
PosenetExtension	Estimate pose with this extension.	MIT App Inventor	20200226	Posenet.aix	Via GitHub
TeachableMachine	Use vision models trained in TeachableMachine with your device's camera.	MIT App Inventor	1	TeachableMachine.aix	Via GitHub

Note: The BluetoothLE extension was made possible, in part, by a grant given by the University Program Office at Intel Corporation.

Figure 7: The extensions that need to be downloaded

After downloading the extensions, return to App Inventor. On the Palette section, click on the Extension tab, and then click on the ***Import extension*** selection (*Figure 8*).

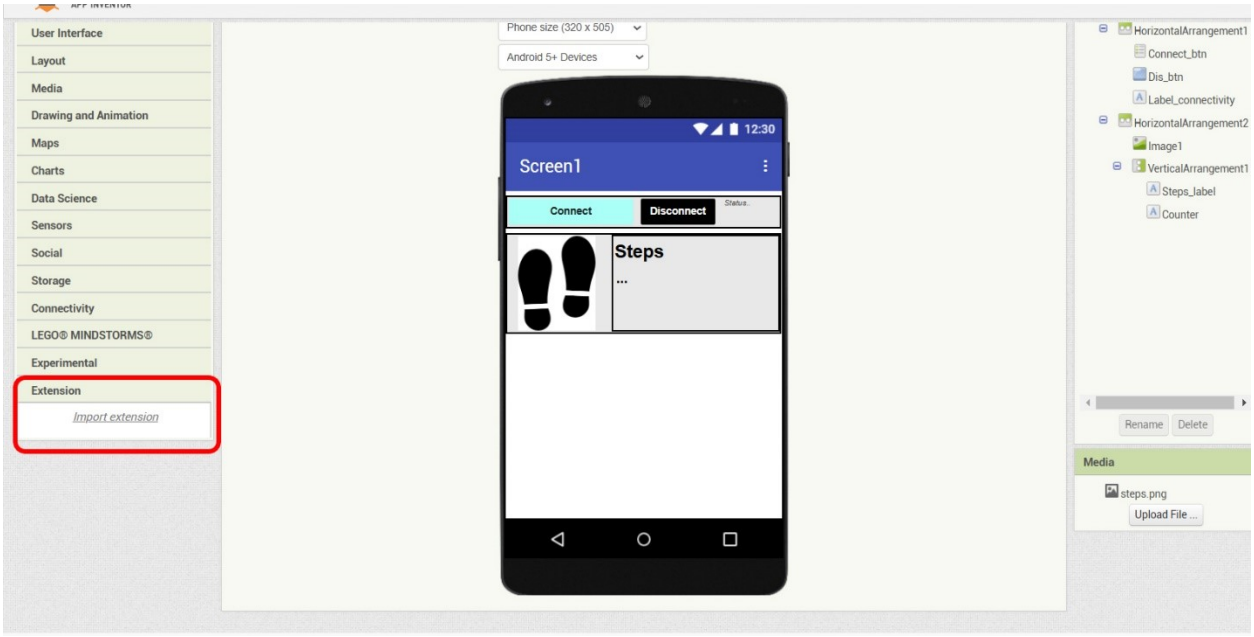


Figure 8: The Importing extensions tab

From the pop-up menu click the *Choose File* button **(1)** to browse to your local folder and select the downloaded extension (*Figure 9*). Make sure that "*From my computer*", located above the *Choose File* button, is selected. Once the extension file has been found and selected, click the *Import* button **(2)**. The imported extension will appear under the Extension tab.

Note: Importing might take a few seconds

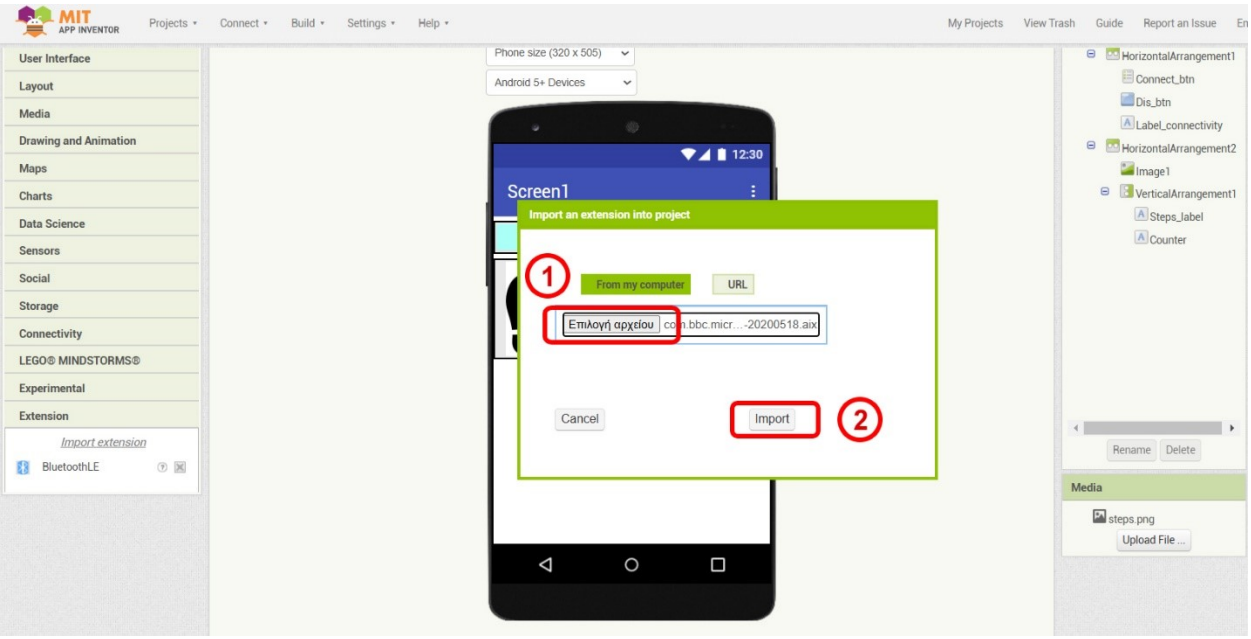


Figure 9: Importing an extension from the computer

To add the extensions to the designed application, drag and drop them into the design area. Extensions are normally non-visible components. Therefore, these components appear below the design area, in the "Non-visible components" section.

After importing the *Microbit.aix* file, you will notice that several extensions appear under the Extension tab. For the purposes of this project, you only need to use the Microbit_Uart_Simple extension (Figure 10).

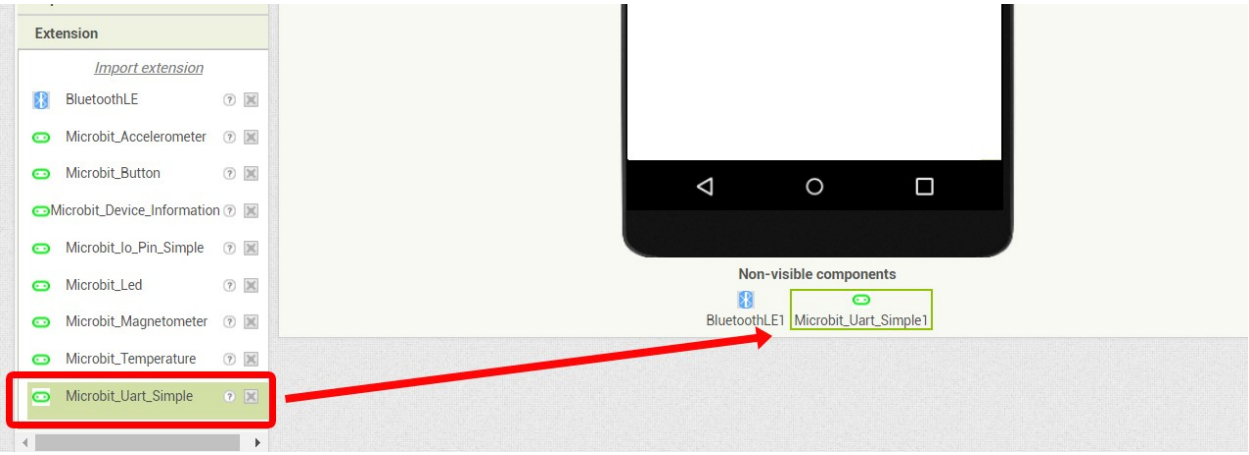


Figure 10: Adding the Microbit_Uart_Simple extension

To use the Microbit_Uart_Simple extension, you also need to set a Bluetooth device. To do this, select the aforementioned extension, go to the Properties menu and in the Bluetooth device field, select BluetoothLE1 from the drop-down menu (Figure 11).

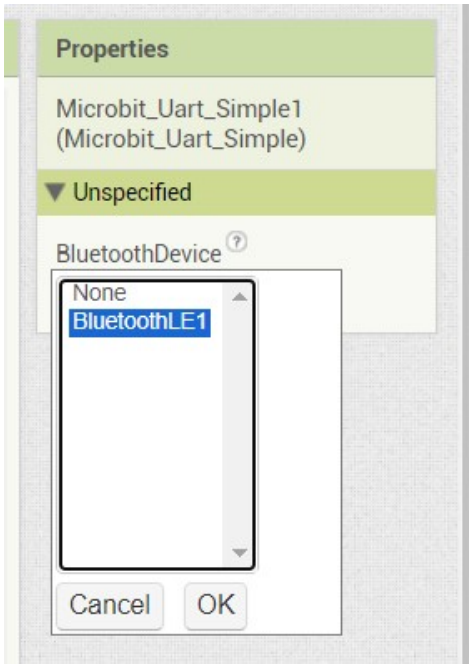


Figure 11: Selecting a Bluetooth device on the properties menu of the Microbit_Uart_Simple extension

- The application is now ready to be programmed. -