

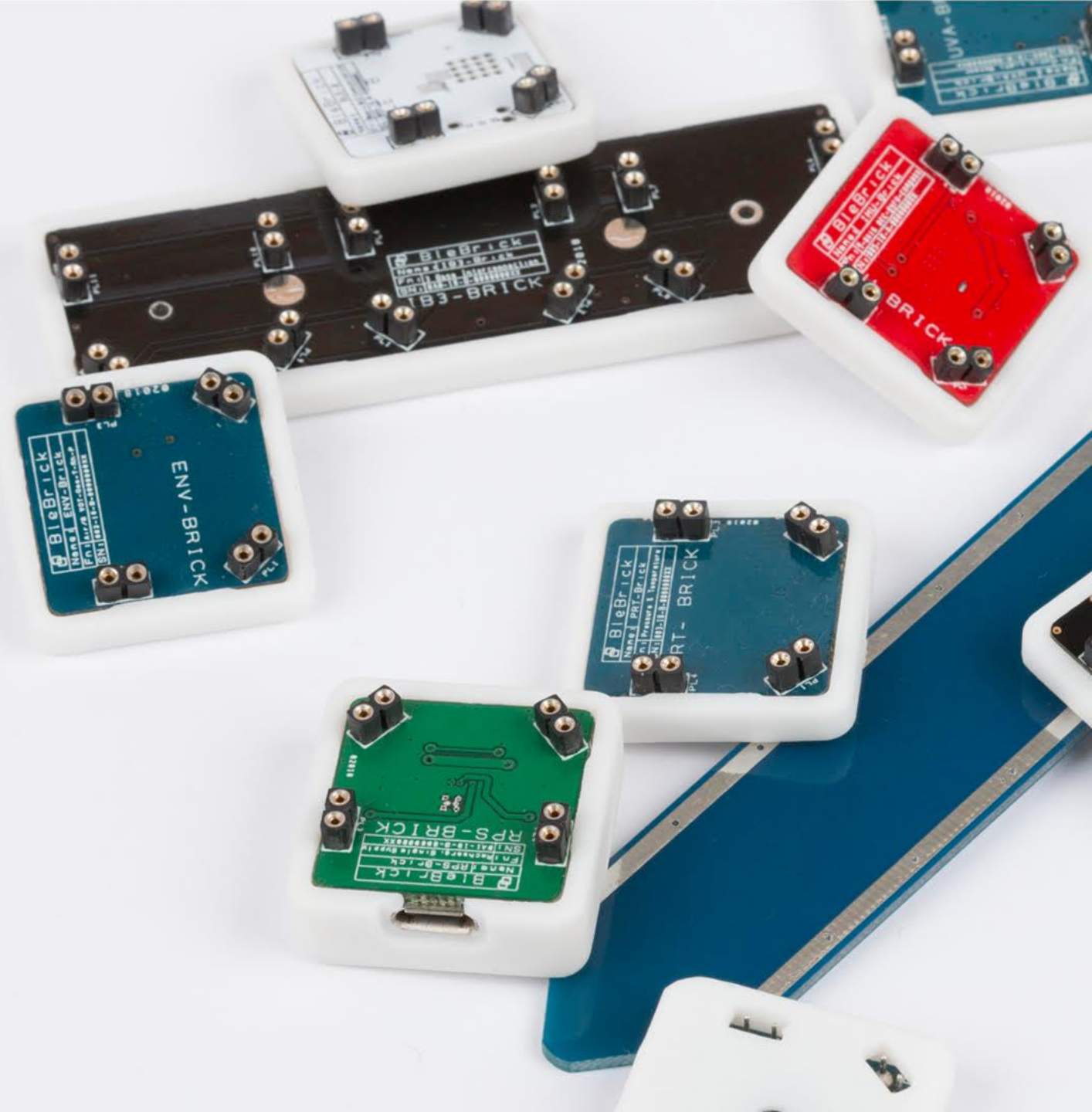


# Blebricks

Bricks to build your ideas

# Bleb App Inventor

Cominciamo





## Ecco alcuni semplici esempi di come possiamo divertirci ad utilizzare i Blebricks

Rilevare la temperatura (usando solo il BLE-B).

- I. Apri MIT App Inventor 2 [ai2.appinventor.mit.edu](http://ai2.appinventor.mit.edu)
- II. Clicca su **Projects** - Start a new project e assegna un nome alla tua app, ad esempio "ReadTemperature"
- III. Carica le estensioni **Basic Bleb App Inventor** selezionando il file `it.bleb.simple.appinventor.aix` che puoi scaricare dalla pagina Downloads del nostro sito [bleb.it](http://bleb.it)  
(Ricorda di caricare sempre le estensioni per ogni nuovo progetto)

The image shows the MIT App Inventor 2 interface. On the left, the 'Palette' is visible with the 'Extension' category selected. A red arrow points from the 'Extension' category to an 'Import extension' dialog box. The dialog box has two tabs: 'From my computer' and 'URL'. The 'From my computer' tab is active, and a file explorer window is open, showing a folder named 'BLEB' containing several files. The file 'it.bleb.simple.appinventor.aix' is selected. A red arrow points from the file explorer to the 'Import extension' dialog box. The dialog box also has a 'Cancel' button.

Nome	Stato	Ultima modifica	Tipo	Dimensione
Apps	✓	04/02/2019 20:03	Cartella di file	
BLEB_AppInventor_021018	✓	04/02/2019 20:06	Cartella di file	
OldApps	✓	04/02/2019 20:04	Cartella di file	
TutorialApps	✓	04/02/2019 20:04	Cartella di file	
Bleb_App_Inventor_Basic.docx	✓ R	15/01/2019 19:21	Documento di Mi...	12
it.bleb.appinventor.16.full.aix	✓	06/02/2019 22:11	File AIX	212
it.bleb.appinventor.16.simpler.aix	✓	06/02/2019 22:11	File AIX	185
it.bleb.appinventor.aix	✓ R	07/02/2019 23:55	File AIX	213
it.bleb.simple.appinventor.aix	✓ R	07/02/2019 23:55	File AIX	186
TutorialApps.zip	✓ R	04/01/2019 20:21	Cartella compressa	1.225

# Bleb App Inventor – Sezione Designer

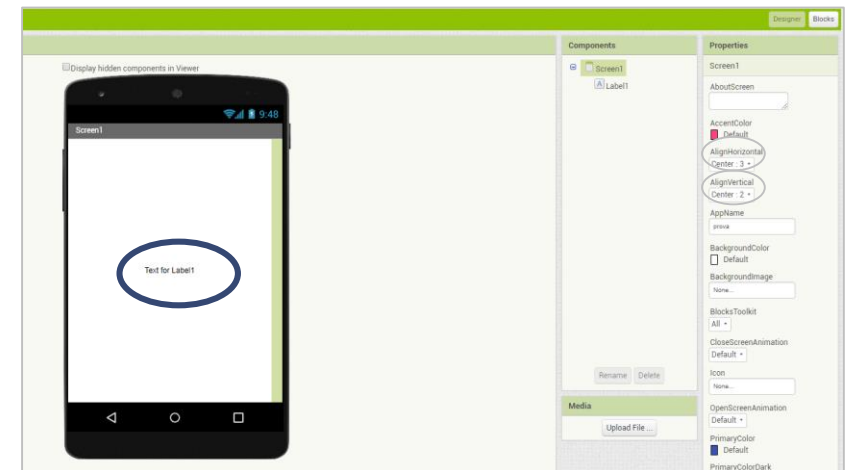
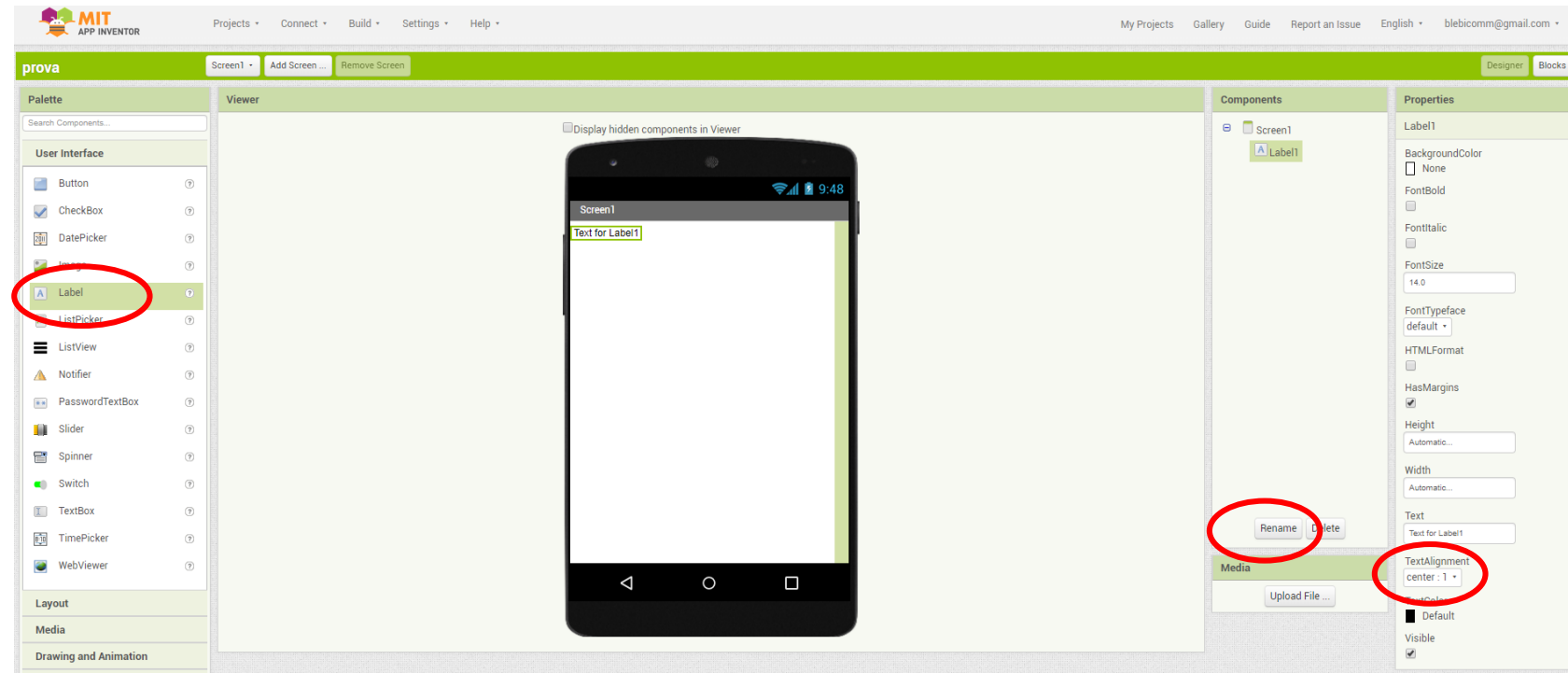
1. In **Palette** (a sinistra) apri il menù **User Interface**, e trascina una «Label» sullo schermo dello smartphone

2. In **Components** (a destra) seleziona la «Label» e clicca su **rename** per assegnarle il nome **TemperatureLabel** \*.

3. In **Properties** (a destra) seleziona «TextAlignment» → «center» per centrare la scritta all'interno della label

4. In **Components**, clicca su «Screen1» e imposta «center» nei campi «AlignHorizontal» e «AlignVertical». In questo modo la scritta risulterà centrale sullo schermo.

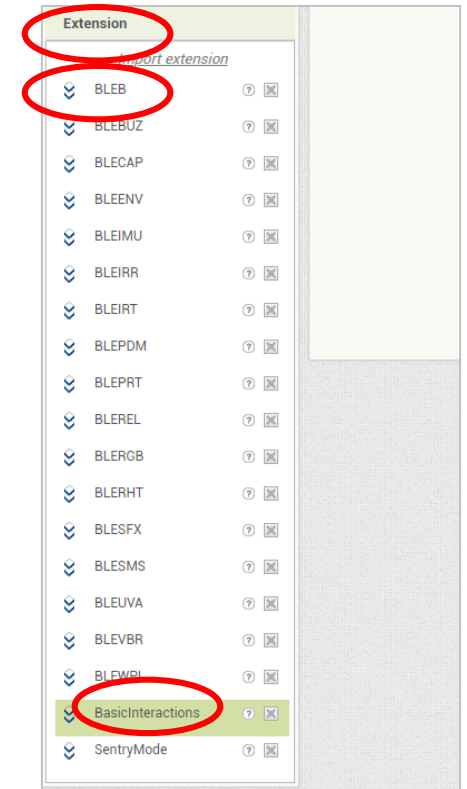
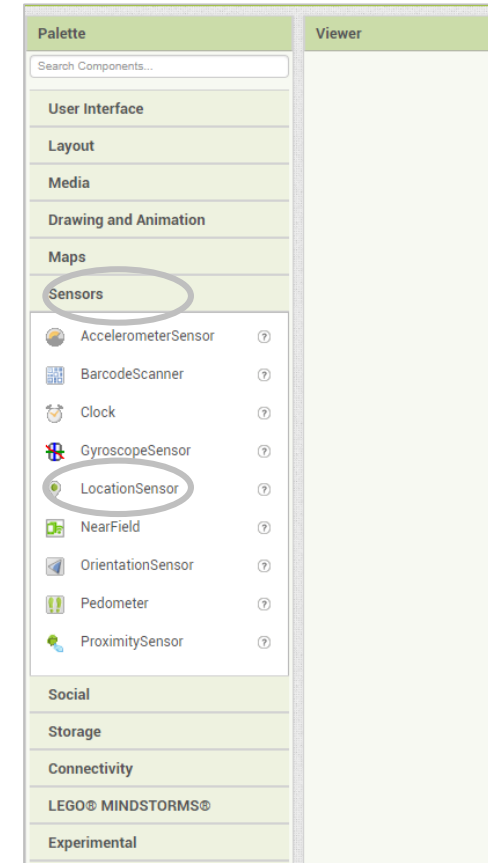
\*è consigliabile rinominare ciascuna Label per potersi ricordare a cosa serve



# Bleb App Inventor – Sezione Designer

5. In **Palette** apri il menù **Sensors**, seleziona e trascina «LocationSensor» sullo schermo (risulterà un componente non visibile ma essenziale per il funzionamento del Bluetooth Low Energy sul tuo smartphone e quindi dei Blebricks).

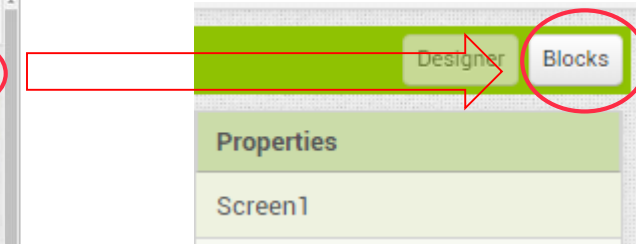
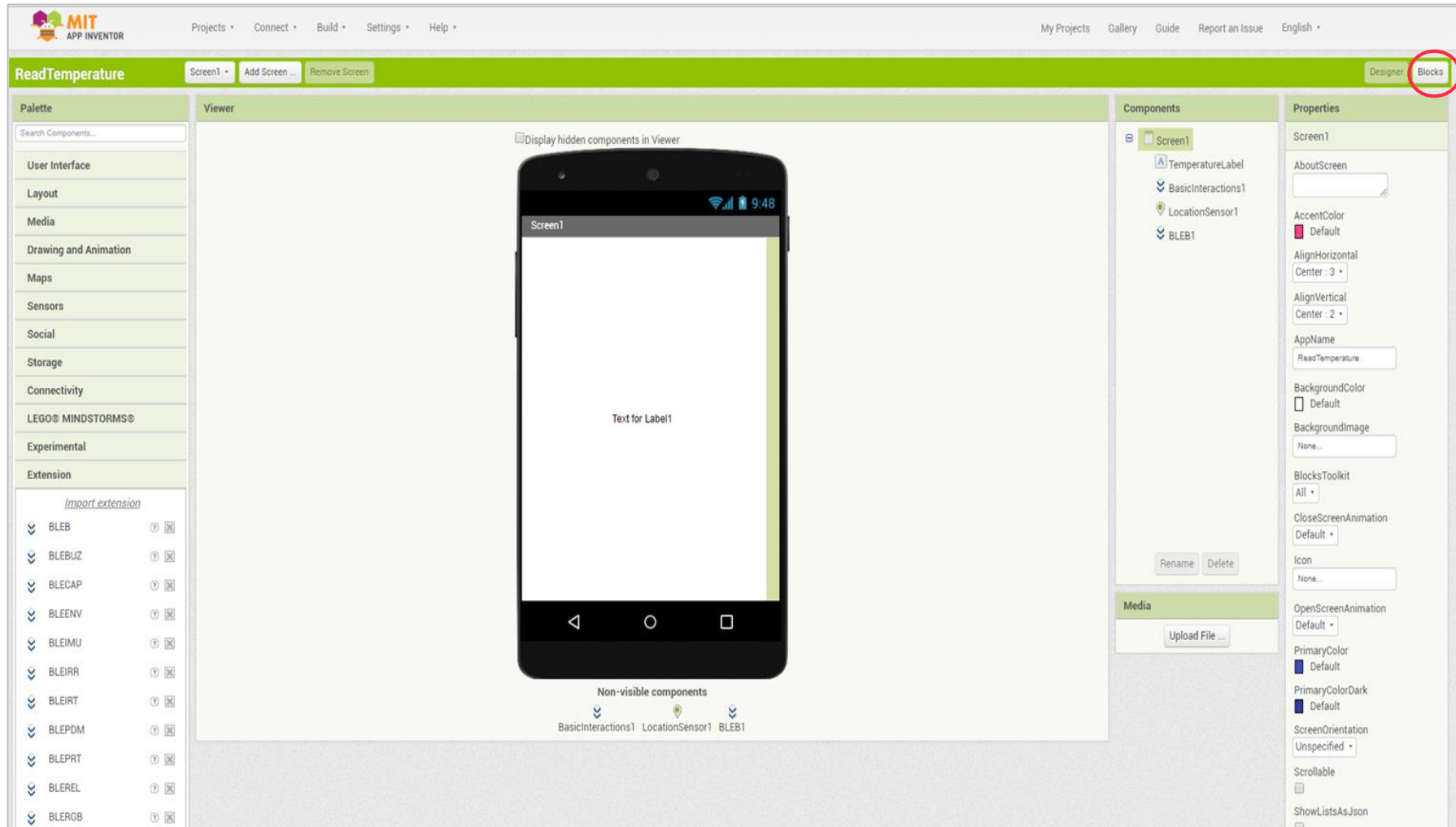
7. In **Palette**, clicca su **Extension** e trascina le nostre estensioni «BLEB» e «BasicInteractions» sullo schermo (risulteranno come componenti non visibili ma necessari per comunicare con i Blebricks).



# Bleb App Inventor – Sezione Designer

Una volta fatto questo, hai completato il lavoro sulla sezione **Designer**, in cui puoi definire l'aspetto della tua app e selezionare i componenti utili.

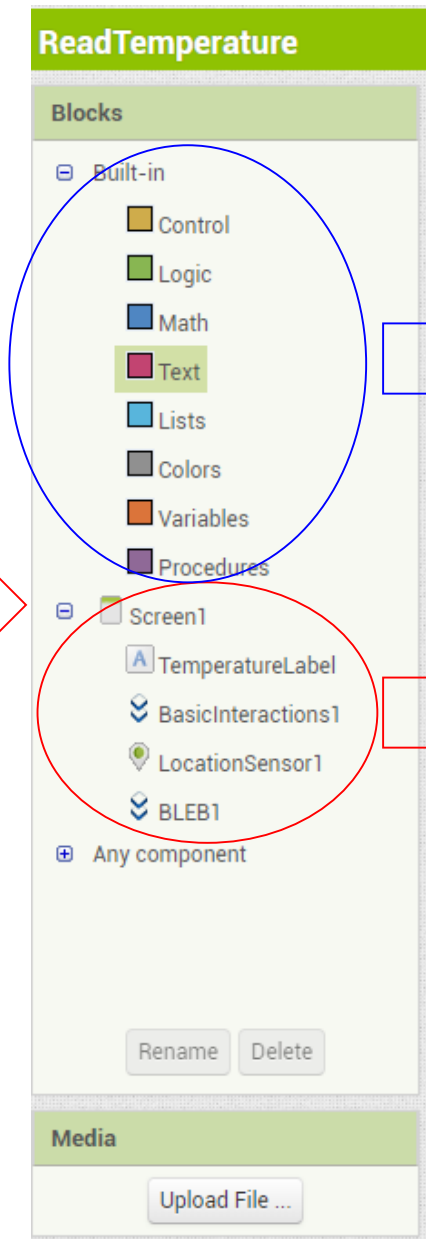
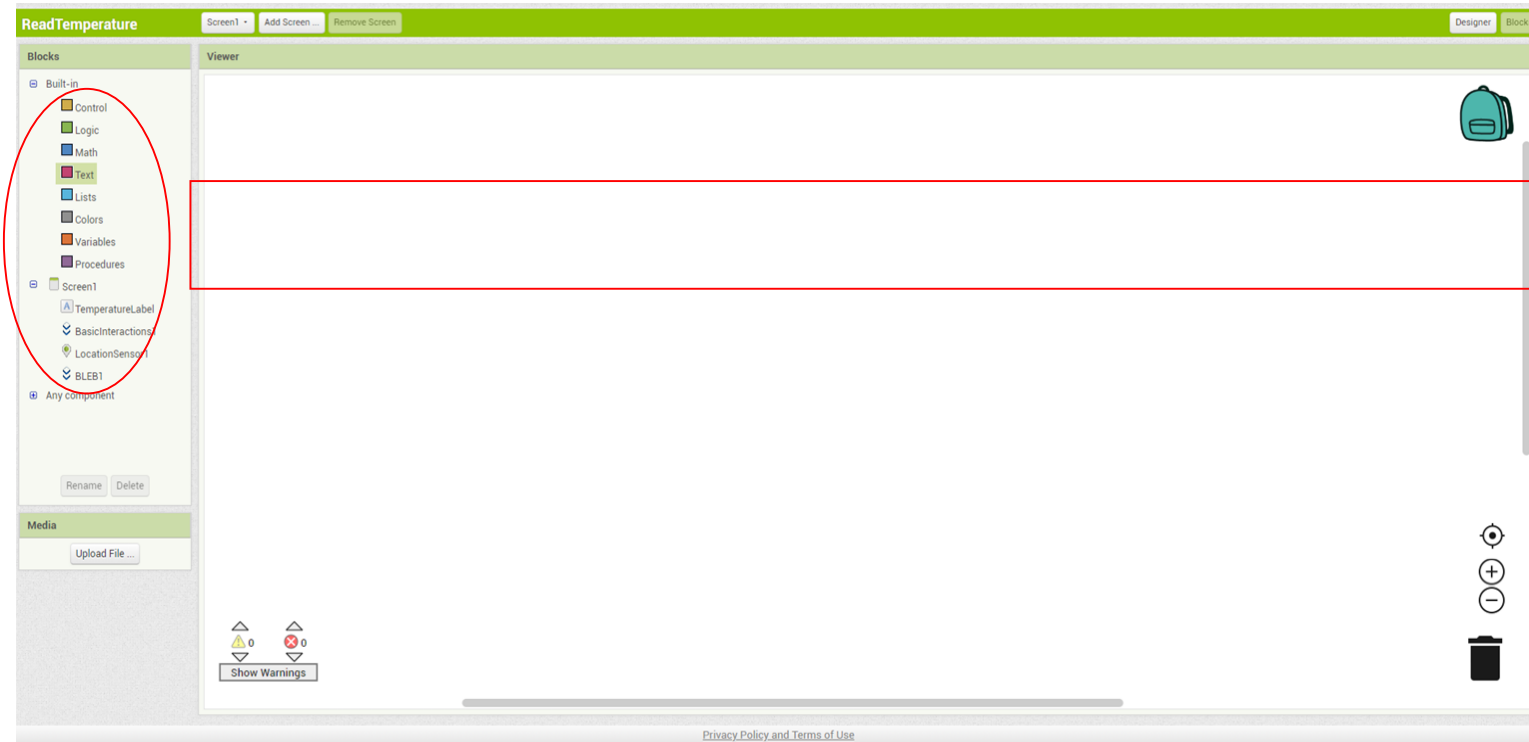
In alto a destra sullo schermo, clicca su **Blocks** per passare alla sezione in cui andremo a definire la parte funzionale della tua app, cioè cosa vogliamo che l'app faccia effettivamente



# Bleb App Inventor – Sezione Blocks

## INIZIAMO A PROGRAMMARE

Sulla sinistra dello schermo, nel menu **Blocks** vengono visualizzati i blocchi elementari (comuni a tutte le applicazioni) e i componenti visibili e non che abbiamo selezionato.

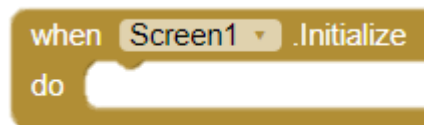


blocchi generici

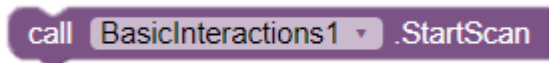
componenti visibili e non

# Bleb App Inventor – Sezione Blocks

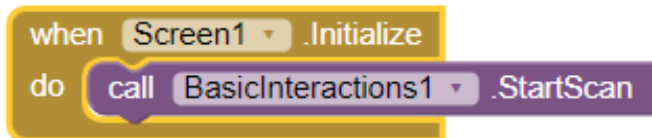
1. Clicca su «Screen1» e trascina sullo schermo il blocco che scatta non appena viene aperta l'app:



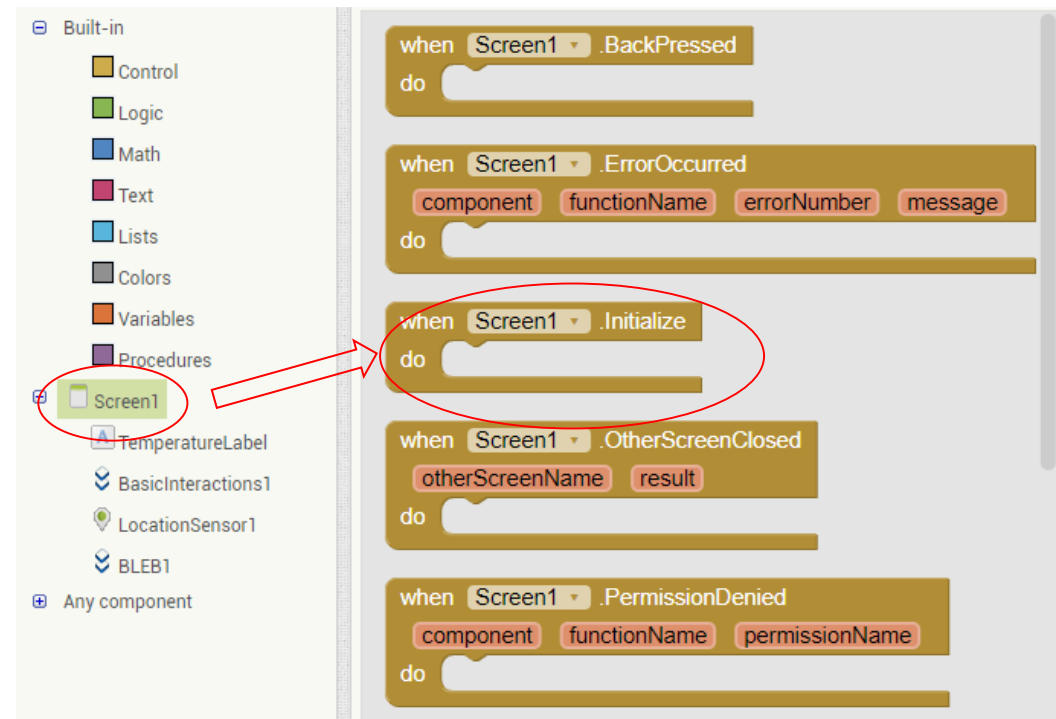
2. Allo stesso modo, clicca su «BasicInteractions1» e trascina sullo schermo il blocco che attiva lo scan Bluetooth sullo smartphone:



3. Posiziona il secondo blocco dentro al primo:



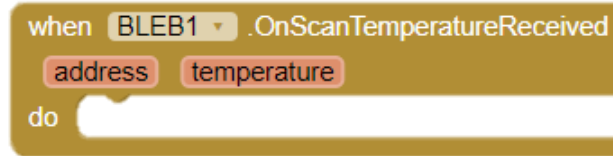
Il blocco che abbiamo così realizzato serve quindi ad **attivare lo scan Bluetooth sullo smartphone** per permettere di rilevare il BLE-B più vicino **appena viene aperta l'app**





# Bleb App Inventor – Sezione Blocks

4. Clicca su «BLEB1» e trascina sullo schermo il blocco che scatta non appena lo smartphone riceve informazioni sulla temperatura misurata da parte del BLE-B:



5. Clicca su «TemperatureLabel», seleziona il blocco che permette di impostare il testo della label e trascinalo all'interno del primo blocco:

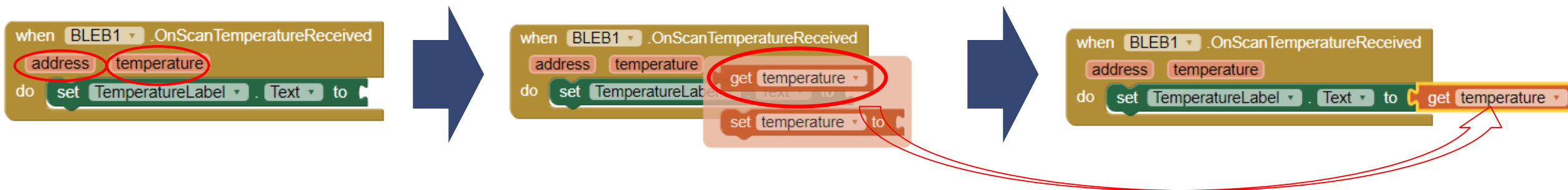


6. A questo punto ci sono due possibilità:

a) Creare ed utilizzare una variabile dal blocco «Variables»

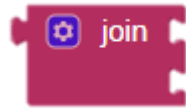
b) Utilizzare le variabili interne già presenti all'interno del blocco («address» e «temperature»), scegliamo questa.

7. Sposta il mouse sopra la variabile «temperature», seleziona «get temperature» così da poter leggere (get) il valore di temperatura ricevuto dal BLE-B ed attaccalo al blocco che permette di impostare il testo della label:



# Bleb App Inventor – Sezione Blocks

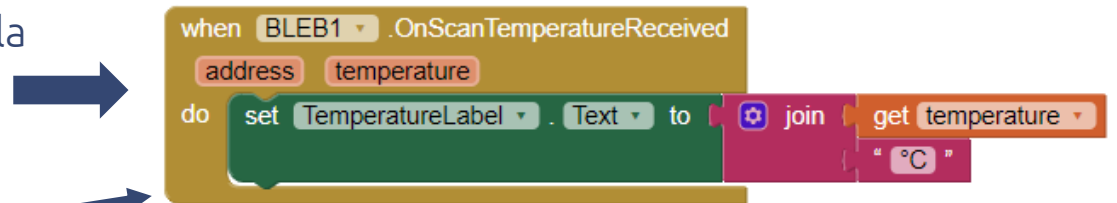
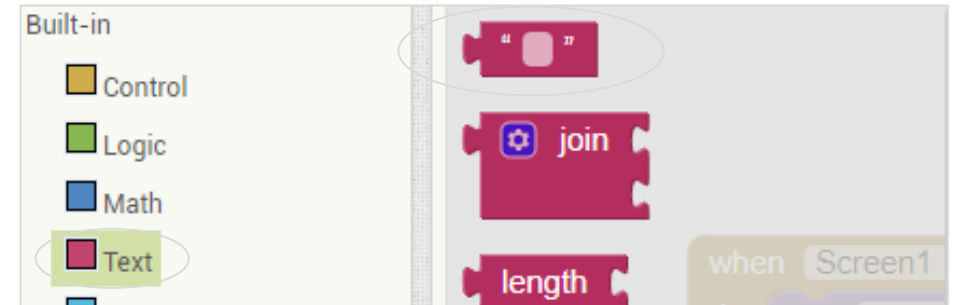
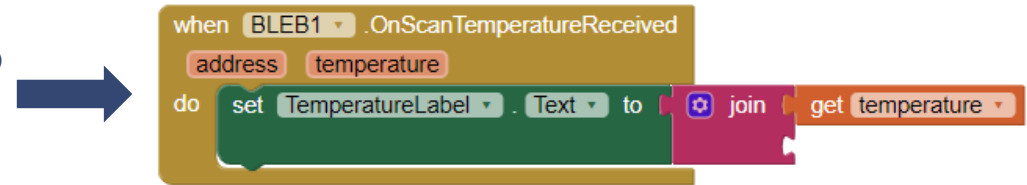
8. Tra i blocchi generici, clicca su «Text» e trascina il blocco che serve unire due testi, come mostrato in figura:



9. Allo stesso modo, trascina sullo schermo il blocco che permette di inserire un testo:



10. Posiziona i blocchi come mostrato in figura e scrivi all'interno del blocco di testo vuoto «°C» per indicare che la temperatura è espressa in gradi Celsius.



Molto bene, la tua prima App con Bleb App Inventor è fatta. Testiamola subito!

Il blocco che abbiamo così realizzato serve quindi a **scrivere il valore della temperatura misurata dal BLE-B** su una **label** ogni volta che **viene ricevuta questa informazione**, **aggiungendo anche l'unità di misura**

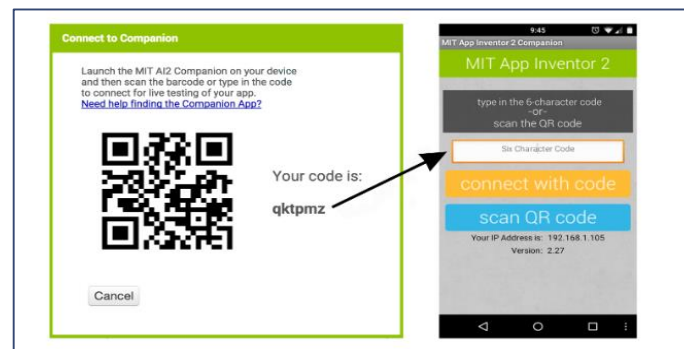
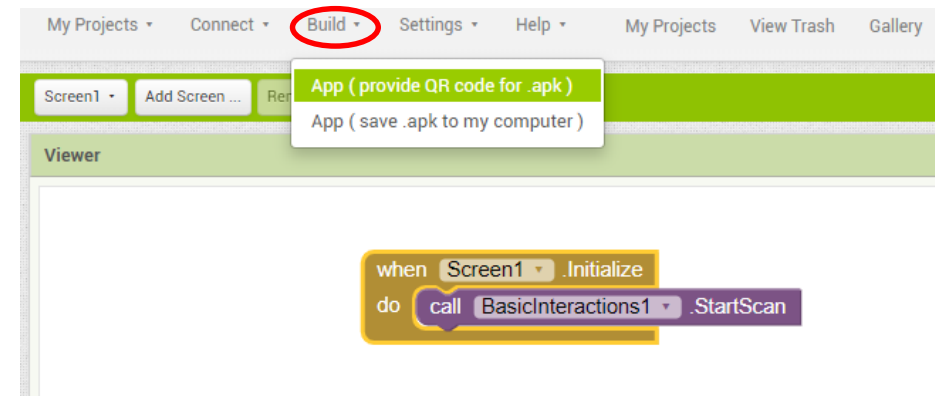
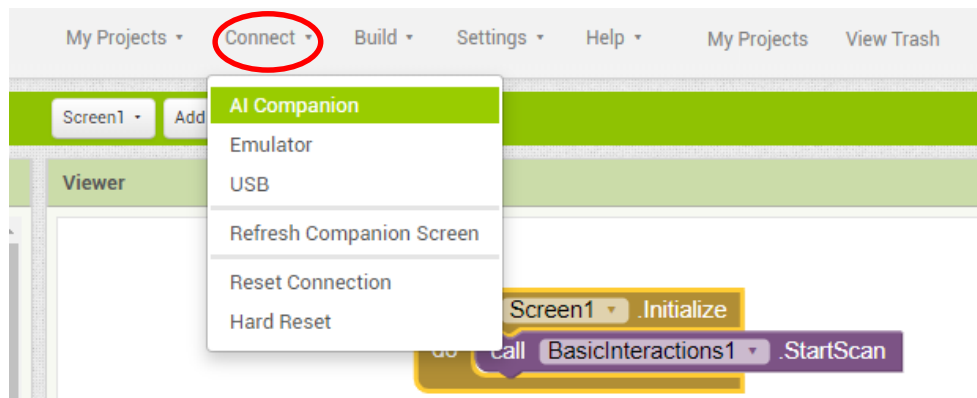
# Bleb App Inventor – Fase di Test

TEST

Per vedere la tua app funzionare sul tuo smartphone, ci sono due possibilità:

a) Provarla «online», cliccando in alto su **Connect** → **AI Companion** e scannerizzando il codice QR con l'app MIT AI2 Companion

b) Esportarla ed installarla sullo smartphone, cliccando su **Build App** e scannerizzando il codice QR (se scegli provide QR code for .apk) oppure copiando il file .apk sulla memoria dello smartphone (se scegli save .apk to my computer)



In entrambi i casi hai bisogno di leggere il QR Code dallo schermo del pc mediante il cellulare, lanciando l'App Mit AI2 Companion (scaricabile da Google Play).

# Bleb App Inventor – Fase di Test

Ci siamo quasi...

Aggiungi la batteria (RPS) al tuo BLE-B

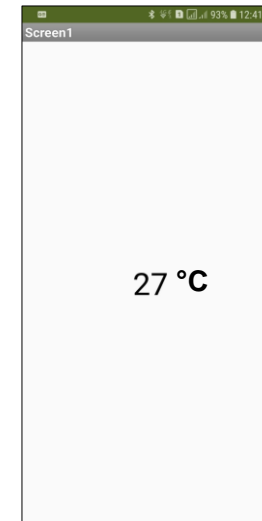


Lancia l'App Mit AI2 sullo Smartphone (debugger or apk) come visto nella pagina precedente.

La tua app è pronta: congratulazioni!

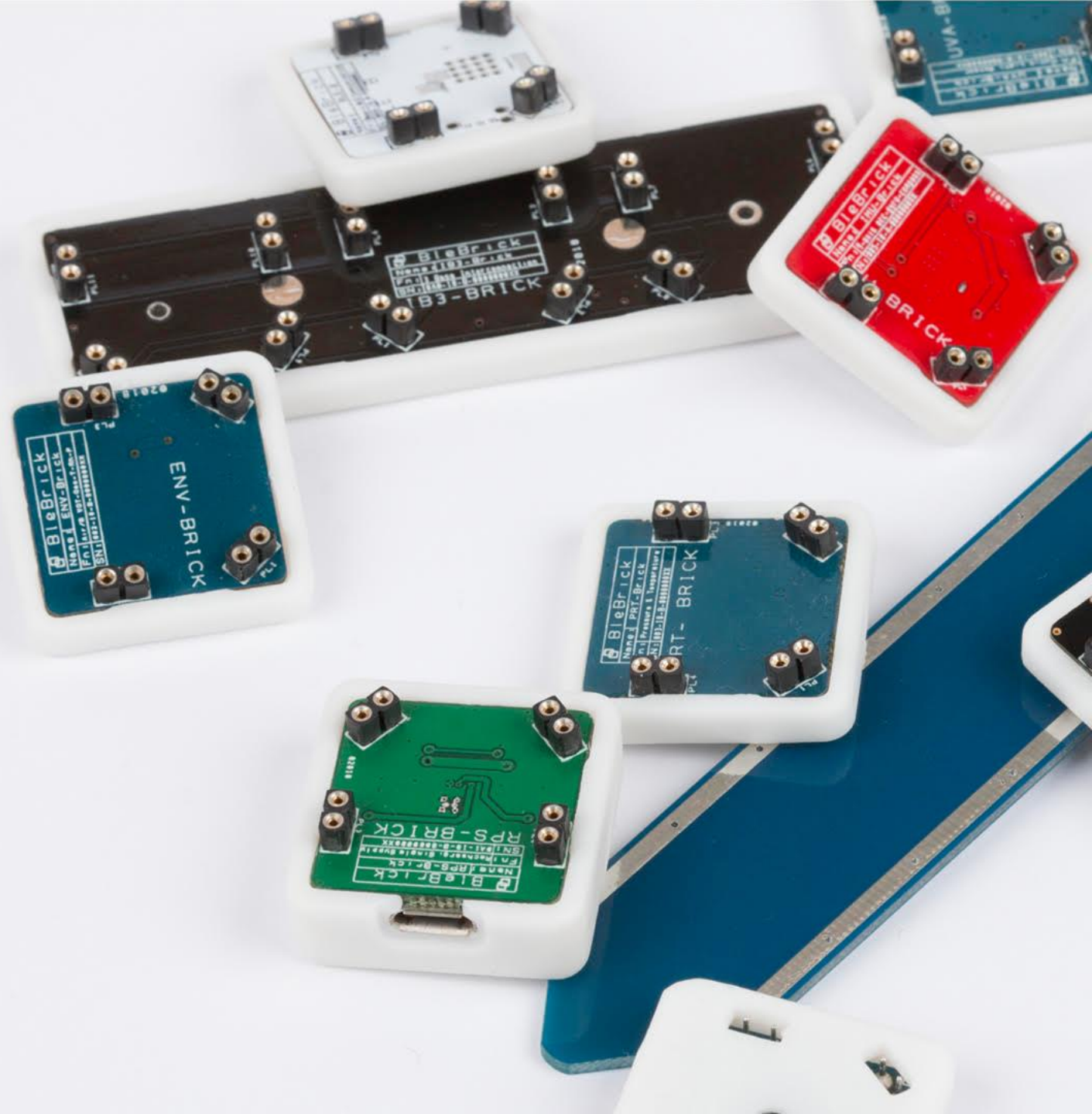


Adesso puoi divertirti a creare app più complesse o diverse, il tutto sempre in maniera molto semplice.  
Questo è tutto!



# Bleb App Inventor

Esempio  
Temperatura e  
pulsante



# Esempio: temperatura e pulsante – Sezione Designer

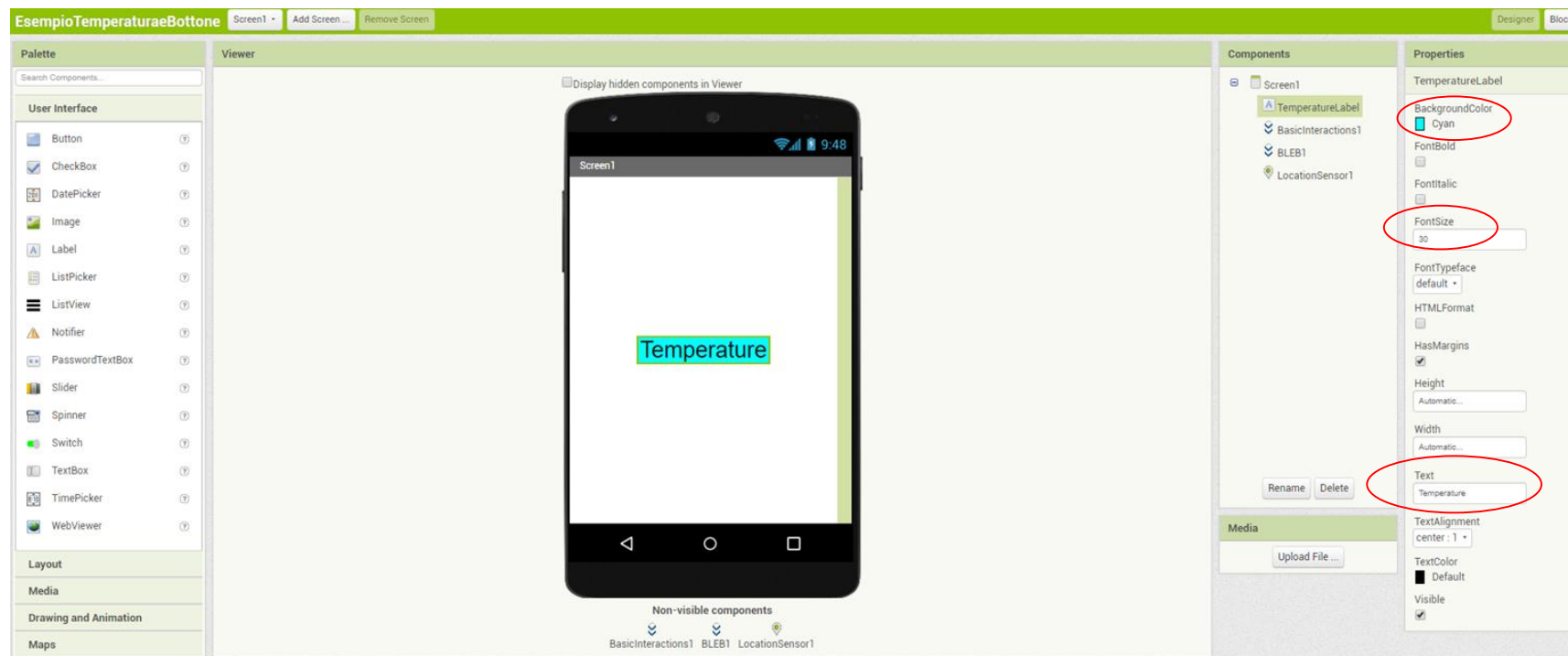
Vediamo come realizzare un'app leggermente più complessa.

Partendo dal precedente esempio, vogliamo leggere anche lo stato del pulsante del BLE-B (se premuto o meno).

1. Rinomina il progetto come “EsempioTemperaturaePulsante”

2. Prova a modificare a piacimento l'interfaccia nella sezione Designer, giusto per divertirti un po' e prendere confidenza, ad esempio:

- in **Components**, seleziona la «TemperatureLabel» e vai su **Properties**
- Modifica il «BackgroundColor» selezionando un colore a piacere, ad esempio «Cyan»
- Cambia il testo della Label scrivendo «Temperature» all'interno del campo «Text»
- Ingrandisci il testo inserendo «30» nel campo «FontSize»

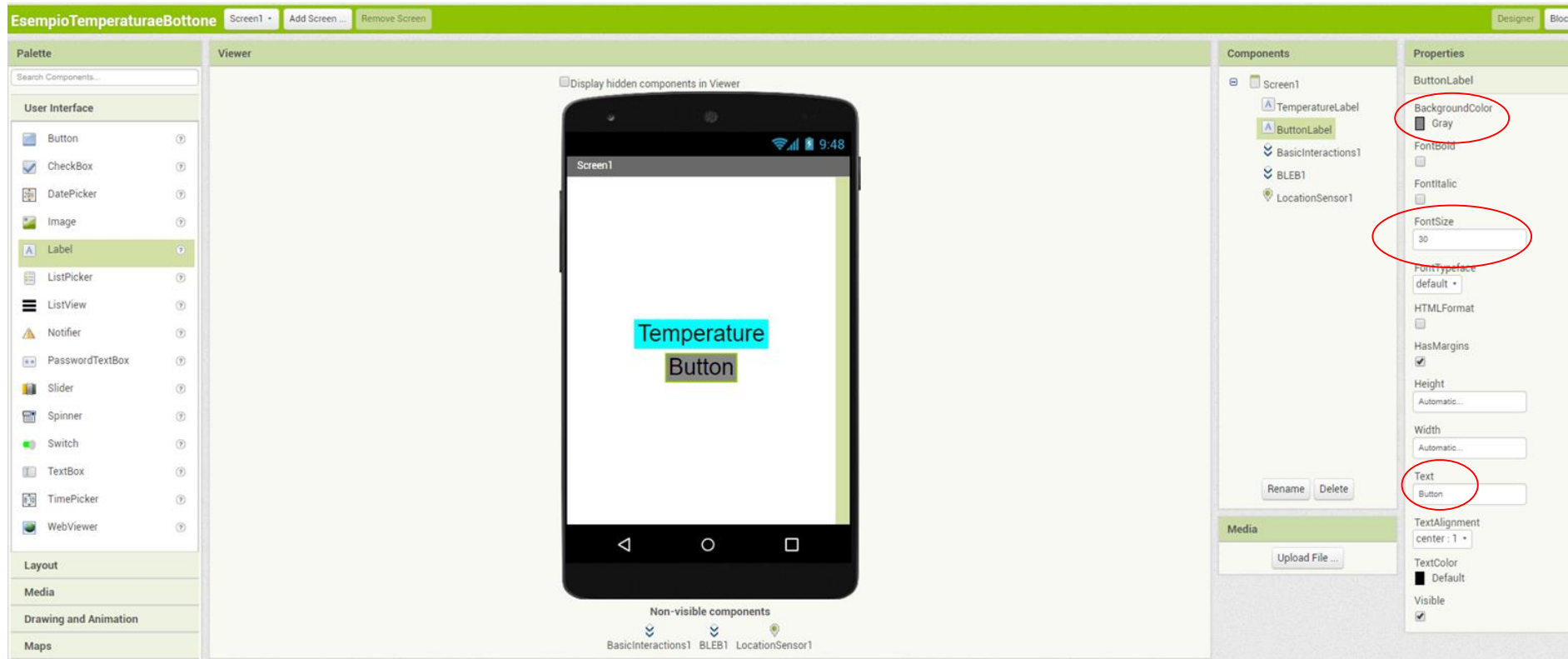


# Esempio: Temperatura e pulsante – Sezione Designer

3. Aggiungi un'altra Label, rinominandola «ButtonLabel»

4. Personalizza la nuova Label come abbiamo appena fatto con la precedente, ad esempio impostando «Gray» come «BackgroundColor», «Button» come «Text» e «30» come «FontSize»

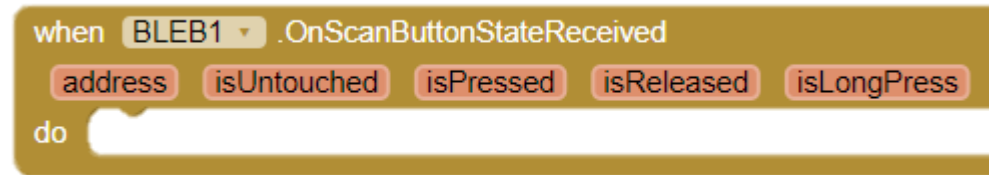
La sezione Designer è completata.



# Esempio: Temperatura e pulsante – Sezione Blocks

Clicca su **Blocks** (in alto a destra) per passare alla parte di programmazione a blocchi in cui definiamo il funzionamento dell'app.

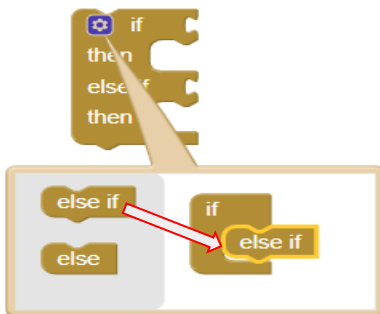
1. Come avevamo fatto nell'esempio precedente col blocco relativo alla temperatura, clicca su «BLEB1» e trascina sullo schermo il blocco che scatta non appena lo smartphone riceve informazioni sul pulsante del BLE-B:



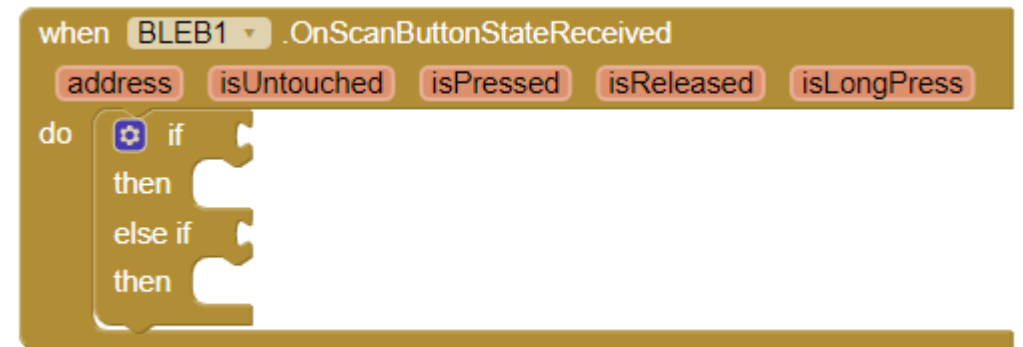
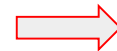
2. Tra i blocchi generici, clicca su «Control» e trascina sullo schermo il blocco condizionale «If»:



3. Espandi il blocco aggiungendo una condizione «else if»; per farlo, clicca sull'icona blu delle impostazioni e trascina il blocco «else if» che comparirà nel rettangolo come mostrato in figura:



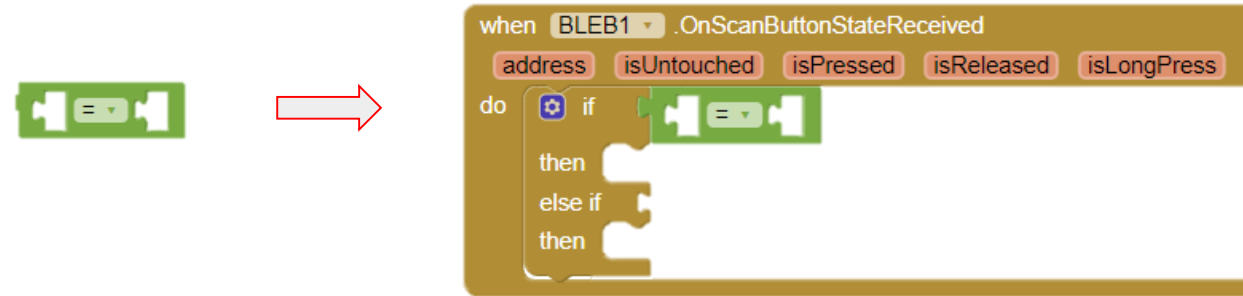
4. Inserisci il blocco «If» all'interno del primo



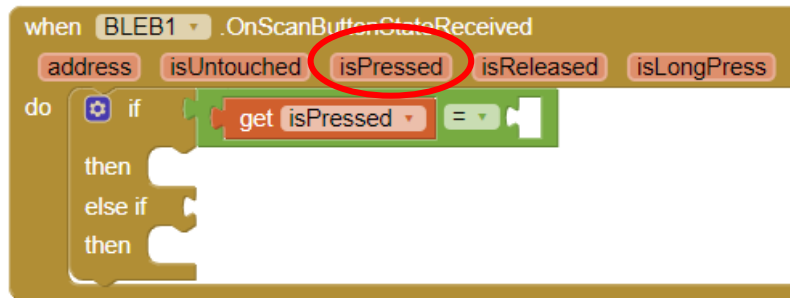


# Esempio: Temperatura e pulsante – Sezione Blocks

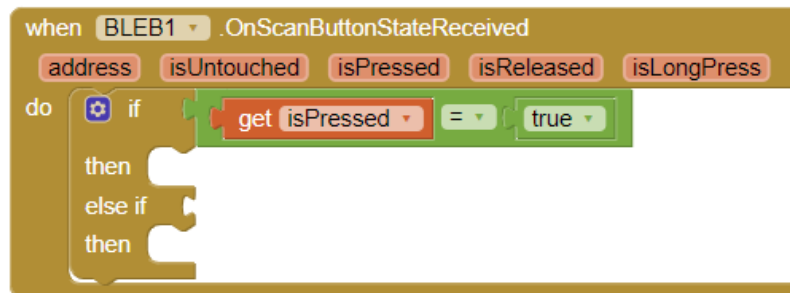
4. Tra i blocchi generici, clicca su «Logic» e collega al primo «If» il blocco che rappresenta l'operatore di uguaglianza (=):



5. Sposta il cursore sopra la variabile «isPressed», seleziona «get isPressed» così da poter leggere (get) se il pulsante del BLE-B è premuto (isPressed sarà vero) oppure no (isPressed sarà falso), e trascinalo in uno dei due spazi del blocco uguaglianza

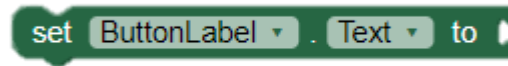


6. Nello spazio rimanente, inserisci il valore «true» (vero) che trovi tra i blocchi generici nella sezione «Logic»:



# Esempio: Temperatura e pulsante – Sezione Blocks

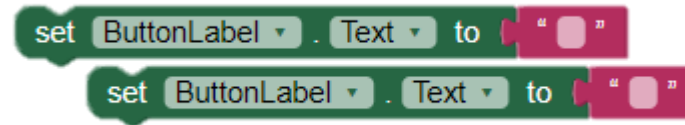
7. Come abbiamo fatto nell'esempio precedente, clicca su «ButtonLabel», seleziona il blocco che permette di impostare il testo della label e connettilo al primo «then» del blocco «if»:



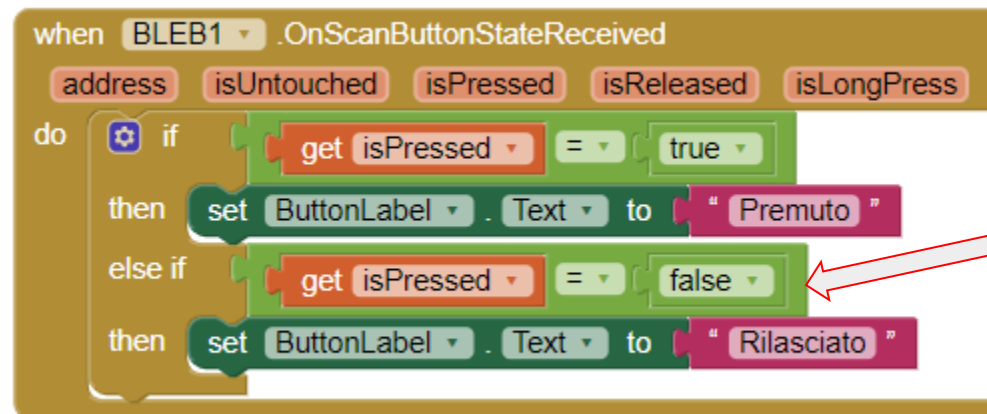
8. Dai blocchi «Text», trascina sullo schermo il blocco che permette di inserire un testo e connettilo al blocco precedente:



9. Clicca col tasto destro sul blocco appena creato e seleziona «Duplicate» per ottenerne una copia:



10. Riempi opportunamente i testi dei due blocchi ed inseriscili all'interno del blocco «if» collegandoli ai due «then»:



Ora dovresti riuscire a creare questo blocco da solo! 😊



# Esempio: Temperatura e pulsante – Sezione Blocks

Riassumendo:

```
when Screen1 .Initialize  
do call BasicInteractions1 .StartScan
```

Questo blocco serve ad **attivare lo scan Bluetooth** sullo smartphone per permettere di rilevare il BLE-B più vicino **appena viene aperta l'app**

```
when BLEB1 .OnScanTemperatureReceived  
address temperature  
do set TemperatureLabel .Text to join get temperature "°C"
```

Questo blocco serve a **scrivere il valore della temperatura misurata dal BLE-B** su una **label** ogni volta che viene ricevuta questa informazione, **aggiungendo anche l'unità di misura**

```
when BLEB1 .OnScanButtonStateReceived  
address isUntouched isPressed isReleased isLongPress  
do if get isPressed = true  
then set ButtonLabel .Text to "Premuto"  
else if get isPressed = false  
then set ButtonLabel .Text to "Rilasciato"
```

Questo blocco serve a **cambiare il testo di una label** in base al **valore dello stato del pulsante del BLE-B** ogni volta che viene ricevuta questa informazione

# Esempio: Temperatura e pulsante - Fase di test



Adesso prova tu a testare la tua app!

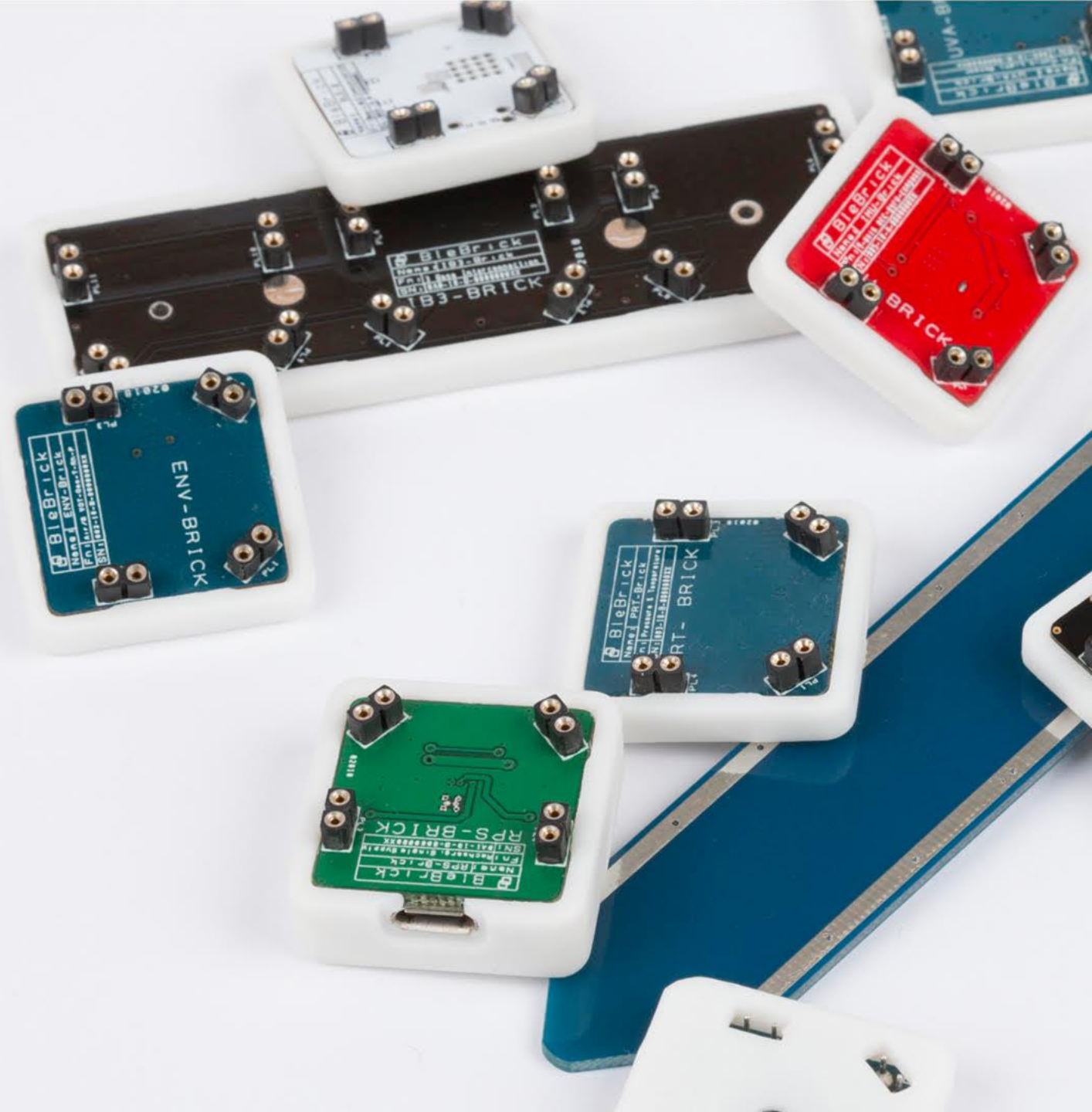
- Genera il QR code e leggilo lanciando l'app MIT AI2 Companion
- Unisci i Blebricks (BLE-B + RPS)
- ...e guardala funzionare sul tuo smartphone!



Help: se hai delle difficoltà in questa fase di test, rileggi la pagina 11.

# Bleb App Inventor

Esempio  
sensore IMU



# Esempio: Sensore IMU

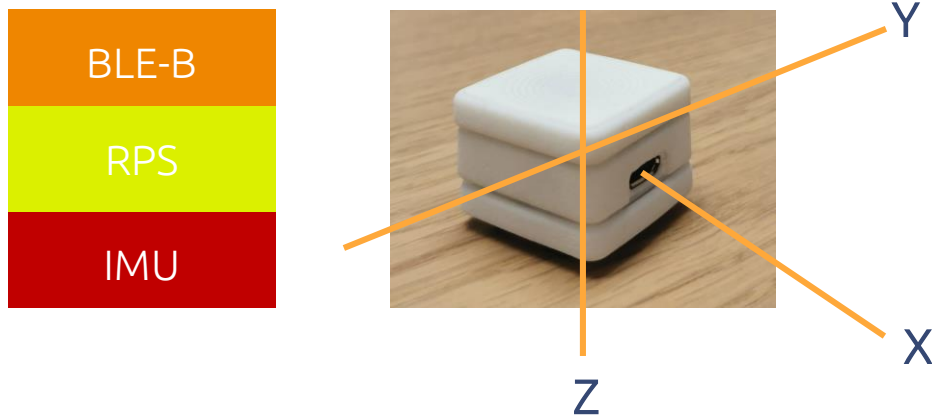
Iniziamo a vedere degli esempi dove oltre al BLE-B, utilizziamo sensori e attuatori, il tutto sempre in modo molto semplice. Iniziamo dal sensore IMU, **ma prima di vedere come si programma**, spieghiamo brevemente il suo funzionamento.

Il sensore IMU può funzionare in 3 modalità:

1. come **Accelerometro** (modalità di default) misura l'accelerazione sui tre assi X,Y e Z
2. come **Magnetometro** misura il campo magnetico sui tre assi X,Y e Z
3. come **Giroscopio** misura il suo orientamento nello spazio, traducendolo in angoli (rollio, beccheggio e imbardata)

Quindi, in base alla modalità che sceglierai, potrai visualizzare solo i relativi dati: ad esempio, configurando l'IMU in modalità giroscopio non potrai leggere dati sull'accelerazione o sul campo magnetico!

**N.B.** L'IMU rimarrà nella modalità in cui lo avrai configurato, anche dopo averlo spento e riaccessso oppure dopo aver chiuso e riaperto l'app!



# Esempio: Sensore IMU – Sezione Designer

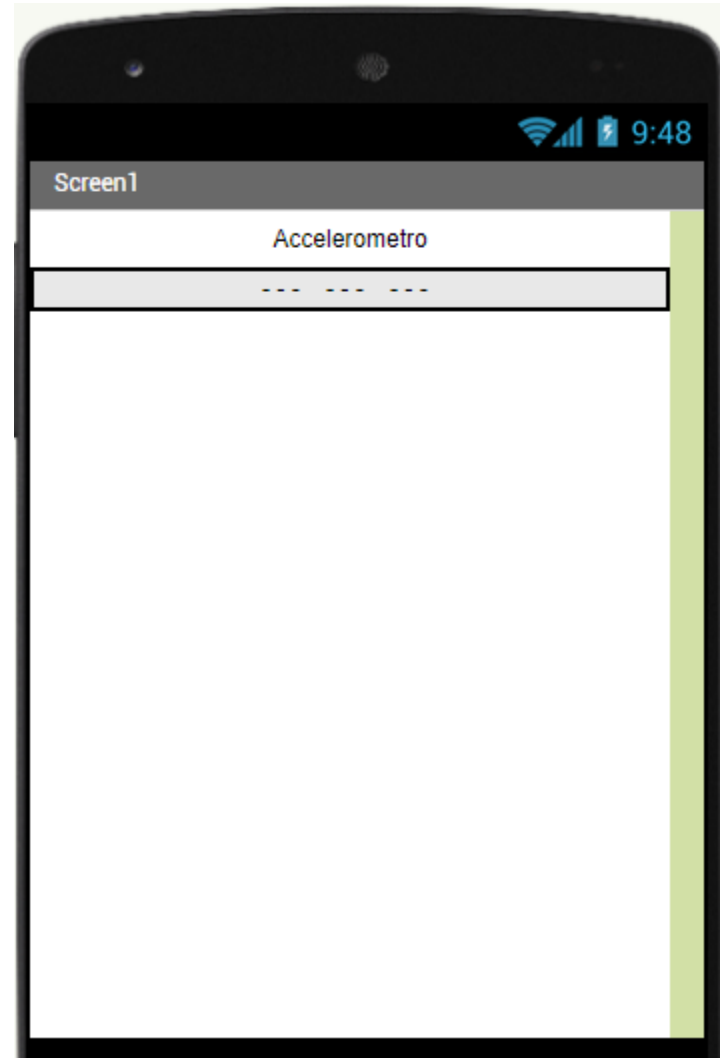
Partiamo col nostro esempio: **Sensore IMU**

1. Apri un nuovo progetto in MIT App Inventor 2, rinominalo, salvalo come **IMUEsempio** e importa le nostre estensioni.
2. Nella sezione **Designer**, segui le indicazioni della tabella qua sotto, che mostra quali blocchi dovrai trascinare sullo schermo, dove trovarli e come eventualmente modificarli:

Component Type	Palette	Rename it to....	Purpose	Change Properties
LocationSensor	Sensors		Necessario al funzionamento del Bluetooth Low Energy	
BasicInteractions	Extension		Necessario al funzionamento dei Blebricks	
BLEIMU	Extension		Necessario al funzionamento del Blebrick IMU	
Label	User Interface	ModeLabel	Mostra la modalità dell'IMU	Width = Fill parent Text = Accelerometro TextAlignment = Center
HorizontalArrangement	Layout	IMUFrame	Organizza le Labels dell'IMU	AlignHorizontal = Center Width = Fill parent
Label	User Interface	XValueLabel	Mostra il valore dell'asse X	Text = - - -
Label	User Interface	YValueLabel	Mostra il valore dell'asse Y	Text = - - -
Label	User Interface	ZValueLabel	Mostra il valore dell'asse Z	Text = - - -

# Esempio: Sensore IMU – Sezione Designer

Una volta fatto, dovrebbe risultarti una schermata simile a questa:

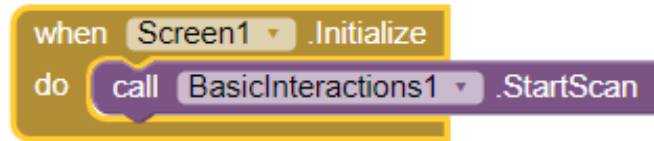


Abbiamo finito con la parte di **Designer**, clicca in alto a sinistra per passare alla sezione **Blocks**!



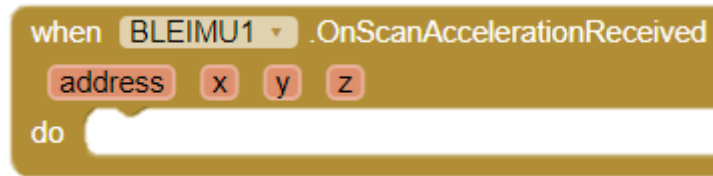
# Esempio: Sensore IMU – Sezione Blocks

1. Clicca su «Screen1» e trascina sullo schermo il blocco che scatta non appena viene aperta l'app, e poi da «BasicInteractions1» prendi il blocco che attiva lo scan Bluetooth sullo smartphone:

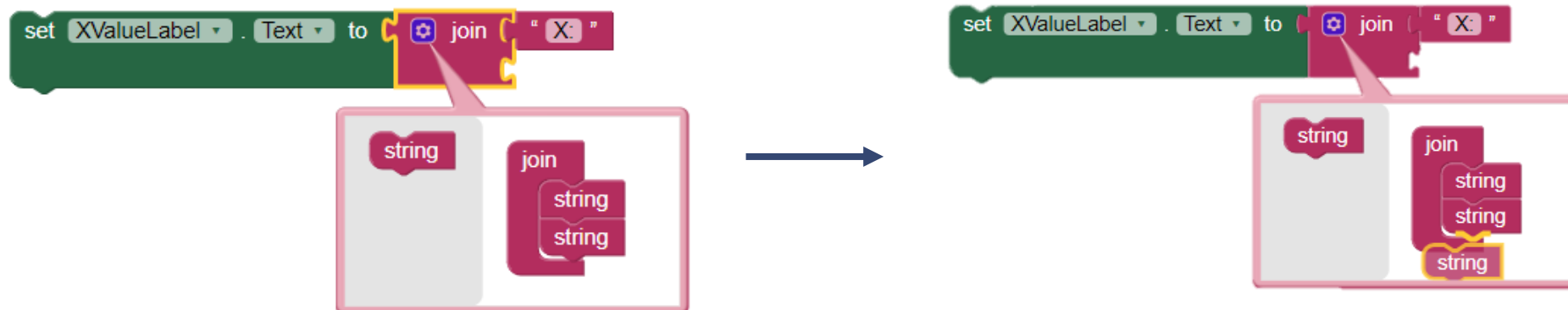


Il blocco che abbiamo così realizzato serve quindi ad **attivare lo scan Bluetooth sullo smartphone** per permettere di rilevare il BLE-B più vicino **appena viene aperta l'app**

2. Clicca su «BLEIMU1» e trascina sullo schermo il blocco che scatta non appena lo smartphone riceve informazioni sull'accelerazione misurata dall'IMU:

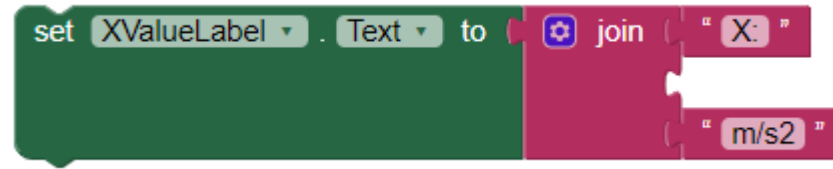


3. Come mostrato negli esempi precedenti, andiamo a creare un blocco che scrive un valore su una Label, agganciandoci però tre blocchi di testo questa volta:

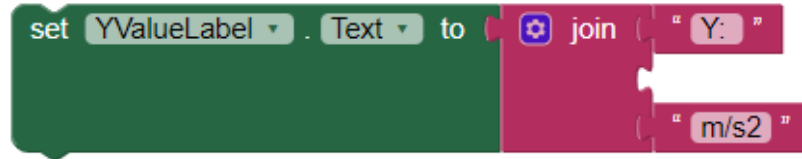


# Esempio: Sensore IMU – Sezione Blocks

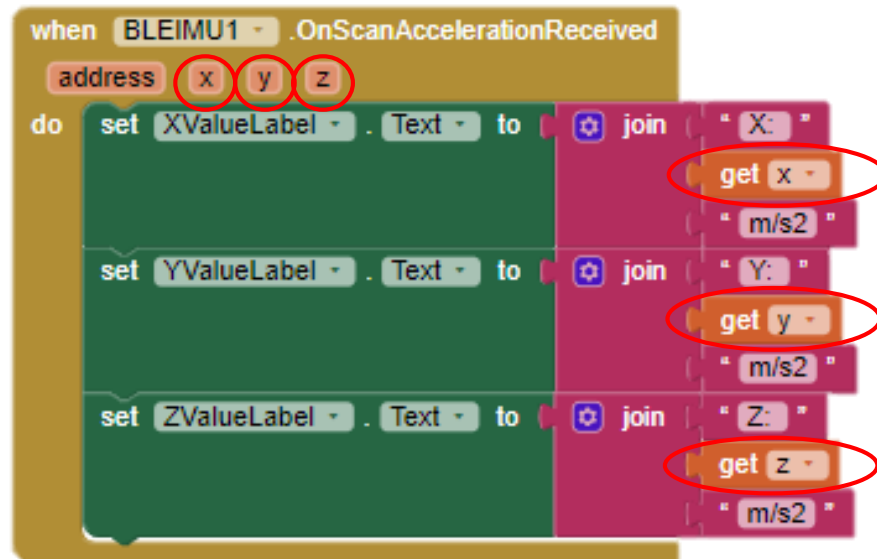
4. Nel terzo blocco scriviamo  $m/s^2$ , che rappresenta l'unità di misura dell'accelerazione ( $m/s^2$ ):



5. Andiamo a ripetere l'operazione per gli altri due assi X e Y (scorciatoia: prova a copiare il blocco e modificarlo!):



6. Posizioniamo i tre blocchi come in figura, trascinando i blocchi con i valori delle variabili «x», «y» e «z»:



Suggerimento: per ottenere i blocchi «get ...» sposta il cursore sulle variabili interne al blocco (qui address, x, y e z)

# Esempio: Sensore IMU – Sezione Blocks

Riassumendo:

```
when Screen1 Initialize
do call BasicInteractions1 StartScan
```



Questo blocco serve ad **attivare lo scan Bluetooth sullo smartphone** per permettere di rilevare il BLE-B più vicino **appena viene aperta l'app**

```
when BLEIMU1 OnScanAccelerationReceived
address x y z
do set XValueLabel Text to join "X: " get x " m/s2"
set YValueLabel Text to join "Y: " get y " m/s2"
set ZValueLabel Text to join "Z: " get z " m/s2"
```



Questo blocco serve a **scrivere il valore dell'accelerazione sui tre assi misurata dal BLE-B sulle label** ogni volta che viene ricevuta questa informazione, **aggiungendo anche il nome dell'asse e l'unità di misura**

Adesso, passiamo a testare la tua App!



# Esempio: Sensore IMU – Fase di test



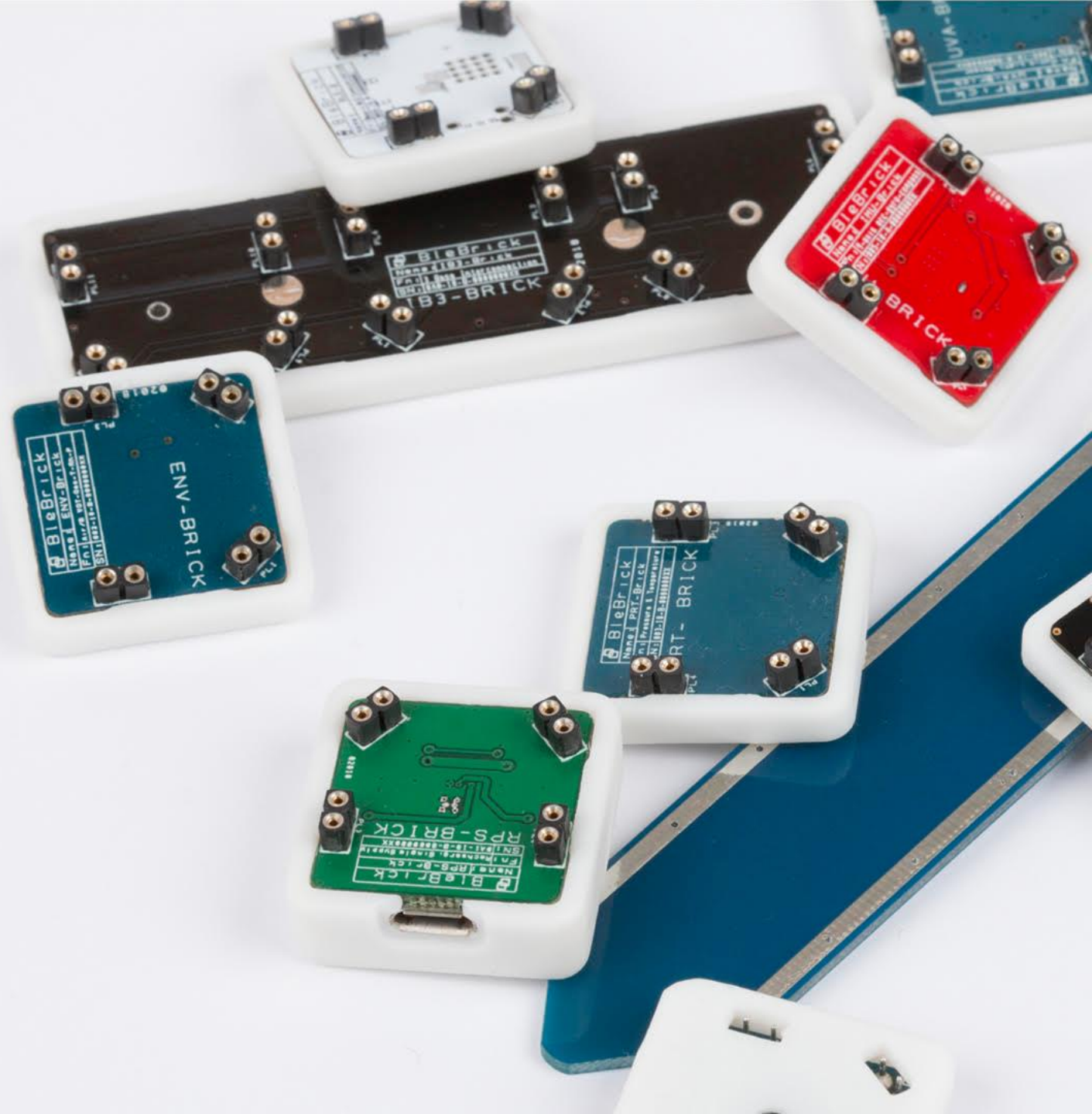
Adesso prova tu a testare la tua app!

- Genera il QR code e leggilo lanciando l'app MIT AI2 Companion
- Unisci i Blebricks (BLE-B + IMU + RPS)
- ...e guardala funzionare sul tuo smartphone!

Help: se hai delle difficoltà in questa fase di test, rileggi la pagina 11.

# Bleb App Inventor

Esempio  
allarme con  
sensore IMU



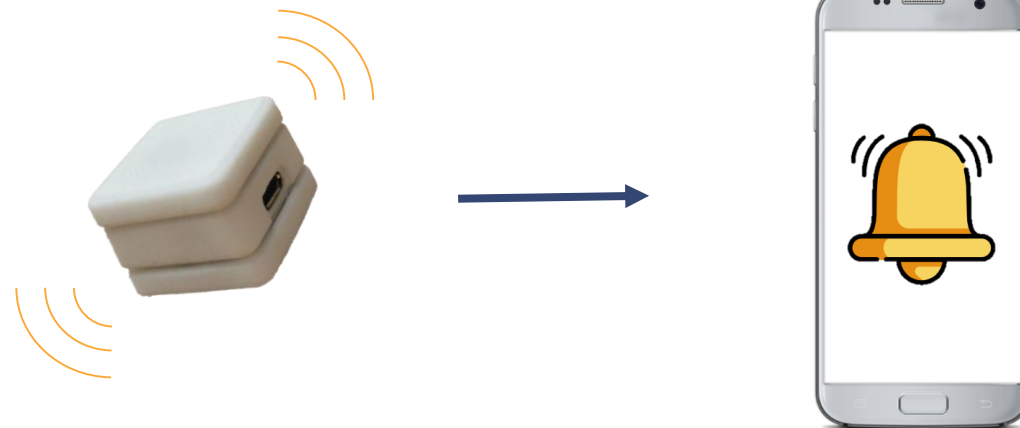
# Esempio: Allarme con sensore IMU

Vediamo come, continuando dal precedente esempio, possiamo creare un allarme in maniera semplice, sfruttando sempre il sensore IMU.

Il **nostro obiettivo** sarà quindi quello di rilevare un'accelerazione col nostro sensore IMU (posizionato ad esempio sotto il sellino della bicicletta) e compararla con una soglia da noi prefissata: quando l'accelerazione supererà tale soglia, la nostra app farà partire un allarme!

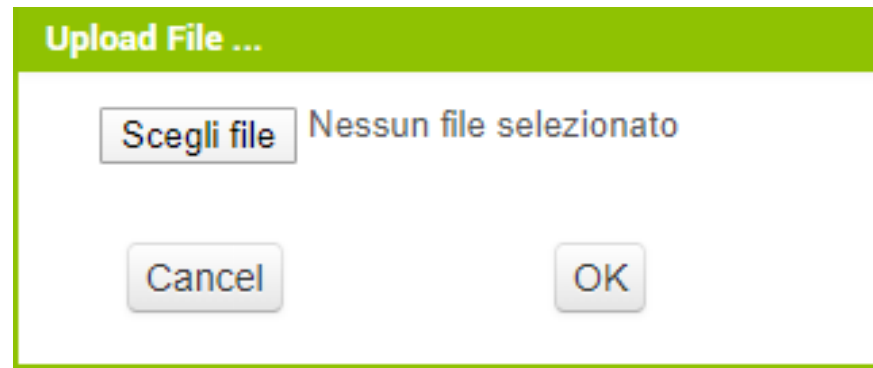
Questa applicazione può trovare utilizzo come antifurto, ma anche come avviso in caso di caduta di una persona anziana o in tanti altri casi, sta a te pensare a quello più adatto!

Nel frattempo, **vediamo come realizzarla!**



# Esempio: Allarme con sensore IMU – Sezione Designer

1. Nella sezione **Designer**, clicca su «Media» e trascina «Sound» sullo schermo (risulterà come un componente non visibile).
2. Clicca sul blocco «Sound» e poi spostati su «Properties»
3. Clicca su «Source» → «Upload File» → «Scegli file» per caricare un file audio che hai sul computer e che sarà il nostro allarme.



Abbiamo finito con la parte di **Designer**, clicca in alto a sinistra per passare alla sezione **Blocks**!

# Esempio: Allarme con sensore IMU – Sezione Blocks

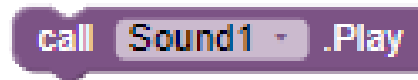
1. Andiamo a definire la condizione di allarme, prendendo alcuni blocchi dal gruppo «Math», tra i blocchi generici:



2. Componi i blocchi come mostrato in figura:



3. Clicca su «Sound1» e trascina sullo schermo il blocco che riproduce il suono da noi selezionato:





# Esempio: Allarme con sensore IMU – Sezione Blocks

4. Tra i blocchi generici, clicca su «Control», trascina sullo schermo il blocco condizionale «If», e aggiungilo al blocco principale insieme agli altri:

```
when BLEIMU1.OnScanAccelerationReceived
  address x y z
do
  set XValueLabel.Text to join " X: "
  get x
  " m/s2 "
  set YValueLabel.Text to join " Y: "
  get y
  " m/s2 "
  set ZValueLabel.Text to join " Z: "
  get z
  " m/s2 "
  if (get x ^ 2 + get y ^ 2 + get z ^ 2) >= 140
  then call Sound1.Play
```



Oltre a quanto detto prima, se la norma dei valori di accelerazione sui tre assi è maggiore o uguale di una certa soglia, questo blocco riproduce un suono

La norma è una funzione matematica che in questo caso serve a misurare l'accelerazione indipendentemente dalla sua direzione o verso: non sappiamo come l'IMU verrà mosso!

Noi abbiamo messo la soglia a 140, ma naturalmente puoi cambiarla per adattarla al tuo caso!

# Esempio: Sensore IMU – Sezione Blocks

Riassumendo:

```
when Screen1.Initialize
do call BasicInteractions1.StartScan
```

Questo blocco serve ad **attivare lo scan Bluetooth** sullo **smartphone** per permettere di rilevare il BLE-B più vicino **appena viene aperta l'app**

```
when BLEIMU1.OnScanAccelerationReceived
address x y z
do
  set XValueLabel.Text to join ["X: ", get x, " m/s2"]
  set YValueLabel.Text to join ["Y: ", get y, " m/s2"]
  set ZValueLabel.Text to join ["Z: ", get z, " m/s2"]
  if (get x ^ 2 + get y ^ 2 + get z ^ 2) >= 140
  then call Sound1.Play
```

Questo blocco serve a **scrivere il valore dell'accelerazione** sui tre assi misurata dal BLE-B sulle **label** ogni volta che viene ricevuta questa informazione, **aggiungendo anche il nome dell'asse e l'unità di misura**; inoltre, se la **norma dei valori di accelerazione sui tre assi è maggiore o uguale di una certa soglia**, riproduce un suono



Questo è tutto:  
passiamo alla fase di test!

# Esempio: Sensore IMU – Fase di test



Adesso prova tu a testare la tua app!

- Genera il QR code e leggilo lanciando l'app MIT AI2 Companion
- Unisci i Blebricks (BLE-B + IMU + RPS)
- Prova ad agitare il sensore e verifica che l'allarme scatti! Se non succede, prova ad abbassare il valore di soglia!

Help: se hai delle difficoltà in questa fase di test, rileggi la pagina 11.