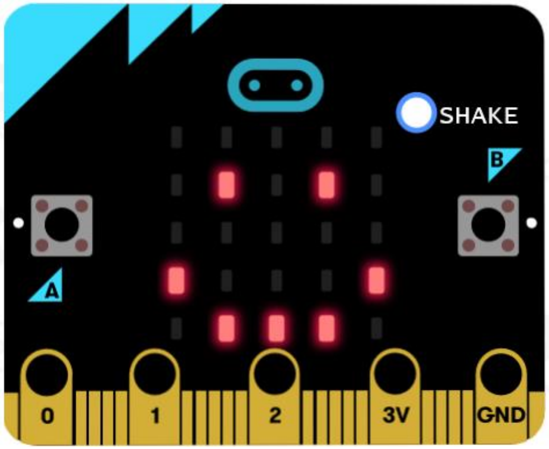


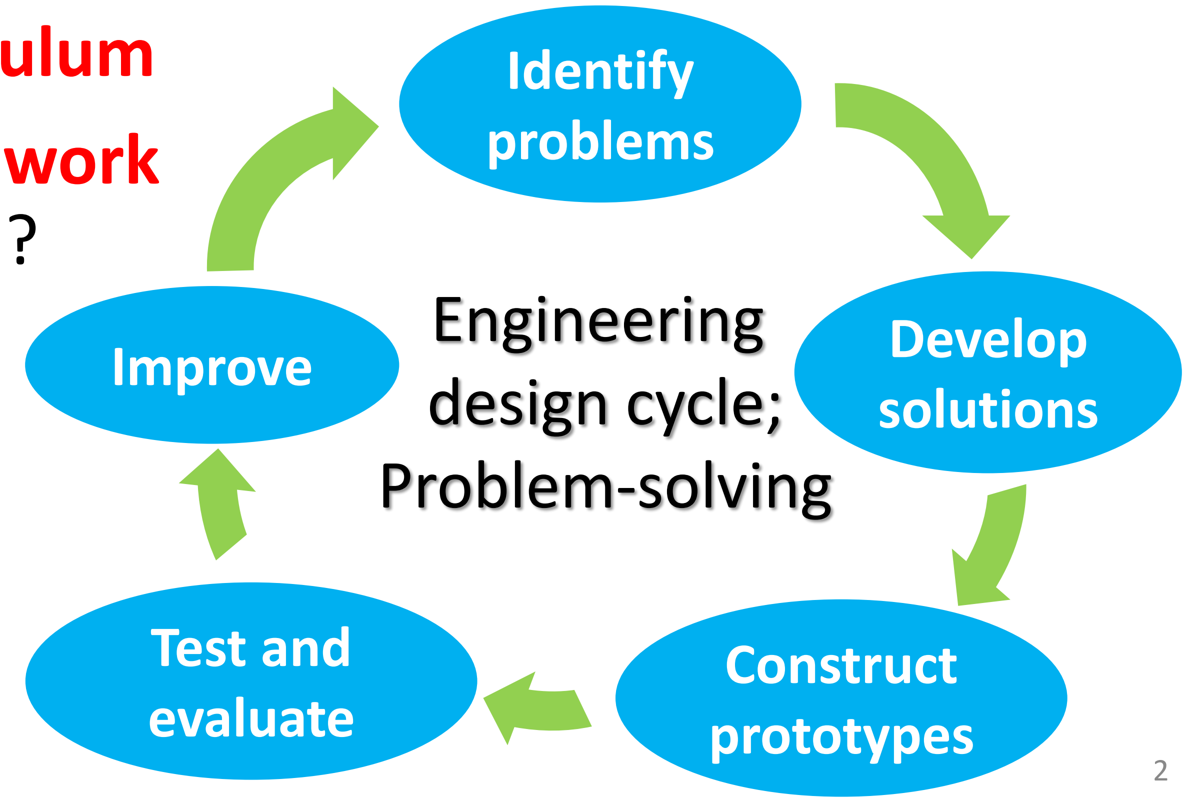
micro:bit and App Inventor for STEM

Dr YEUNG Chi Ho Bill

The Department of Science and Environmental Studies
The Education University of Hong Kong

Difficulties in integrating STEM in school-based curricula?

- **Integration** of STEM in single subject? Across subjects?
- Cross-disciplinary **coordination of teaching time** of topics to be included in each STEM activities?
- **No universal curriculum**
- **Free-style** vs **framework** in STEM curriculum?
- **Progression** across levels?
- STEM in **junior primary** levels?
- **Assessment**?



Identify problems

Develop solutions

Construct prototypes

Test and evaluate

Improve

Engineering design cycle; Problem-solving

Learning objectives of STEM

- Before designing STEM activities, one can plan ahead the **learning objectives of "STEM":**

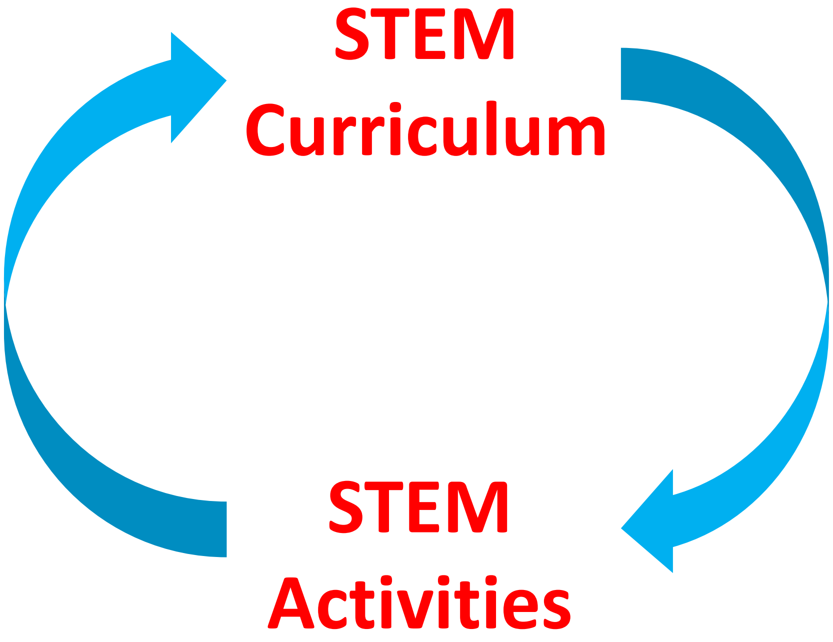
Six levels of cognitive processes in the revised Bloom's taxonomy

		S	T	E	M	___
lower-order thinking	Remember (記憶)					
	Understanding (理解)					
	Apply (應用)					
higher-order thinking	Analyze (分析)					
	Evaluate (評鑑)					
	Create (創造)					

Reference: Dr LEE Yeung Chung

How are STEM curriculum and STEM activities inter-connected?

Generic objectives of S, T, E, M



STEM Curriculum

Complete the objectives

Set up the objectives

STEM Activities

Activity-specific objectives
Connect the generic objectives of S, T, E, M

Reference:
Dr LEE Yeung Chung

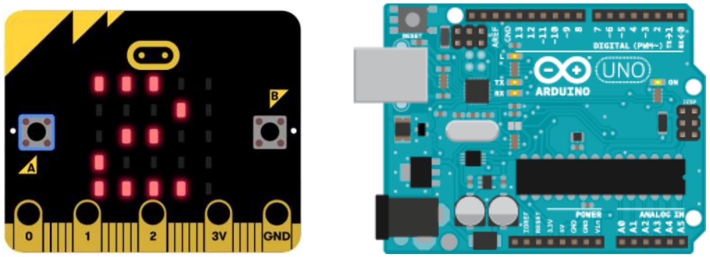
Tools and platforms

- Various **electronic tools, platforms** and **micro-controllers** are for **STEM education**, suitable for students at various levels

Construction toys

Education robots

- Demonstrate the integration between **design** and **coding**

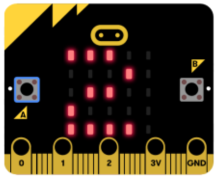


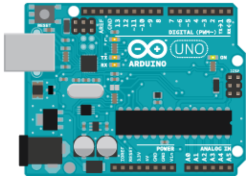
Micro-controllers

- But please note that:

1. These tools are only **platforms**, they are **not equivalent to STEM education**
2. **STEM education can be conducted without this kind of tools**
3. The introduced tools are **only a subset** of all available tools, there are other suitable tools

Overview of tools





MIT APP INVENTOR

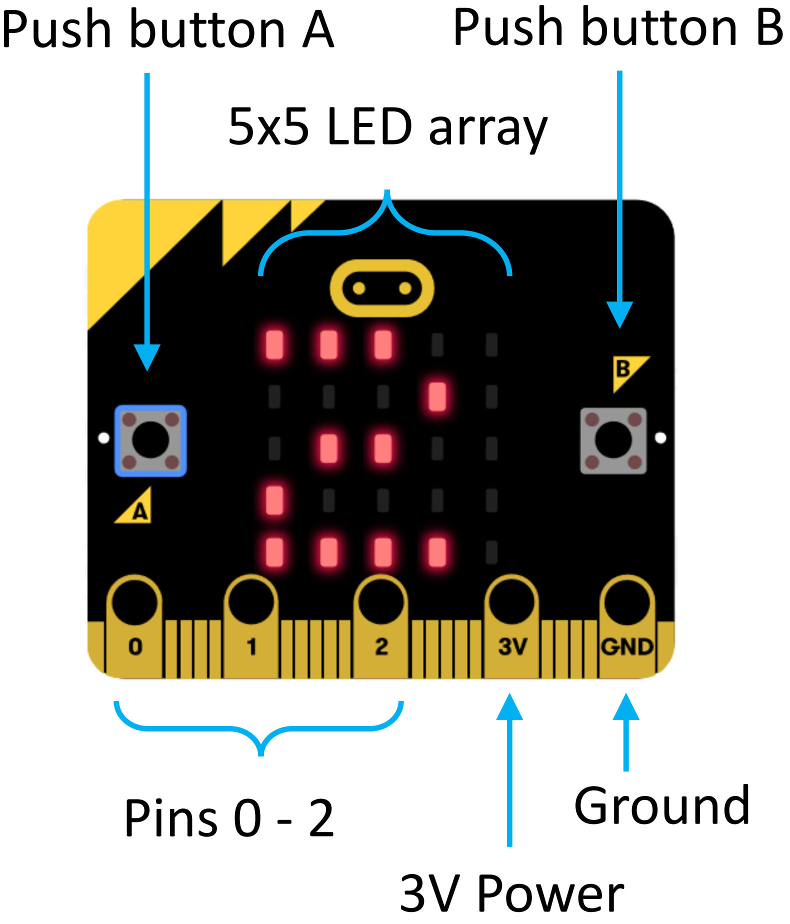
	littleBits	mBot	micro:bit	Arduino	App inventor
Developer	Ayah Bdeir USA, 2011	MakeBlock, China, 2011	BBC UK, 2015	Open source Italy, 2003	Google & MIT, 2010
Suitability					
Circuit connection	Magnetic electronic building blocks	Plug and play	Crocodile wire /breadboard with jumpers	Breadboard with jumpers	No (can join micro:bit or Arduino)
Circuit knowledge required?					
Coding required?					
Coding languages					
Price					

Tool 1: micro:bit

micro:bit

- micro:bit is a micro-controller designed by **BBC** for **computer and coding education**
- It has

1. two control buttons
2. one reset button
3. one 5x5 LED display array
4. one 3-axis accelerometer
5. one 3-axis magnetometer
6. a 2.4GHz radio module (for Bluetooth communication)



Advantages of using micro:bit?

- **Research findings** on the use of micro:bits:

1. 90% of students said the micro:bit showed them that anyone can code.
2. 86% of students said the micro:bit made Computer Science more interesting.
3. 70% more girls said they would choose Computing as a school subject after using the micro:bit.
4. 85% of teachers agree it has made ICT/Computer Science more enjoyable for their students.
5. Half of teachers who've used the micro:bit say they now feel more confident as a teacher, particularly those who say they're not very confident in teaching Computing.

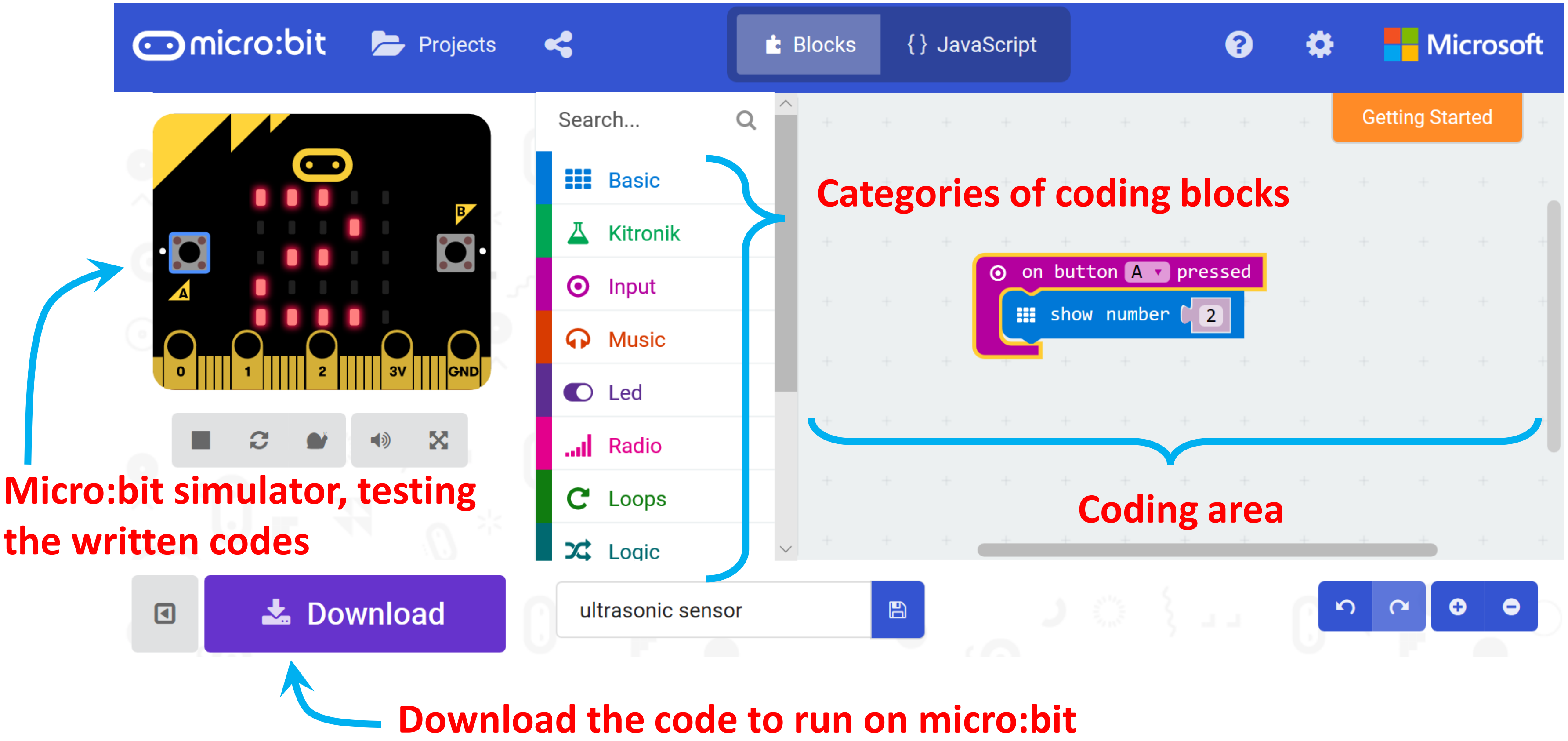
Reference:

[1] "Creating cool stuff" – Pupils' experience of the BBC micro:bit, Proceedings of the 48th ACM Technical Symposium on Computer Science Education: SIGCSE 2017, Sentance, S., Waite, J., Hodges, S., MacLeod, E., & Yeomans, L. E. (2017)

[2] Microbit's website: http://microbit.org/teach/, retrieved on 14th Sept, 2017

Part A - Coding with micro:bit

- micro:bit is most easily connected to **computers** through **a USB cable** through its micro-USB connector
- By going to http://microbit.org/code/, and choose **let's code**, we arrive at the coding area:



Categories of coding blocks

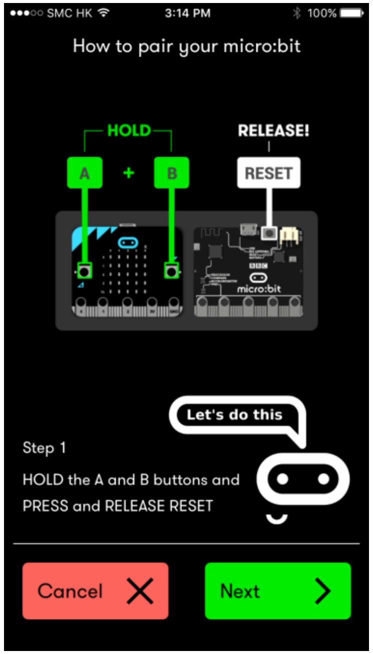
Coding area

Micro:bit simulator, testing the written codes

Download the code to run on micro:bit

Control via Mobile Devices

- If computers are not available, one can code the micro:bit with **mobile devices**
- First, install the app "**micro:bit**" on the mobile devices
- Follow the instructions on http://microbit.org/guide/mobile/

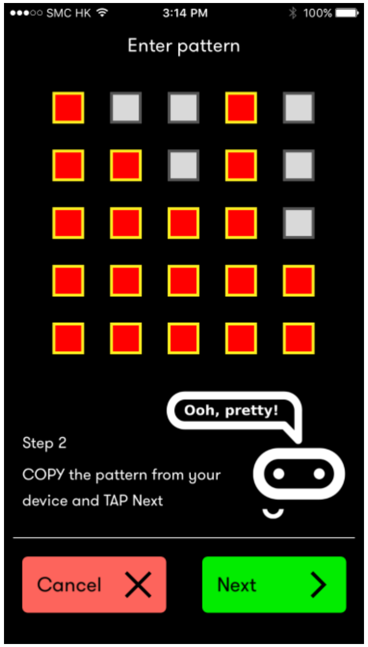


How to pair your micro:bit

Step 1

HOLD the A and B buttons and PRESS and RELEASE RESET

1. HOLD A & B bottons, at the same time PRESS and RELEASE RESET



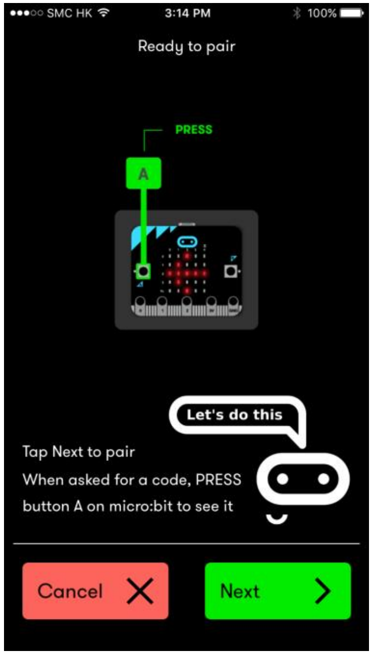
Enter pattern

🟥⬜⬜🟥⬜
🟥🟥⬜🟥⬜
🟥🟥🟥🟥⬜
🟥🟥🟥🟥🟥
🟥🟥🟥🟥🟥

Step 2

COPY the pattern from your device and TAP Next

2. COPY the pattern from your device



Ready to pair

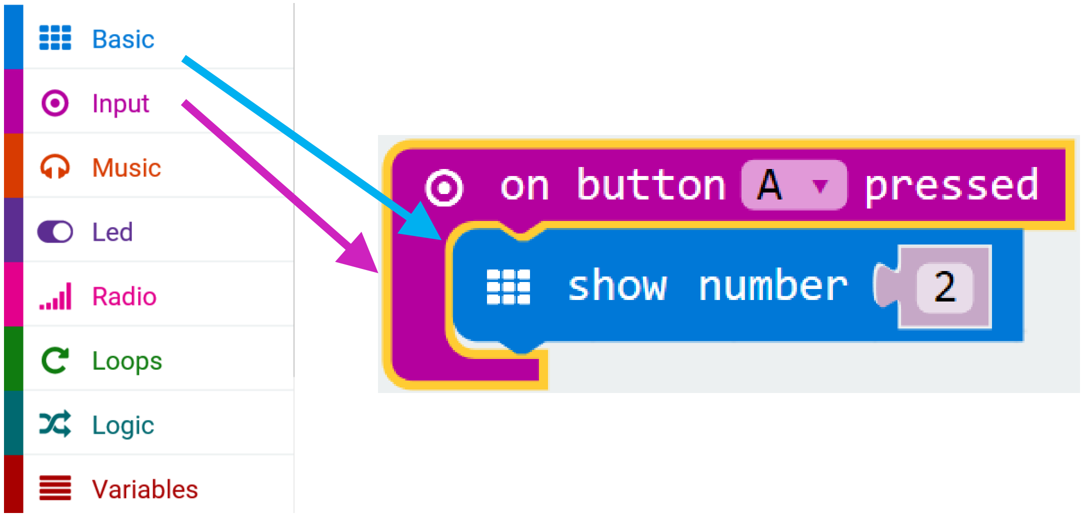
Tap Next to pair

When asked for a code, PRESS button A on micro:bit to see it

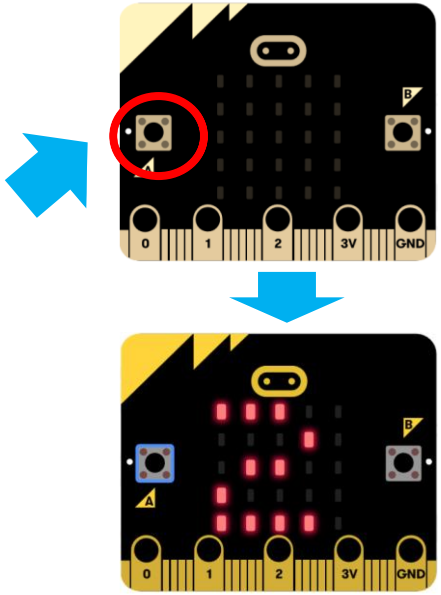
3. When asked a code, PRESS A to see it

Part B - Basic Operations

- Open a **browser**, go to http://microbit.org/code/
- Click "**Let's code**" under "JavaScript Blocks Editor"
- **Coding blocks** are grouped into various categories
- Input the following codes:



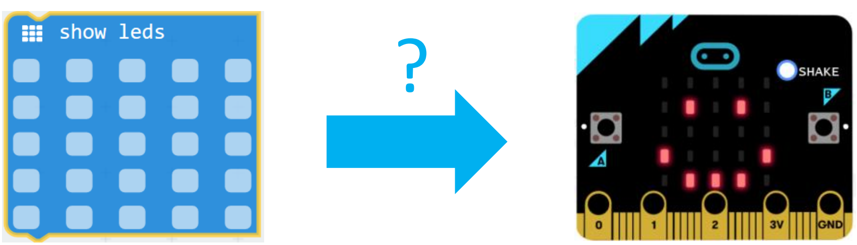
```
on button A pressed
    show number 2
```

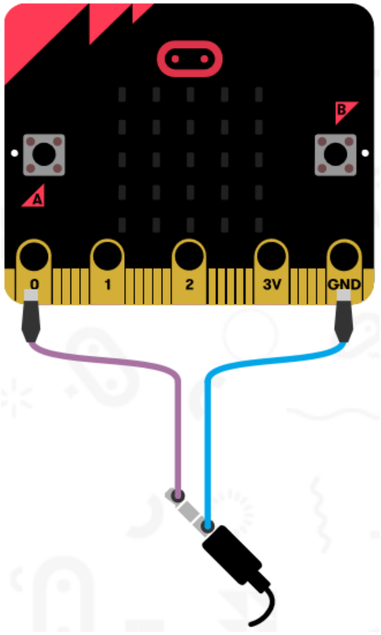


- **Connect** the micro:bit to the computer using a **USB cable**
- Click "**Download**" and save the file in the "MICROBIT (D:)" (micro:bit may be on a different drive in your computer)
- **Press the "A" button** on the micro:bit

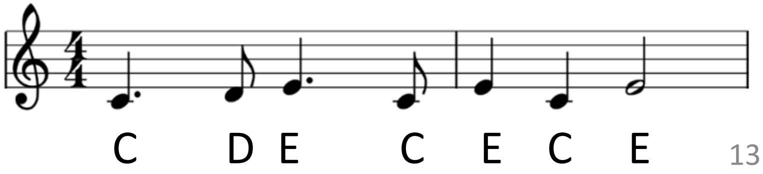
Build-in LED and Music blocks

- **STEAM education** – micro:bit is also a good tool for STEAM (STEM+ART) education since we can display artworks on the LED array, or code to produce music
- Exercise 1: use “**Basic**” → “**show leds**” block to code the micro:bit to display a smile for 1 second when it is shaked



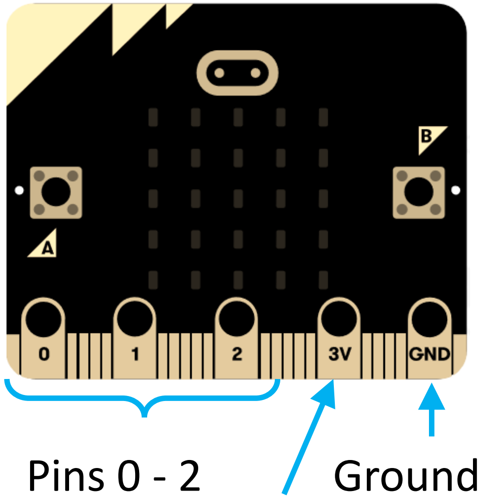


- To produce sound from micro:bit, one can **connect the micro:bit with speakers or earphones** as shown in the right figure
- Exercise 2: use the blocks in the “**Music**” category to code the melody “Do-Re-Me” in “Sound of Music”



Part C - External Output Devices

- We can **connect** the micro:bit with **other components** using crocodile-clip wires, and the different pins on the micro:bit

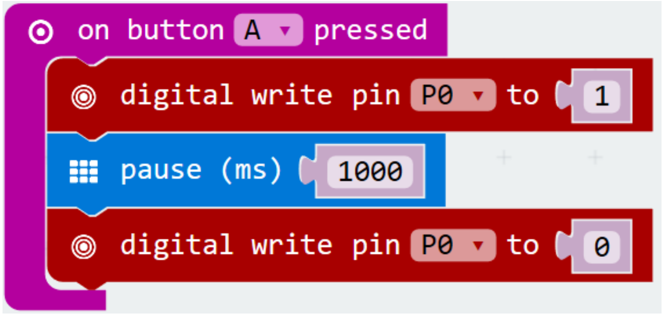


Pins 0 - 2

Ground

3V Power

- Example: connect an LED across pin 0 and GND, and use "**Pin**" and this code to turn it on:



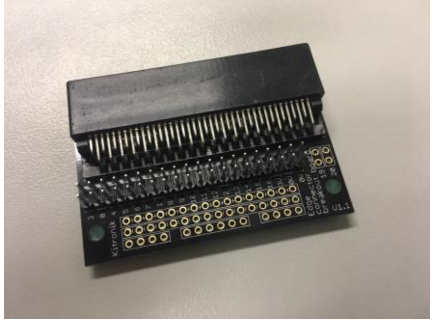
```
on button A pressed
    digital write pin P0 to 1
    pause (ms) 1000
    digital write pin P0 to 0
```

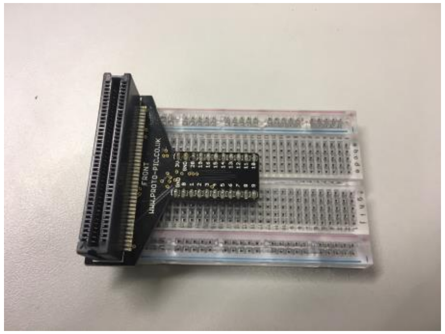
- Exercise 1: use one green and one red LED, and to simulate **traffic light for the pedestrian**

- Exercise 2: use the LED array on the micro:bit to **count down the green and red signals** as in real traffic light
- Exercise 3: add buzzer to **remind pedestrian with sounds**

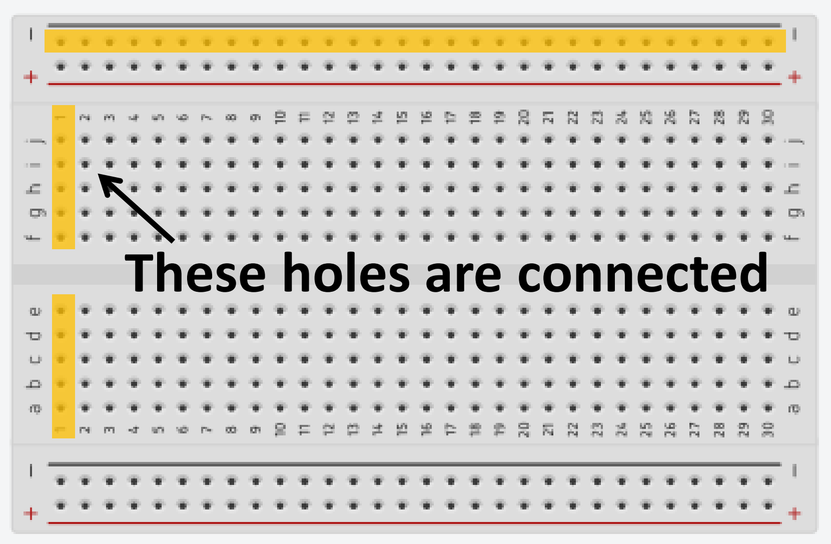
Edge connectors and breadboard

- **Edge connectors** can extend the pins on the micro:bit such that jumper wires can be used.





- **Breadboard** can also be used together with edge connectors to wire more complex circuits (wholes highlighted in **the same row/column are connected**)



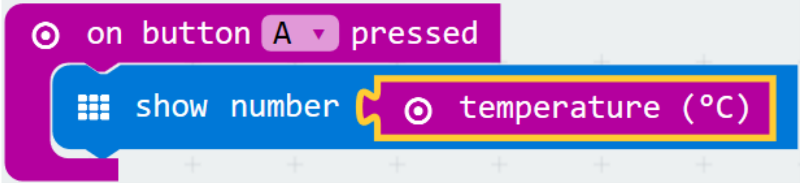
These holes are connected

- Exercise 4: use an edge connection and a breadboard to connect a circuit to simulate the traffic light for vehicles

Part D – Sensors: Build-in Sensors

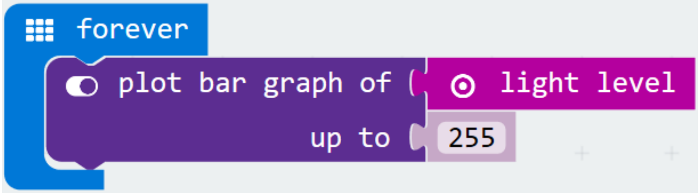
- One important function of smart devices is to **read data from sensors and react with actions**
- There are **several build-in sensors** in micro:bit where readings can be read directly (in the “**Input**” catergory):

Temperature sensor:



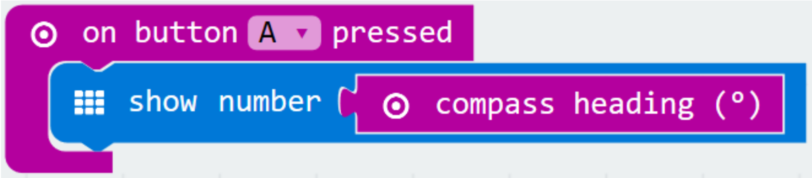
```
on button A pressed
    show number temperature (°C)
```

Light sensor (0-255):



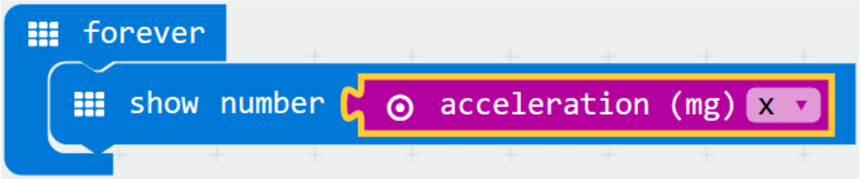
```
forever
    plot bar graph of light level
                up to 255
```

Compass reading:



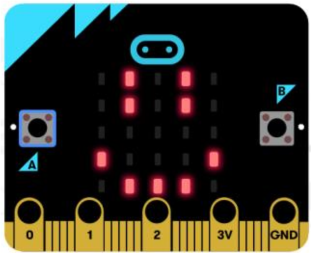
```
on button A pressed
    show number compass heading (°)
```

Accelerometer reading:



```
forever
    show number acceleration (mg) x
```

Move the micro:bit in a circle, until a smiley face appears, before compass reading can be read

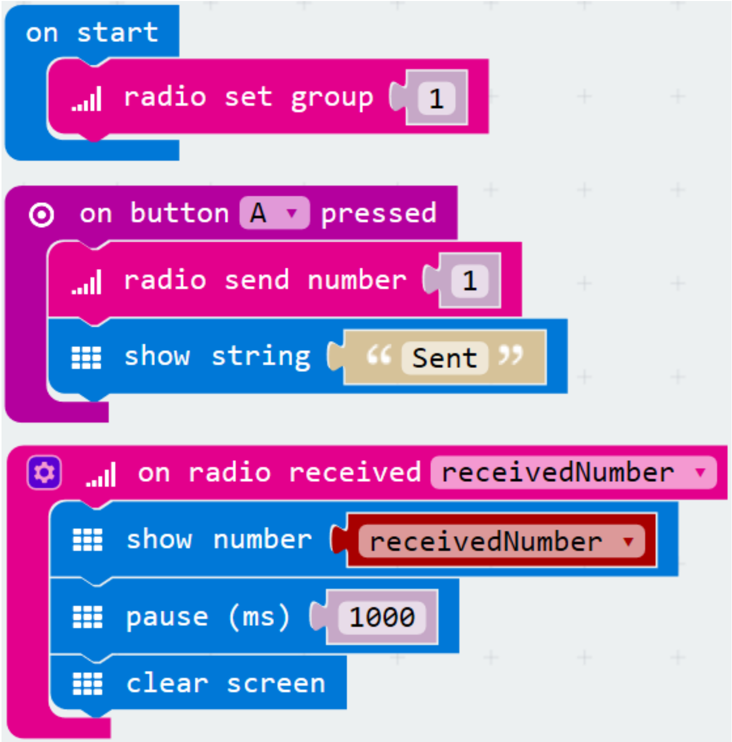


Acceleration can be in x, y, z direction, or in the total strength

Exercise: make an electronic compass with arrows show on the array

Part E – Radio Communication

- A **2.4GHz radio module** is built-in the micro:bit for wireless radio communication with other devices or micro:bit
- The range of the transmission is less than 100m
- Now, pair with another group to use the following code on both micro:bit at a distance to each other



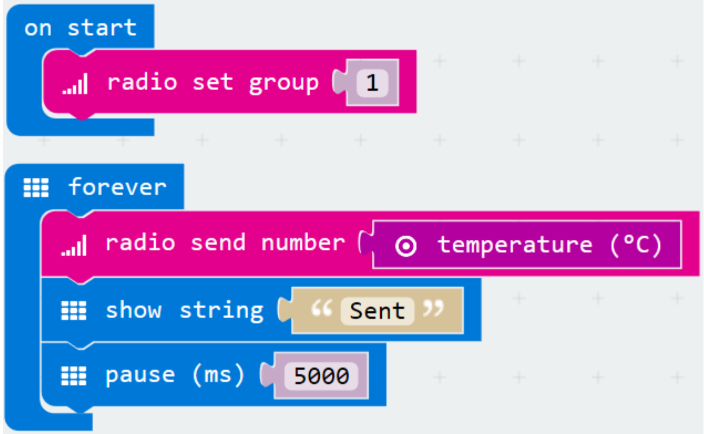
```
on start
    radio set group 1

on button A pressed
    radio send number 1
    show string "Sent"

on radio received receivedNumber
    show number receivedNumber
    pause (ms) 1000
    clear screen
```

- Exercise 1: construct a **high temperature alarm** with a micro:bit, a buzzer, and a LM35
- Exercise 2: construct **a remote temperature sensor** which send temperature reading to another micro:bit at a distance apart

Example 1 – **Integration with Science**

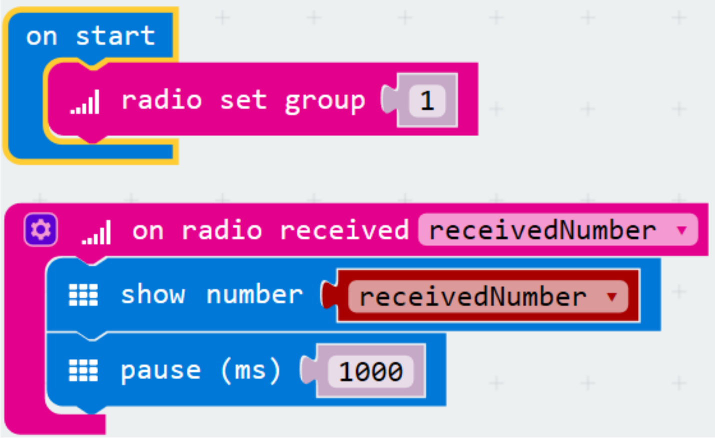
Remote sensor experiment

- Codes on the **sender** micro:bit and the **receiver** micro:bit



```
on start
    radio set group 1

forever
    radio send number temperature (°C)
    show string "Sent"
    pause (ms) 5000
```

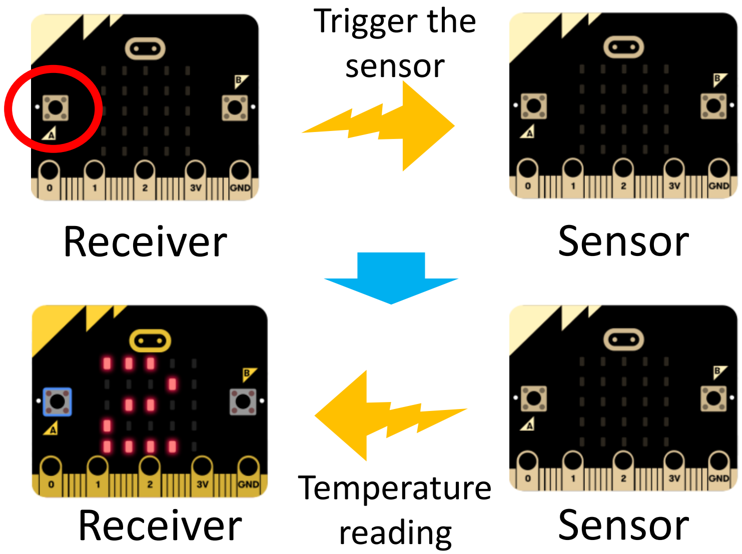
Sender (sensor) micro:bit



```
on start
    radio set group 1

on radio received receivedNumber
    show number receivedNumber
    pause (ms) 1000
```

Receiver (sensor) micro:bit

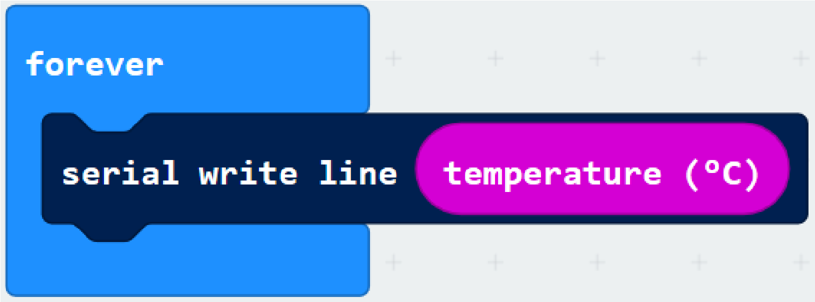
- Exercise 1: Develop **a remote sensor system** which returns the temperature of a remote sensor module only when the control button of the receiver module is triggered.



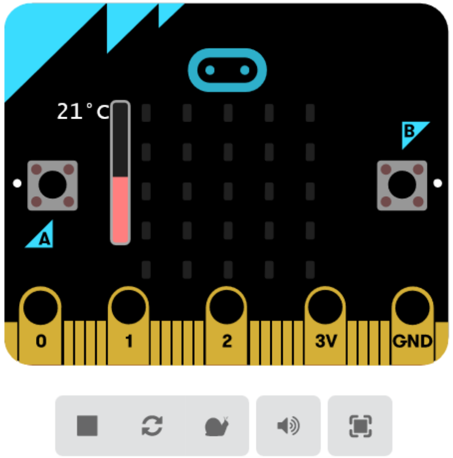
Part F – Sensor outputs on computers

- We can **show/read/record the sensor values of micro:bit continuously on computers** by using the “Serial” blocks and the Windows 10 App “Makecode” (or other compatible software

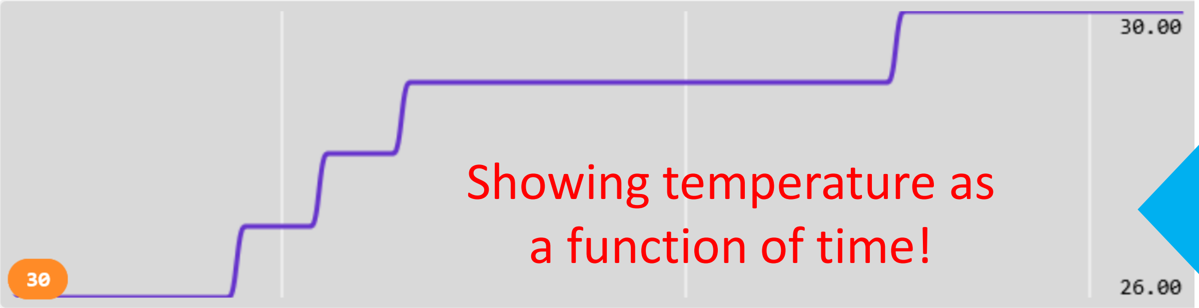
- First, download the following codes to micro:bit



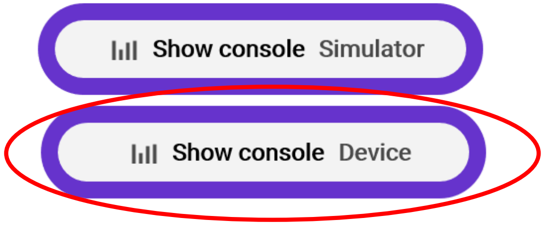
```
forever
    serial write line  temperature (°C)
```



- Then, select “**Show console Device**”

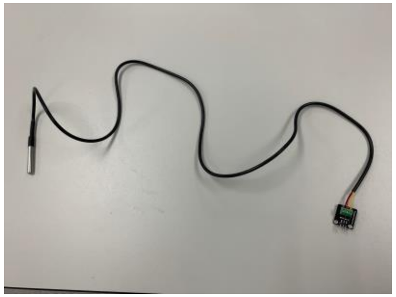


Showing temperature as a function of time!

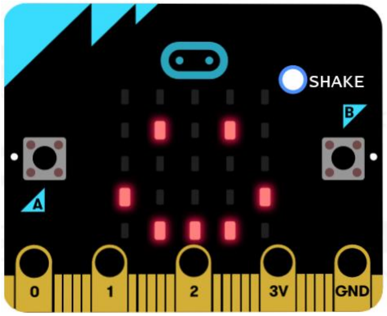


Example 2 – **Integration with Science**
Micro:bit as automatic data-loggers

- Use **temperature sensor DS18B20** to measure the cooling curve of a cup of hot water at 5s/10s interval
- The micro:bit code for DS18B20 has to be added by extension: https://github.com/DFRobot/pxt-ds18b20
- We may use "**Export data**" or "**Copy text**" (top right hand corner) in "**Show console device**", or "Serial monitor" in the platform "Arduino IDE" (but close "Makecode" first)

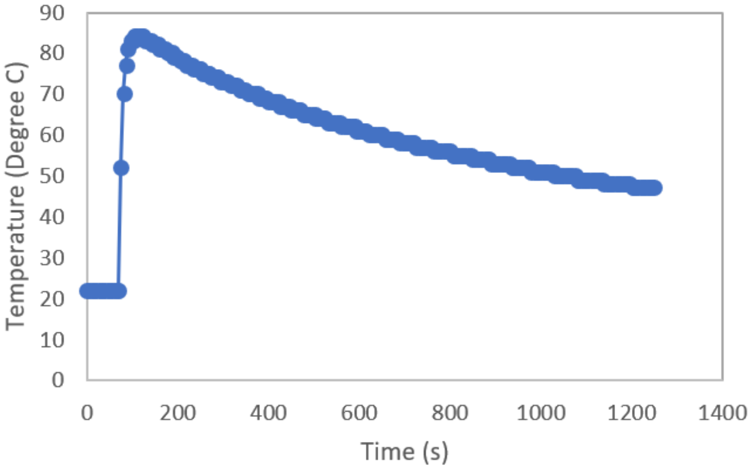


+



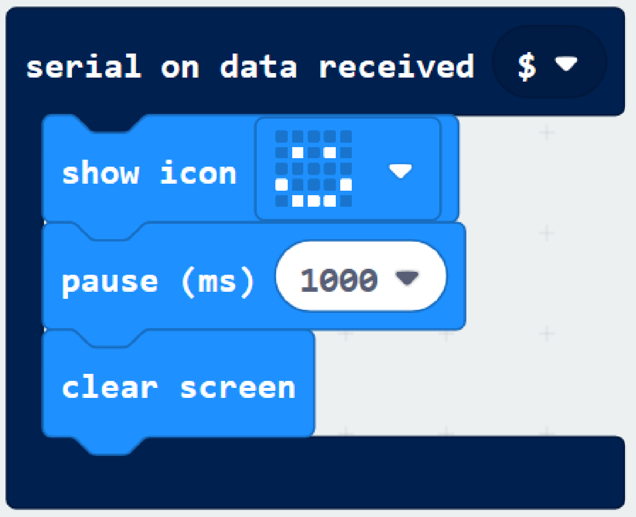
+

=



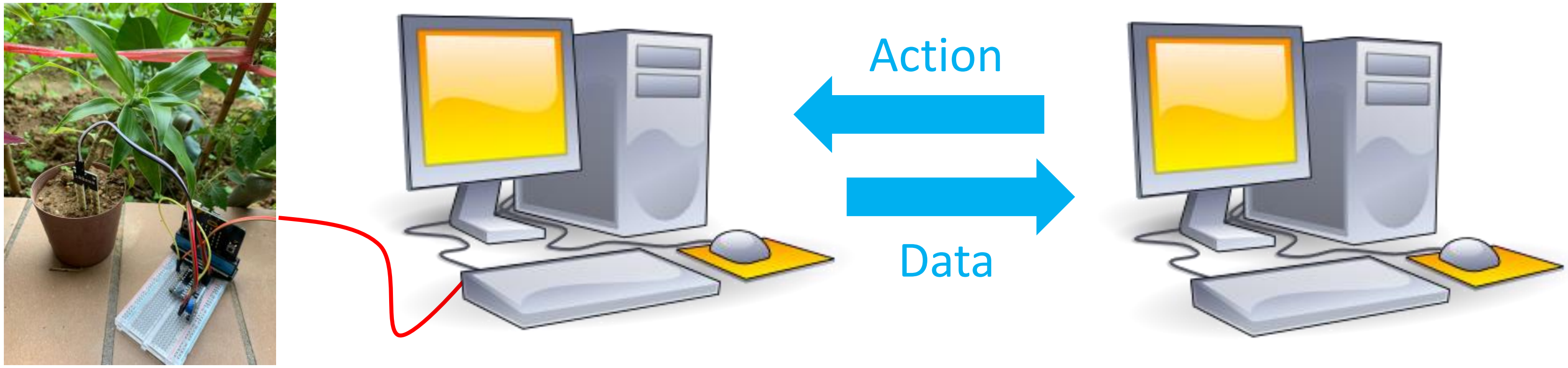
Part H - To control micro:bit via computers

- Other than reading data, we can also control micro:bit via computers and the “**serial on data received**”, for example:



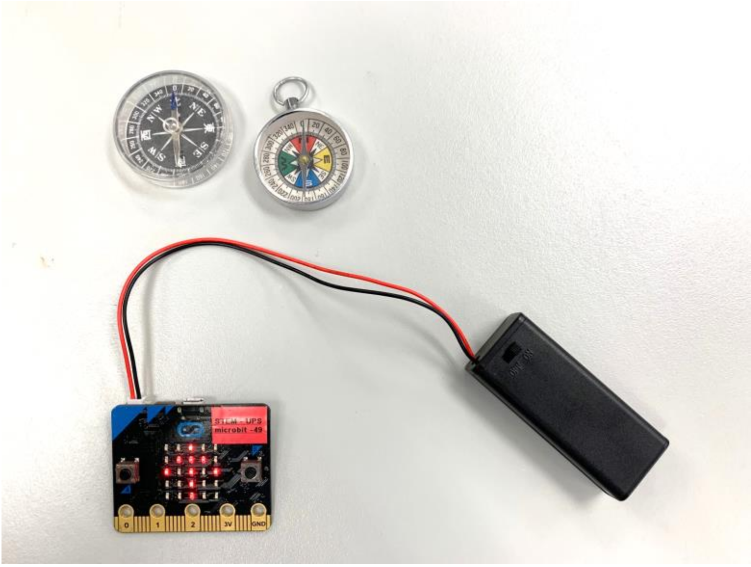
```
serial on data received  $ ▾
    show icon  ▾
    pause (ms)  1000 ▾
    clear screen
```

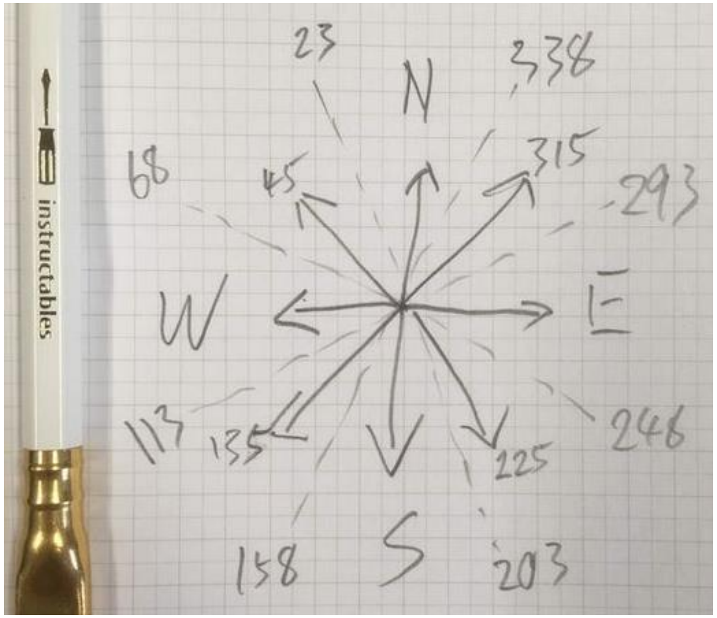
- We have to use a **computer terminal emulator program** to input to micro:bit, e.g. the serial monitor in the Arduino IDE (but we have to close the “Makecode” App first)
- This is an easy way to achieve the **Internet of Things (IoT)**



Example 3 – **Integration with Mathematics**
Micro:bit Digital Compass

- We will now use the magnetic field sensor of micro:bit to build a digital compass which always points to the north
- However, **micro:bit outputs a compass bearing angle**, how can we **make an equation** to convert the angle into an arrow direction?



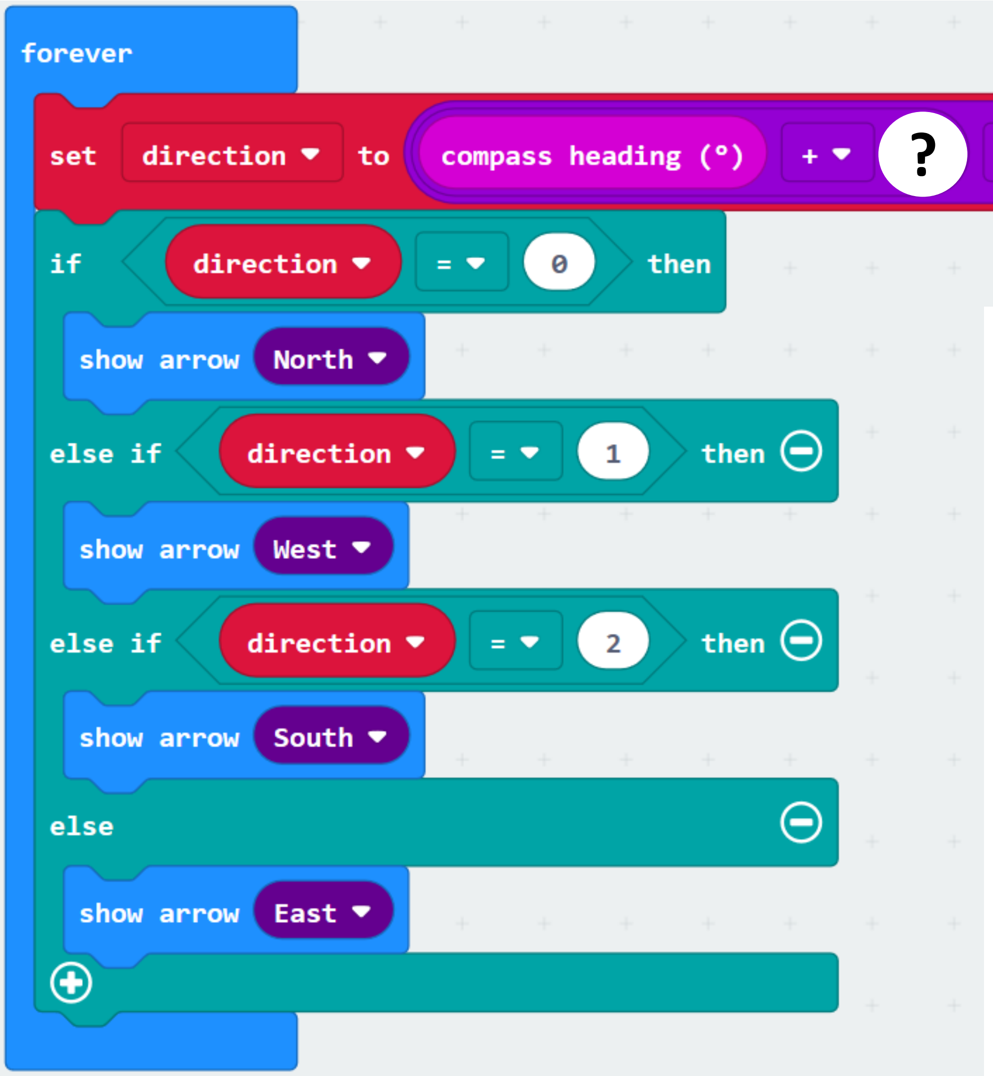


23 - 68 NW
68 - 113 West
113 - 158 SW
158 - 203 South
203 - 248 SE
248 - 293 East
293 - 338 NE
Others North

Result reference:
https://www.instructables.com/id/Microbit-Compass/

Example 3 - Micro:bit digital compass

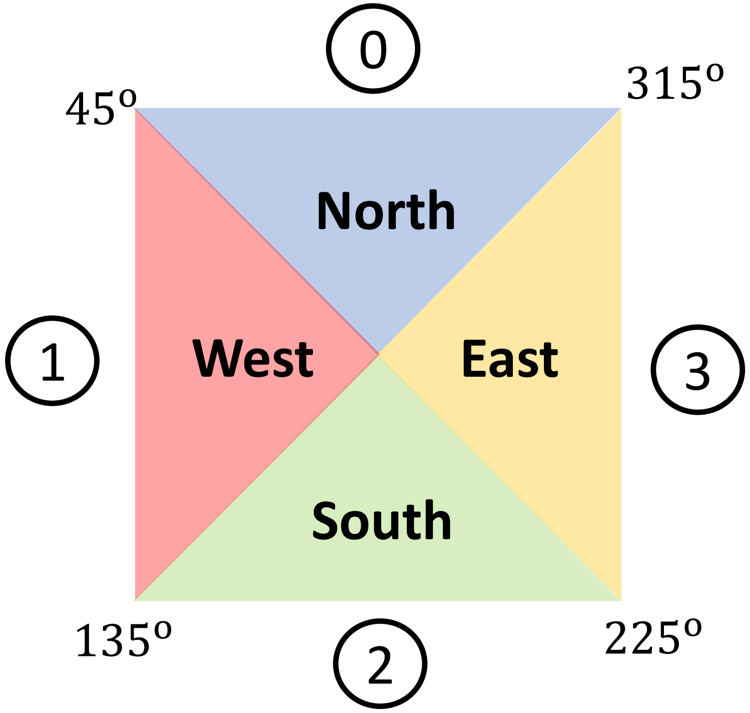
- Simple version: **4 direction**



```
forever
    set direction ▾ to ( compass heading (°) + ▾ ? integer ÷ ▾ ? )
    if direction ▾ = ▾ 0 then
        show arrow North ▾
    else if direction ▾ = ▾ 1 then ⊖
        show arrow West ▾
    else if direction ▾ = ▾ 2 then ⊖
        show arrow South ▾
    else ⊖
        show arrow East ▾
    ⊕
```

Exercise 2:

How to make a compass with **8 directions**?



Tool 2:
App Inventor

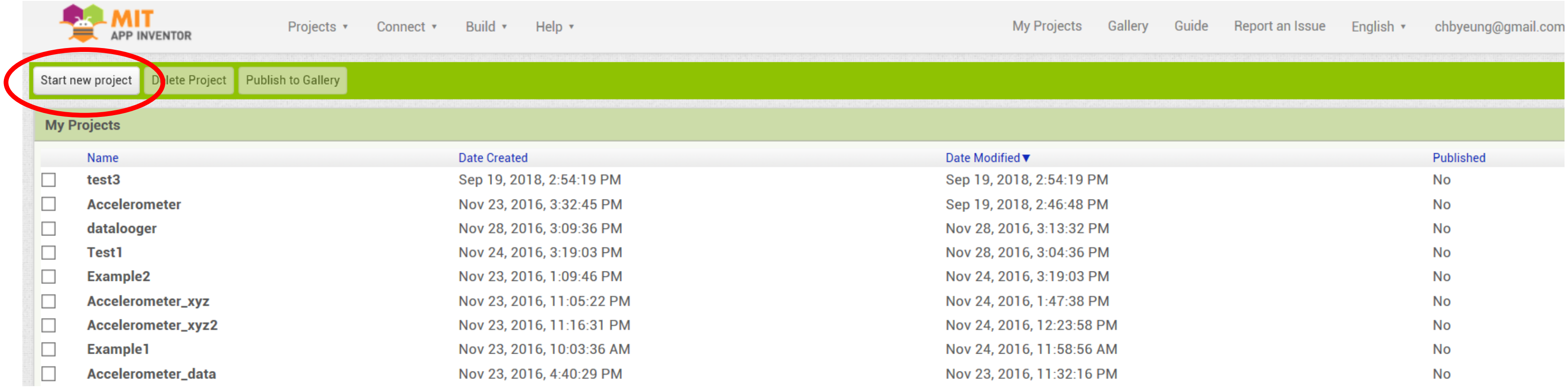
MIT App Inventor 2

- App inventor is an **open-source web application** originally developed by **Google**, now maintained by the **Massachusetts Institute of Technology (MIT)**
- It allows simple App development based on the Android **operating system**
- It employs a **Scratch-like** block-programming platform, which can be easily used by scratch users or programming new comers
- The App inventor can:

1. **Create App** which can be run on mobile devices
2. **Integrate various sensors and components** on the mobile devices
3. Connect to Google's Firebase for **data storage**

How to start?

- In a computer, open a browser (other than Internet explorer) and go to http://ai2.appinventor.mit.edu
- Login with a **google account**



	Name	Date Created	Date Modified▼	Published
□	test3	Sep 19, 2018, 2:54:19 PM	Sep 19, 2018, 2:54:19 PM	No
□	Accelerometer	Nov 23, 2016, 3:32:45 PM	Sep 19, 2018, 2:46:48 PM	No
□	datalooger	Nov 28, 2016, 3:09:36 PM	Nov 28, 2016, 3:13:32 PM	No
□	Test1	Nov 24, 2016, 3:19:03 PM	Nov 28, 2016, 3:04:36 PM	No
□	Example2	Nov 23, 2016, 1:09:46 PM	Nov 24, 2016, 3:19:03 PM	No
□	Accelerometer_xyz	Nov 23, 2016, 11:05:22 PM	Nov 24, 2016, 1:47:38 PM	No
□	Accelerometer_xyz2	Nov 23, 2016, 11:16:31 PM	Nov 24, 2016, 12:23:58 PM	No
□	Example1	Nov 23, 2016, 10:03:36 AM	Nov 24, 2016, 11:58:56 AM	No
□	Accelerometer_data	Nov 23, 2016, 4:40:29 PM	Nov 23, 2016, 11:32:16 PM	No

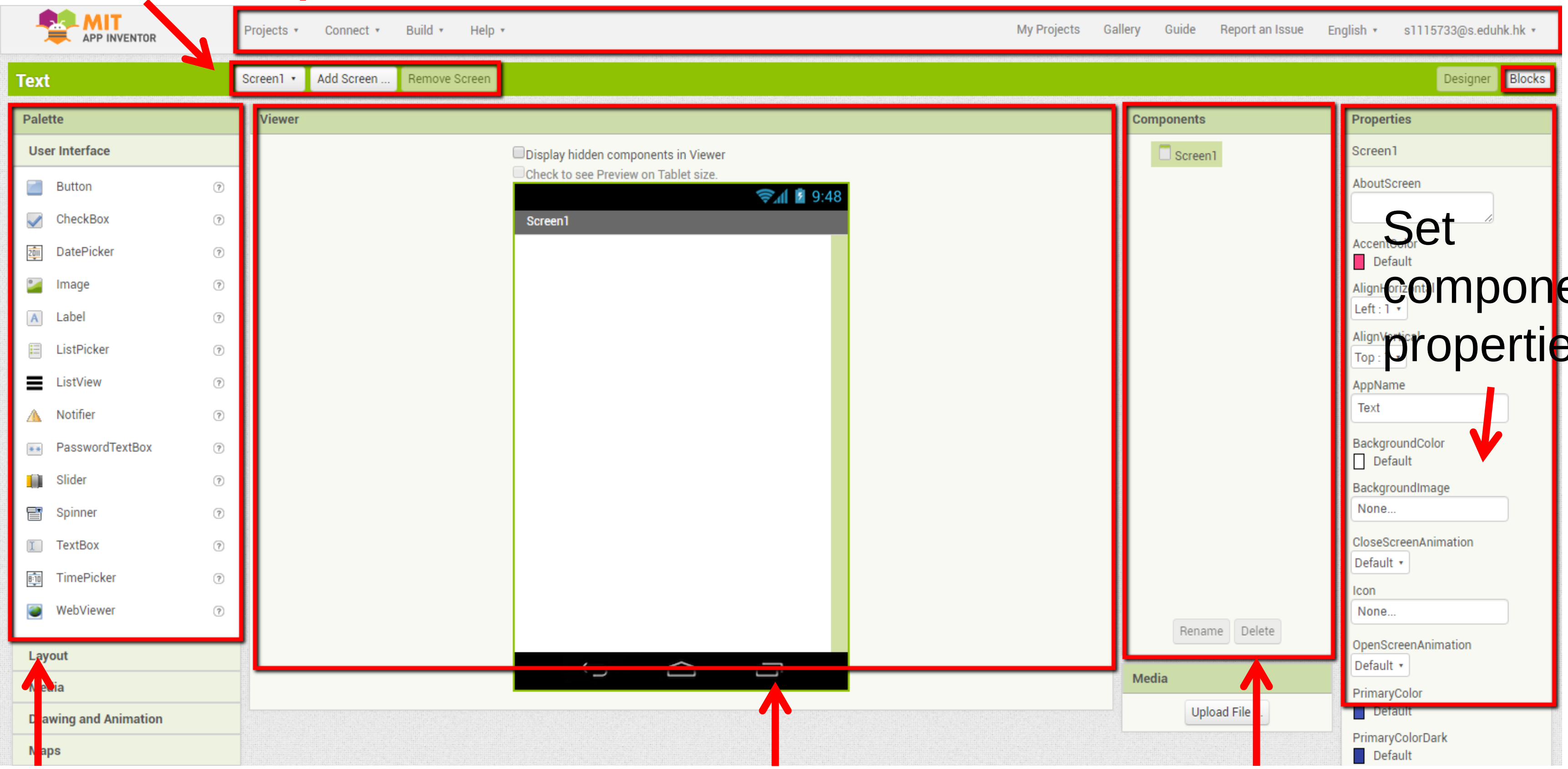
- If your account has used App Inventor before, you will see a list of all of your previous projects
- To start a new project, click "**Start new project**"

Designer Interface

Add, delete and choose the screen you want

System's function menu

"Blocks" button: go to the designer or blocks tab



Set components properties

Palette: Drag components from here to the viewer to add them to your app

Viewer: What the app in your phone will look like

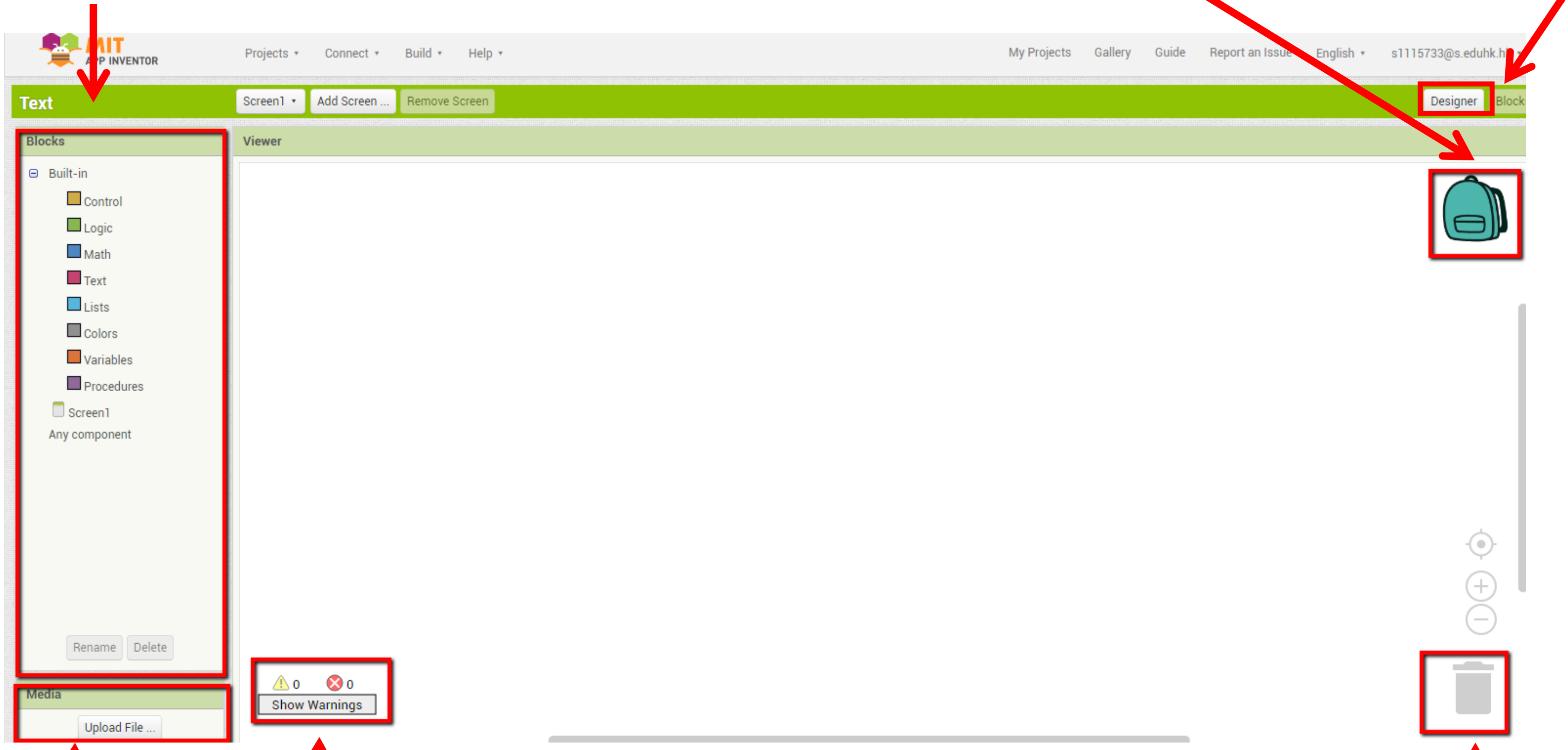
Delete and rename your components

“Blocks” Interface

Built-in drawers: Drag blocks with different behavior from here to the viewer

Duplicate and paste blocks in different screens and different projects

Go back to “**Designer**” page



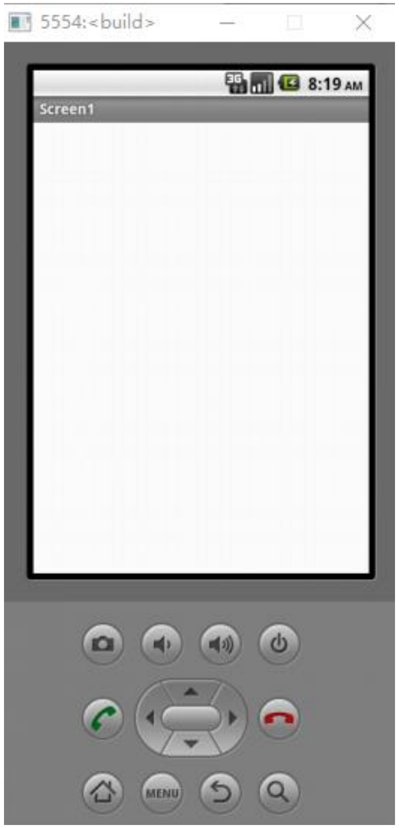
Upload media

Show warning and mistakes

Bin: Drag the blocks you want delete here

Connecting App Inventor with mobile devices?

- **AI Companion (Android phone):**
- In an Android phone, download the app "**MIT App Inventor 2**"
- In the browser page, click "**Connect**", then "**AI companion**". Open the App inventor App on the phone and scan the **QR code** to connect the mobile devices to the browser App inventor

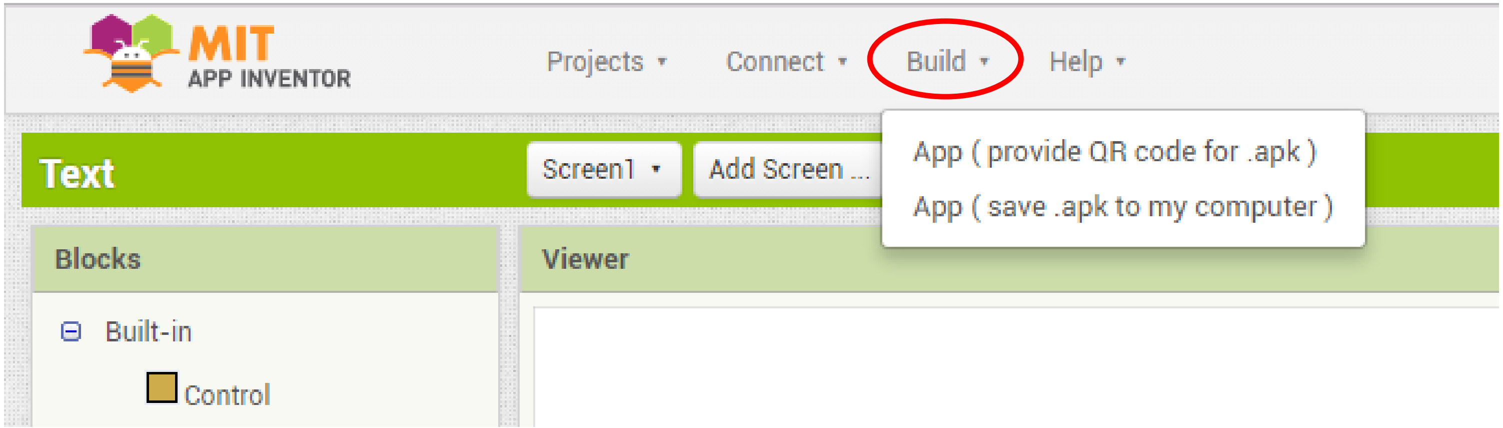


- **Emulator (Computer):**
- Open aiStarter.exe installed in your computer when you installed MIT App Inventor 2. Click the "**Connect**" button in the browser page, then "**Emulator**".

Output App as .apk file

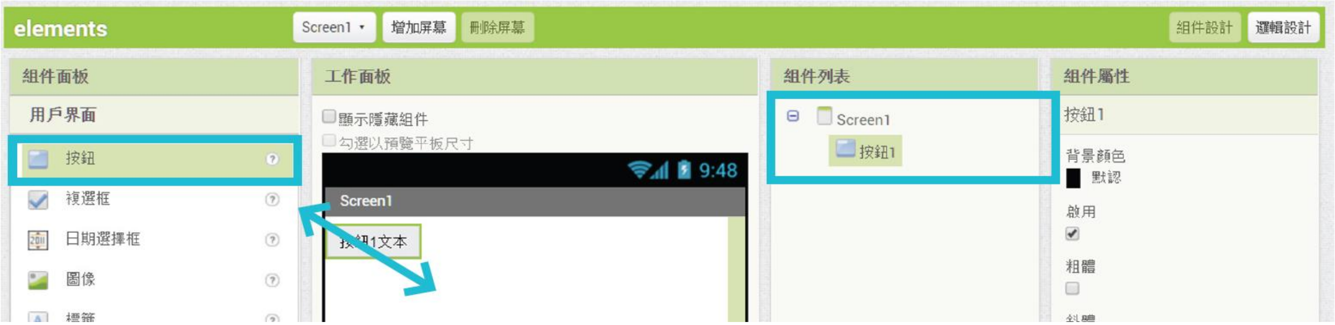
- Click “Build”, and the either

1. open MIT AI2 Companion in your phone to scan the QR code (Click the first row)
2. save the created App as .apk file in your computer (Click the second row)

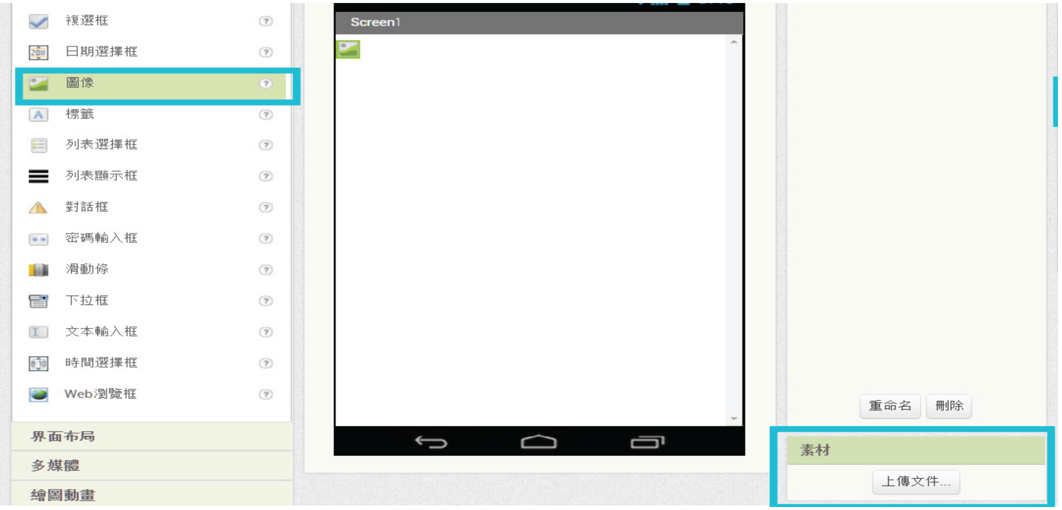


The Basic Elements (1)

- **Button:** Codes are executed it is clicked; its properties can be changed in the properties panel, e.g. text on the button



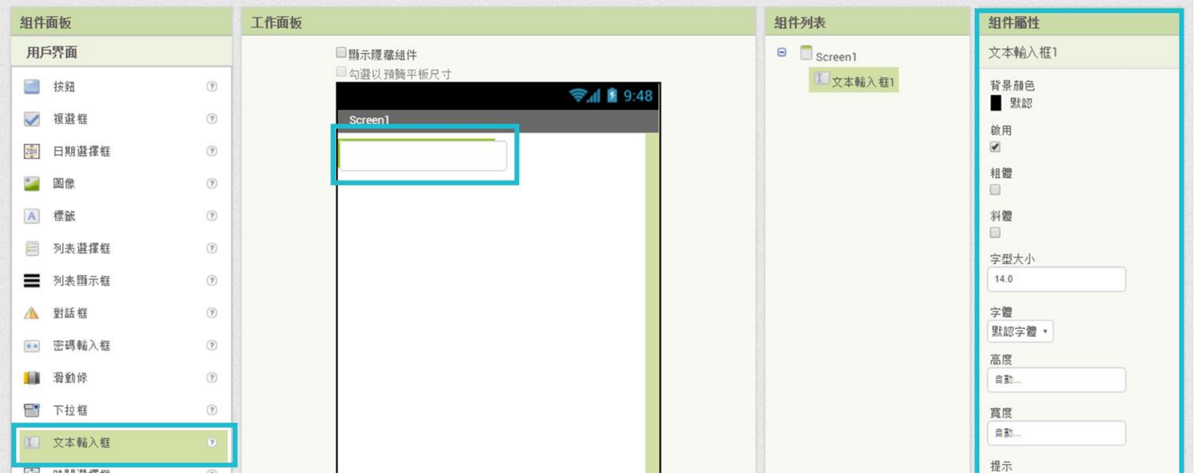
- **Image:** The image you have uploaded



Upload your image here

The Basic Elements (2)

- **Label:** Showing text
- **Textbox:** Entering text by users



文本輸入框1

背景顏色
■ 默認

啟用
☑

粗體
□

斜體
□

字型大小
14.0

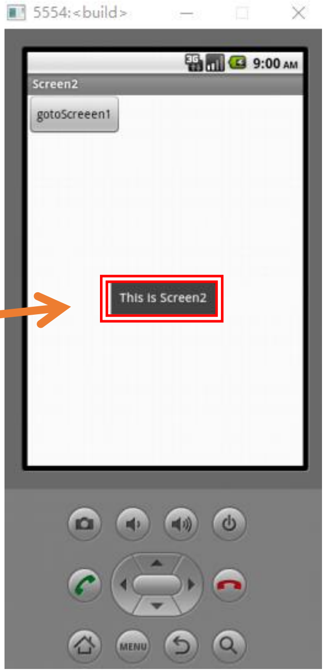
字體
默認字體 ▾

高度
自動...

寬度
自動...

提示

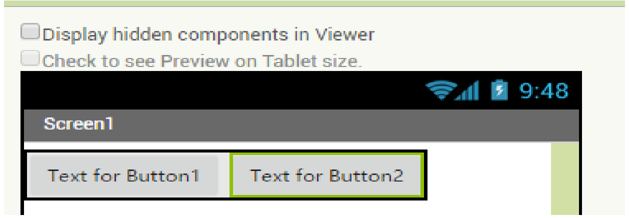
- **Notifier:** Pop-up warning messages or hints



This is Screen2

The Basic Elements (3)

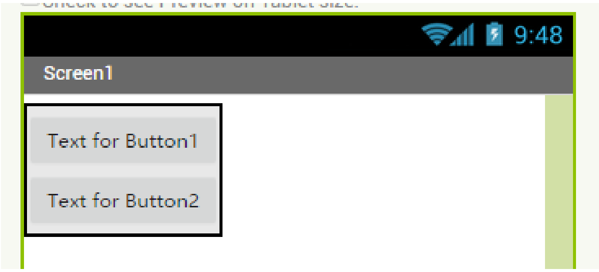
- **Layout:** alignment of components
  - **Horizontal Arrangement:** align components horizontally



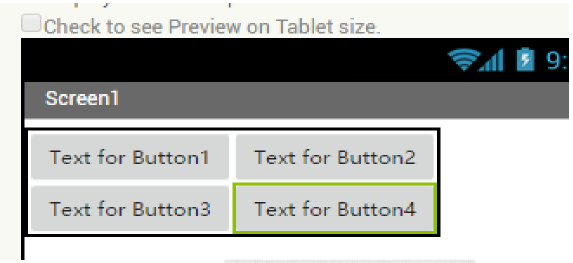
□ Display hidden components in Viewer

□ Check to see Preview on Tablet size.

  - **Vertical Arrangement:** align components vertically

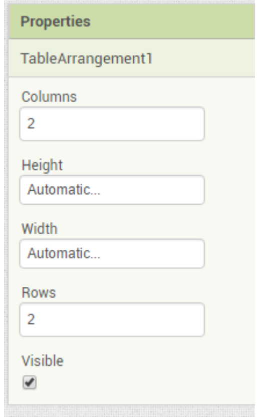


  - **Table Arrangement:** align components in a table



□ Check to see Preview on Tablet size.

  - their number of columns and rows can be changed in **properties** on the right



Properties

TableArrangement1

Columns: 2

Height: Automatic...

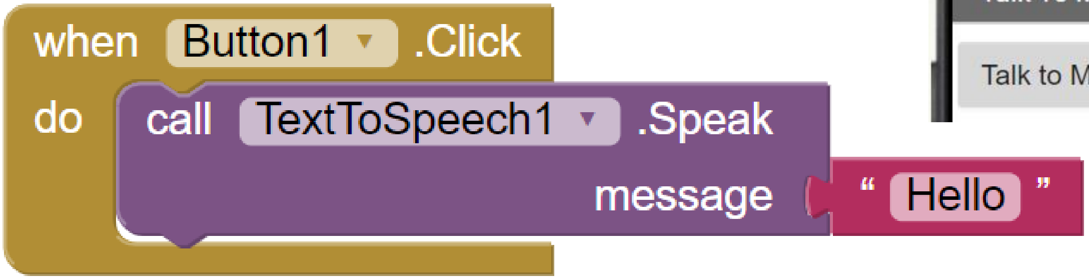
Width: Automatic...

Rows: 2

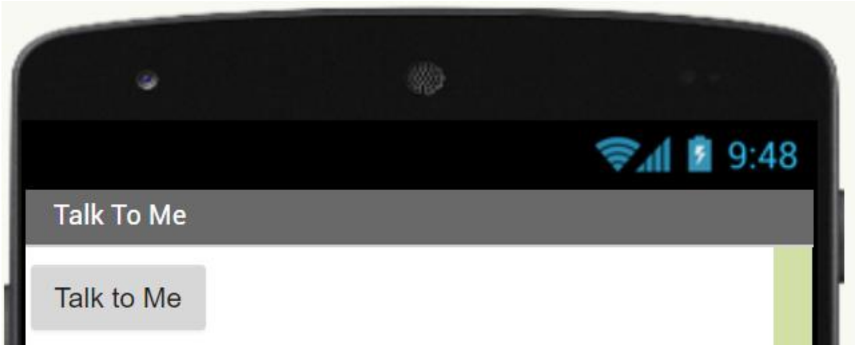
Visible: ☑

Example 1: The first app “Talk to me”

- In the “**Designer**“ interface, on the left column, go to “**User interface**” and drag a “**Button**” to the central “**Screen1**”
- Click the button and in “**properties**” on the right column, re-name the Button by changing the “Text” to “**Talk to me**”
- Go to “**Media**” on the left column and then drag “**TextToSpeech**” to the screen
- Click “**Blocks**“ on right hand corner and go to the **block interface**
- Type the following code:

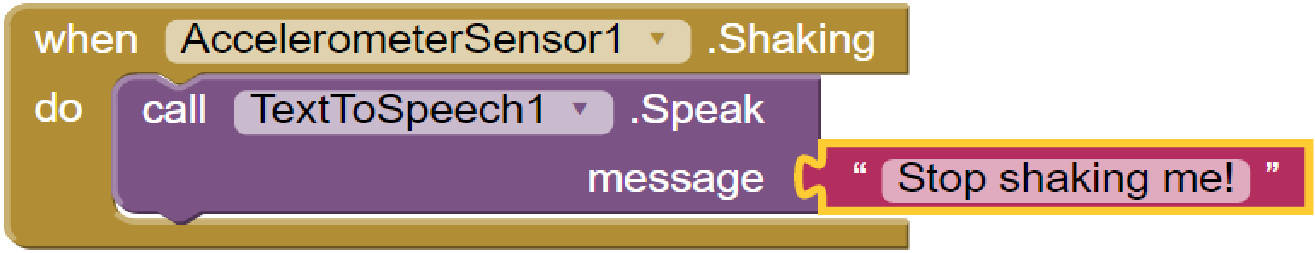


```
when Button1 ▾ .Click
do   call TextToSpeech1 ▾ .Speak
                        message  " Hello "
```



Example 2: Use the Accelerometer to trigger the App

1. In the "**Designer**" interface, go to "**Sensors**" and drag "**AccelerometerSensor**" to the screen
2. Go to the "**Blocks**" interface, and build the following blocks:



```
when AccelerometerSensor1 ▾ .Shaking
do  call TextToSpeech1 ▾ .Speak
                     message  " Stop shaking me! "
```

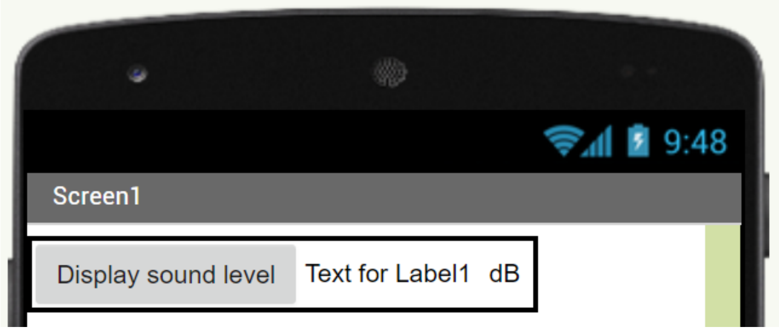
3. Type **"Stop shaking me!"** in the textbox
4. Shake your phone to experience the App

Exercise

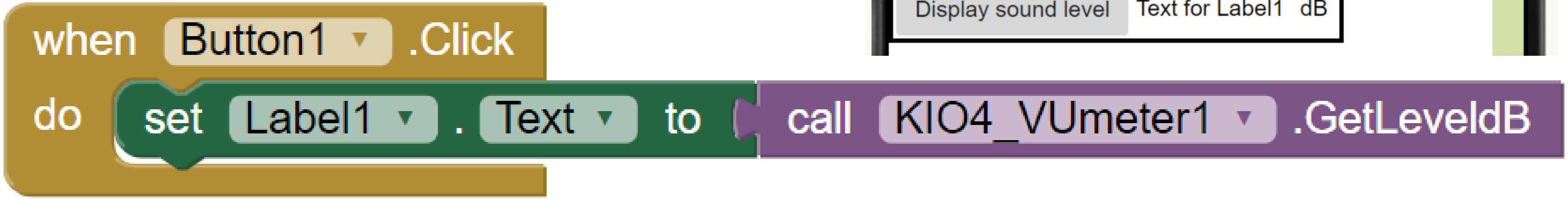
- Develop the following App – Enter your name in a **textbox**, when the **phone is shaken**, the App **says hello** to the user

Example 3: **Integration with Science**
Build a Sound Meter App

- We will now build an app to measure sound level
- First of all, we have to import an extension "com.KIO4_VUmeter.aix" (Google it), by clicking the last row "**Extension**" in the left column
- Drag **1 button**, **2 labels** and "**KIO4_VUmeter**" to screen
- Convert the button to "Display sound level", and one of the labels to "dB", as shown here:



- Input the following code:

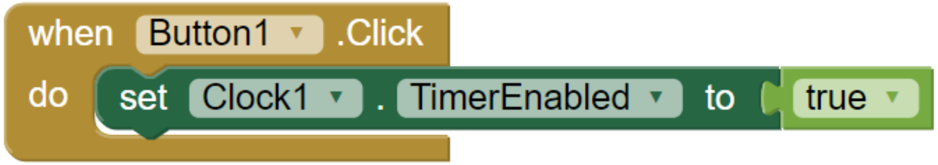


```
when Button1 .Click
do  set Label1 . Text to  call KIO4_VUmeter1 .GetLeveldB
```

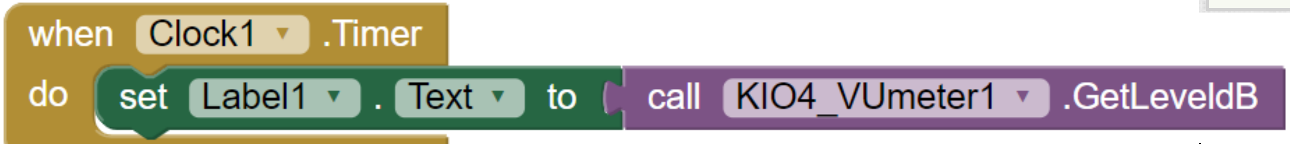
- Click "Display sound level" on the app to show sound level

Continuous measurement

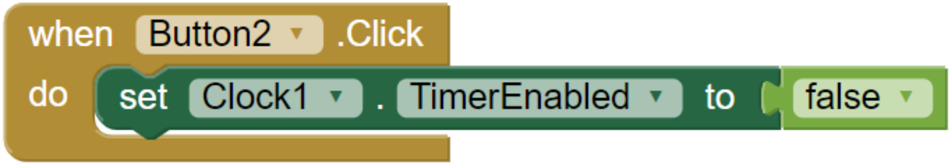
- To continuous display the reading of sensors, one has to use "Clock" in sensors
- Add **2 buttons**, **2 labels**, **KIO4_VUmeter** and **1 clock** to the screen; change one button to "Start", one button to "Stop", one label to "dB"
- For the clock, uncheck "**TimerEnabled**", and set the "**TimerInterval**" to 1000
- Build the following code:



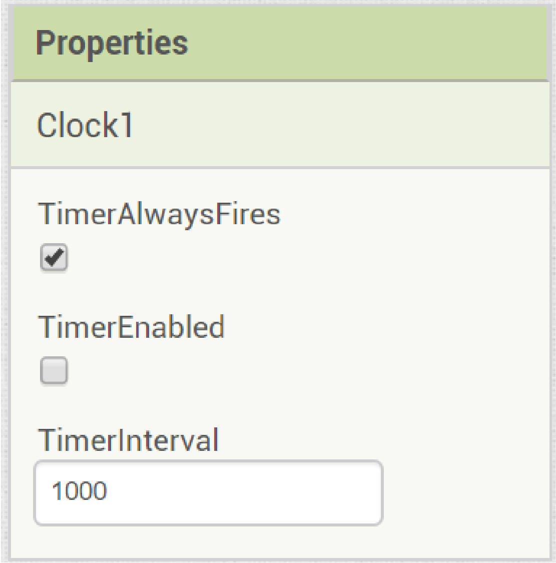
```
when Button1 .Click
do  set Clock1 . TimerEnabled to true
```



```
when Clock1 .Timer
do  set Label1 . Text to call KIO4_VUmeter1 .GetLeveldB
```



```
when Button2 .Click
do  set Clock1 . TimerEnabled to false
```



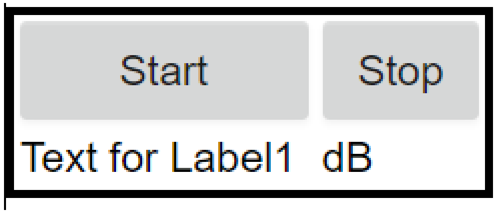
Properties

Clock1

TimerAlwaysFires
☑

TimerEnabled
□

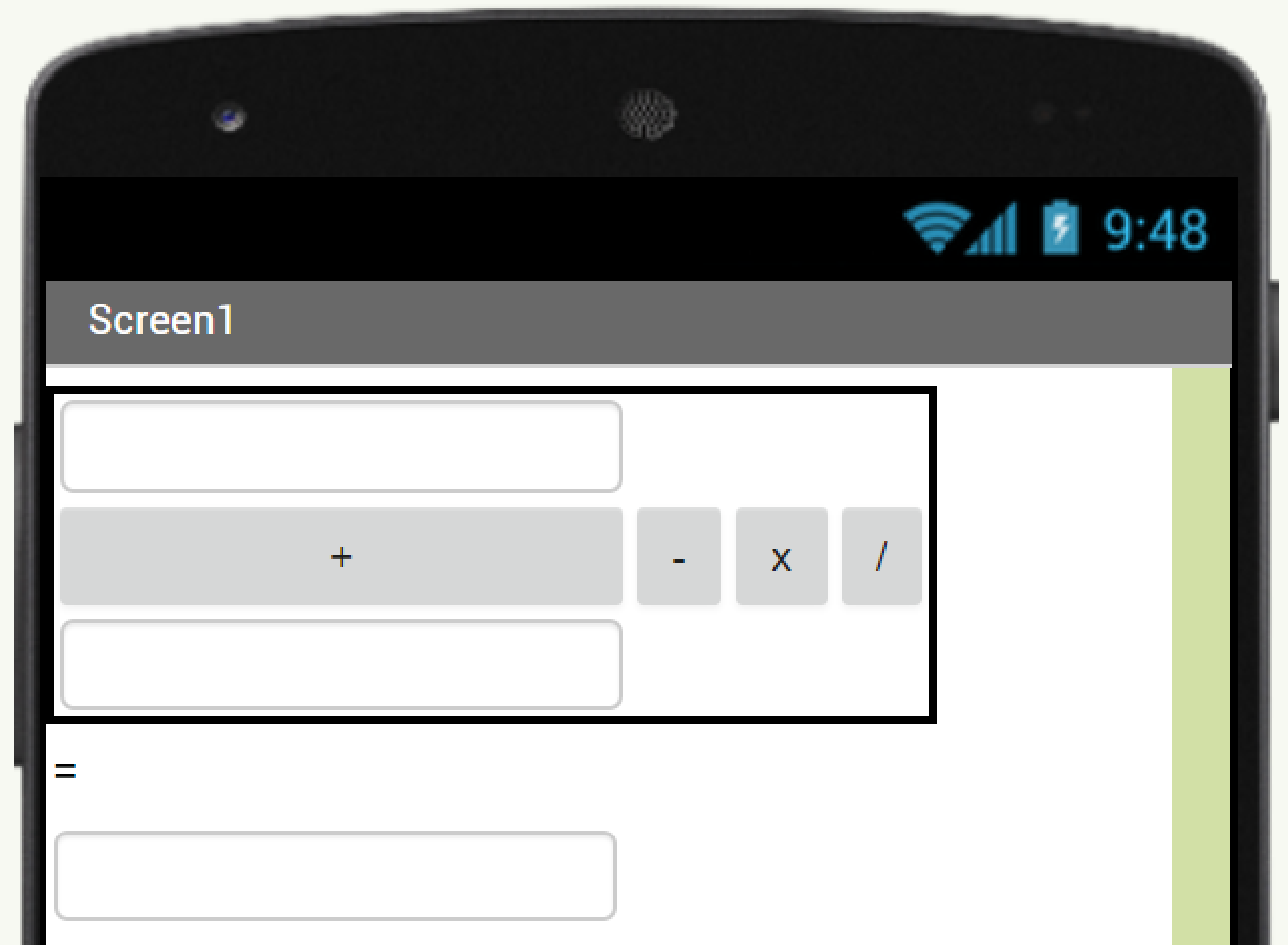
TimerInterval
1000



Example 4: **Integration with Mathematics**

Build your Calculator

- Use the blocks in “**Math**”, and the techniques we learnt in the previous exercises to build the following **calculator**:



Summary

- STEM electronic tools can be potential platform to integrate STEM into different school-based subject curriculum

- **Micro:bit** – micro-controllers which can be coded by **scratch-like web-based coding platform, to read** internal sensors, external sensors, and external devices
- Example of STEM activities: remote sensing experiments, data-logging, digital compass

- **MIT App Inventor** – a platform to **create App** which can be run on mobile devices, and can be **integrated to various sensors and components** on the mobile devices
- Example of STEM activities: using sensors on mobile phones as data-logger, making calculators