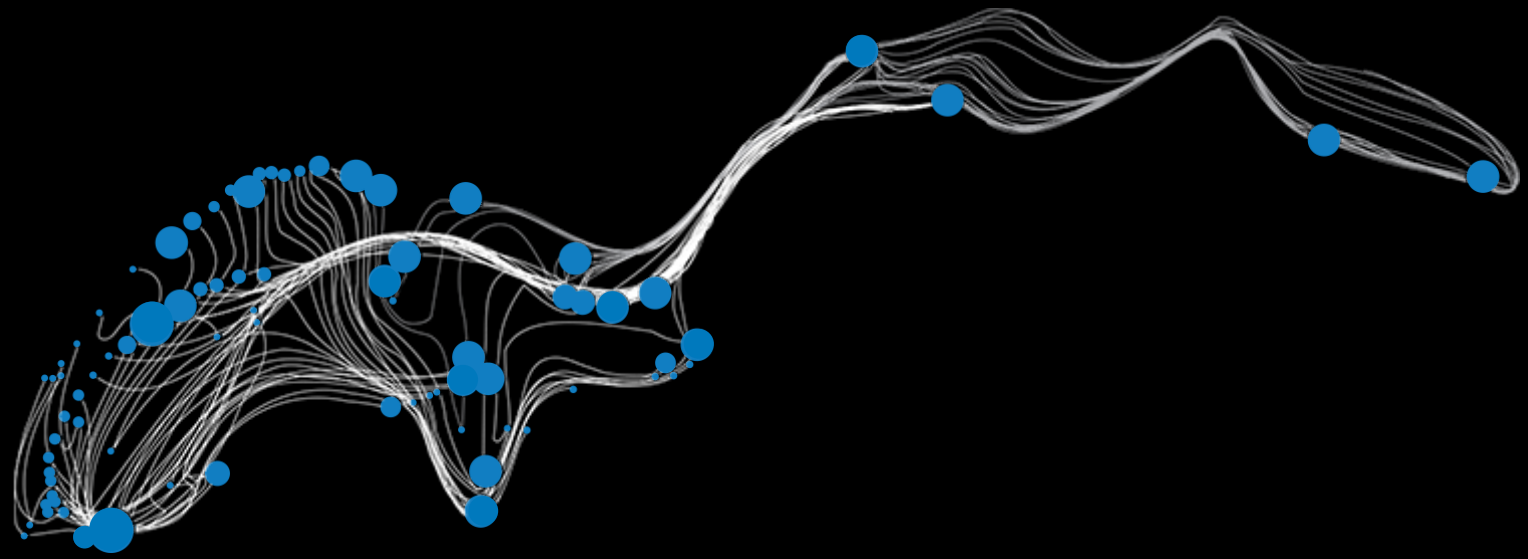
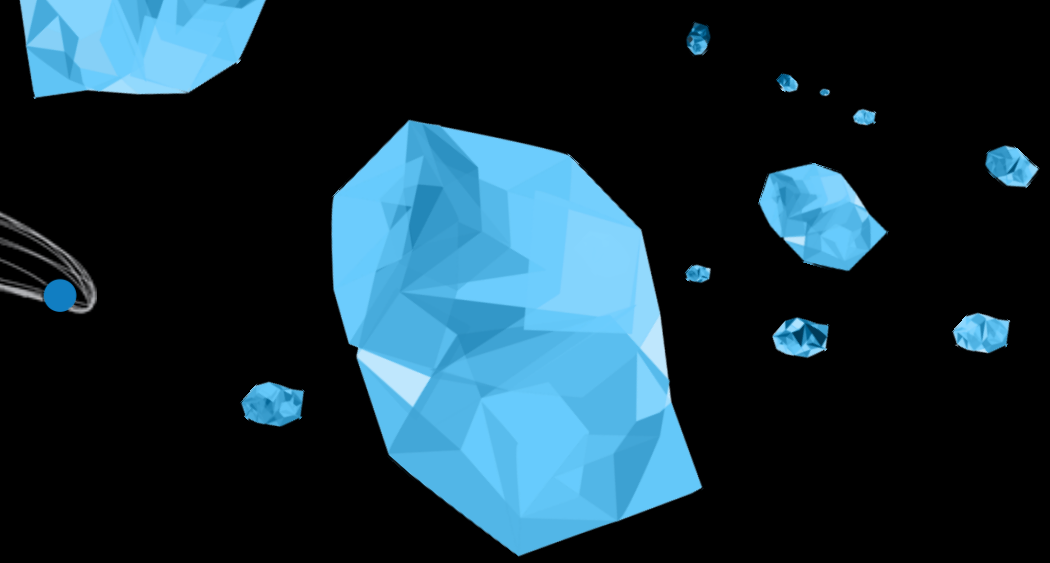
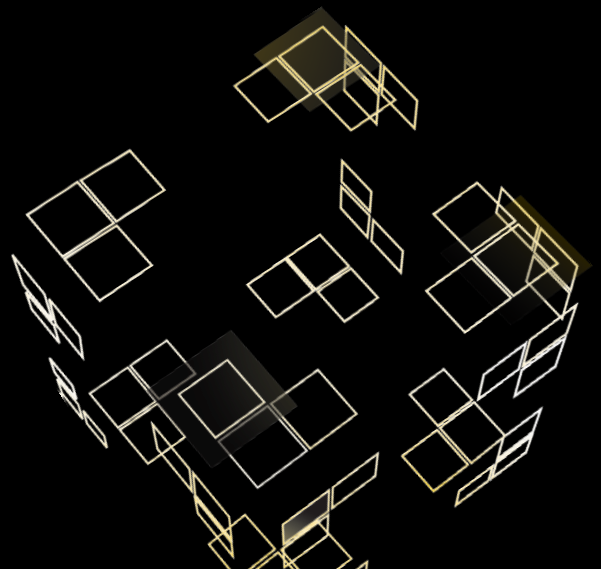
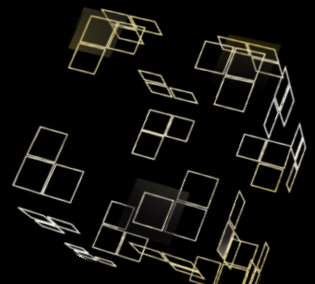




Workshop “prototyping”

By Fjodor van Slooten





UNIVERSITY OF TWENTE.

CONTENT

Workshop prototyping

- Recap App Development M4
- Physical prototypes
  - Building connected sensors using an ESP module (Wifi)
  - Control electronics from an Android App via Bluetooth
- App Mockups (‘Virtual’ protypes)
- Intro to Mobile App Development

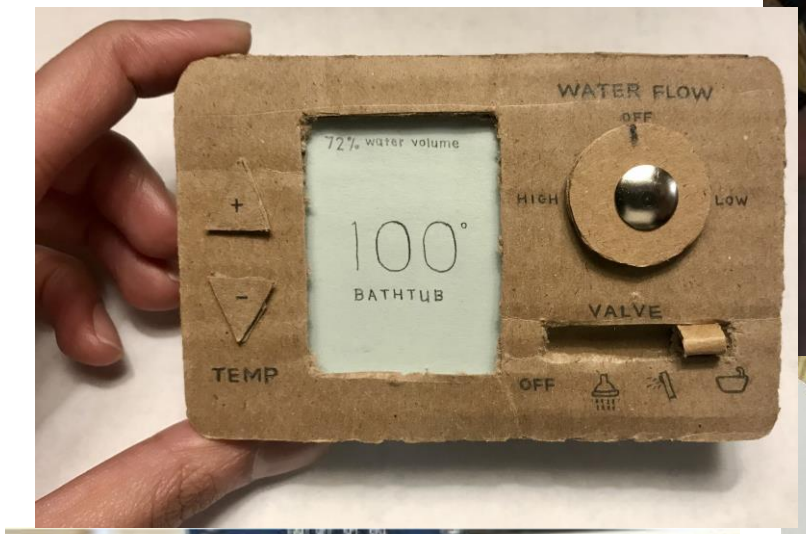
Including 'hands-on' tutorials you can work on today

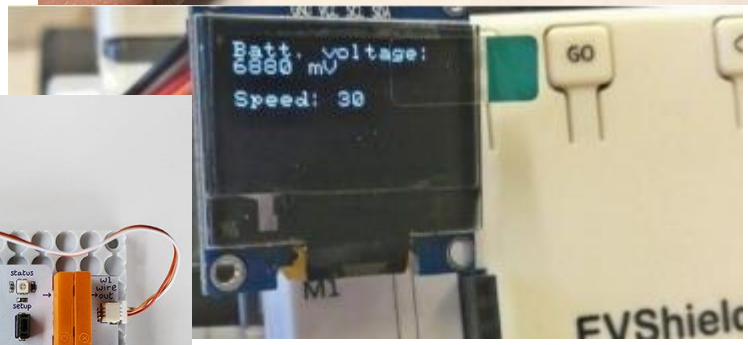
Contact: f.vanslooten@utwente.nl, W241

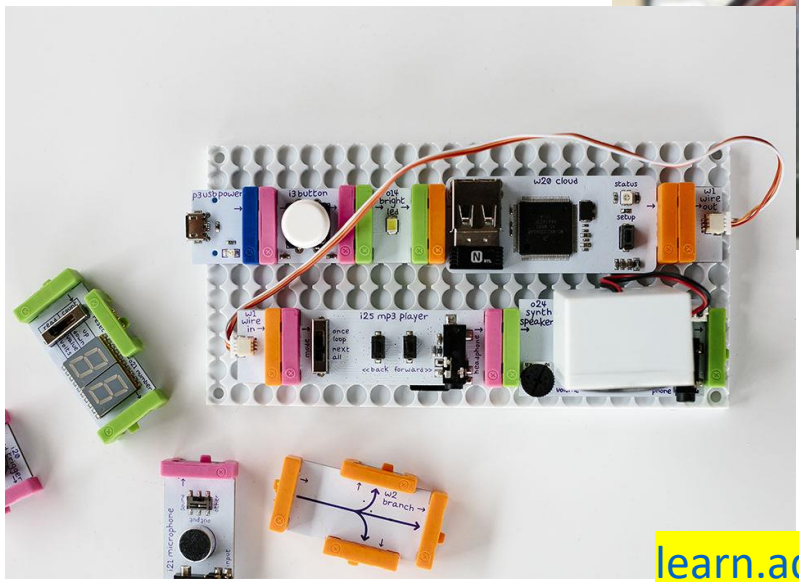
This presentation & tutorials available at: vanslooten.com/workshop

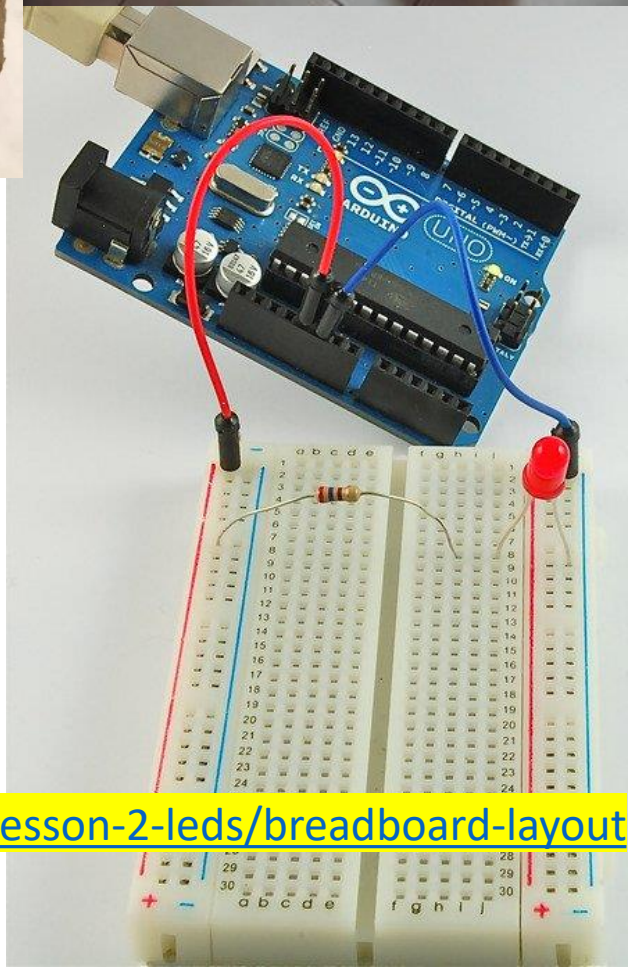
RECAP

- Physical prototypes
  - Cardboard, wood, paper
  - Electronics: Arduino
  - Lego Mindstorms
  - LittleBits









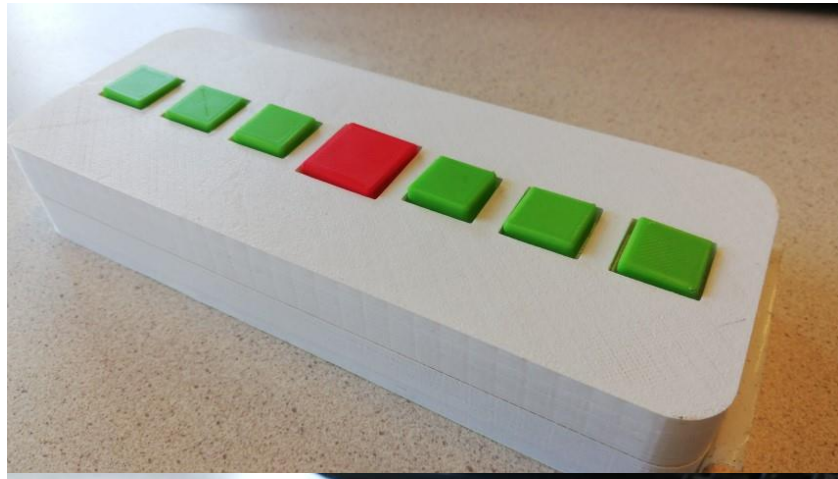
learn.adafruit.com/adafruit-arduino-lesson-2-leds/breadboard-layout

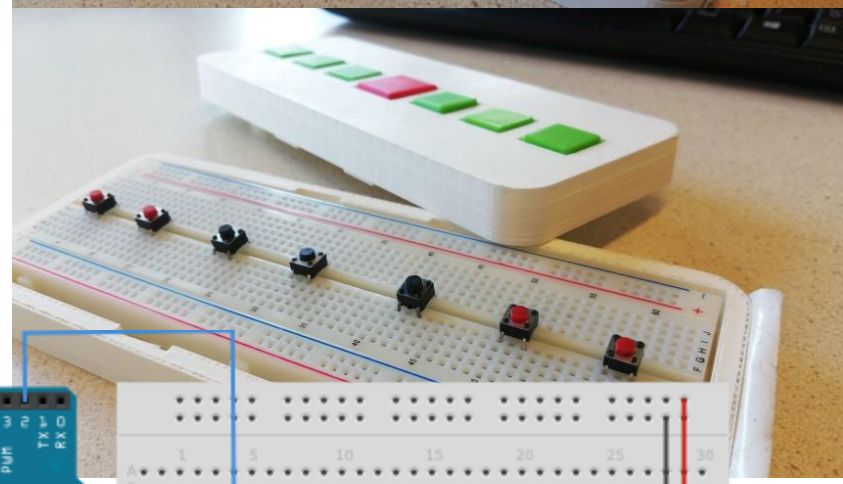
Arduino programming: a button

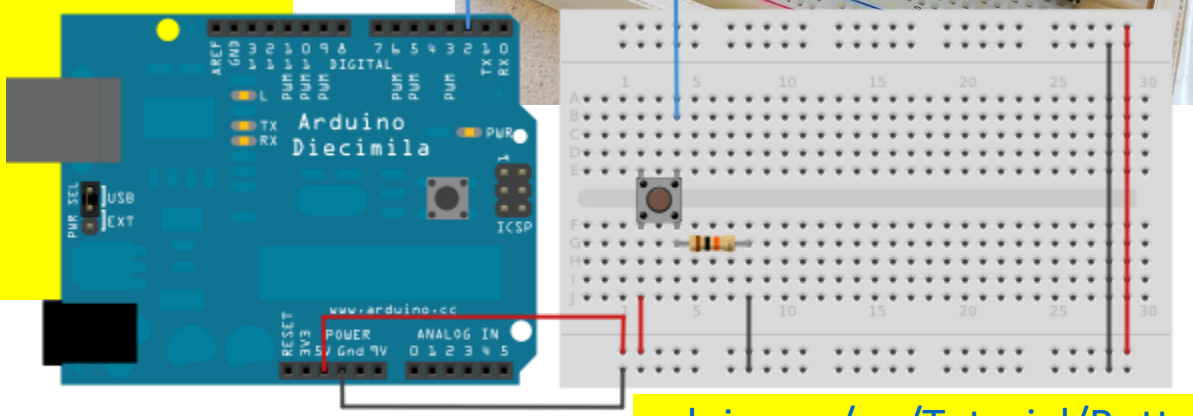
[simplest kind of sensor]

```
// read the state of the pushbutton value:
buttonState = digitalRead(buttonPin);

// check if the pushbutton is pressed.
// If it is, the buttonState is HIGH:
if (buttonState == HIGH) {
    // turn LED on:
    digitalWrite(ledPin, HIGH);
} else {
    // turn LED off:
    digitalWrite(ledPin, LOW);
}
```







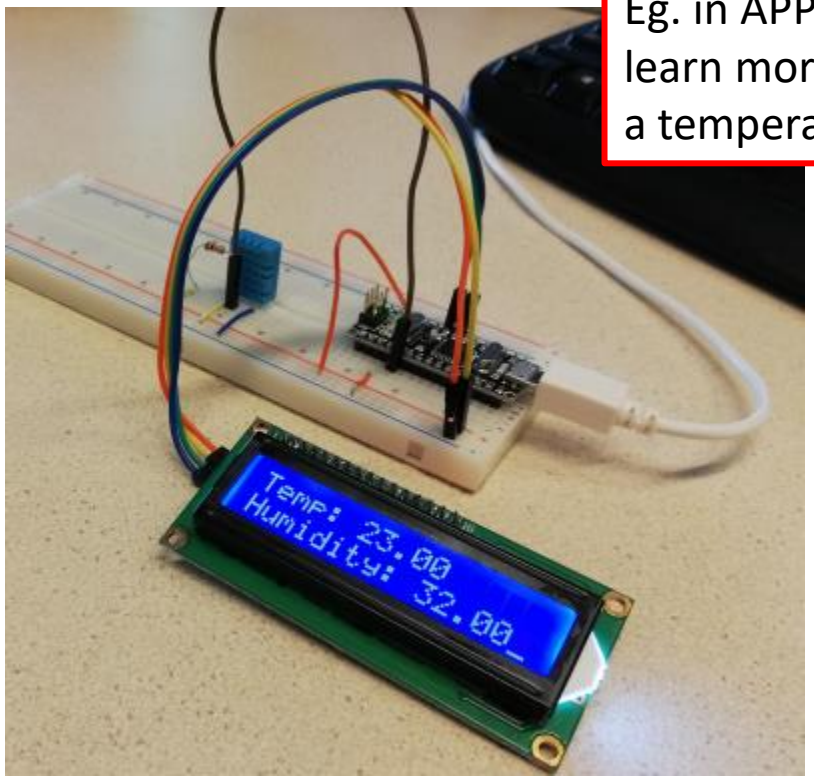
arduino.cc/en/Tutorial/Button

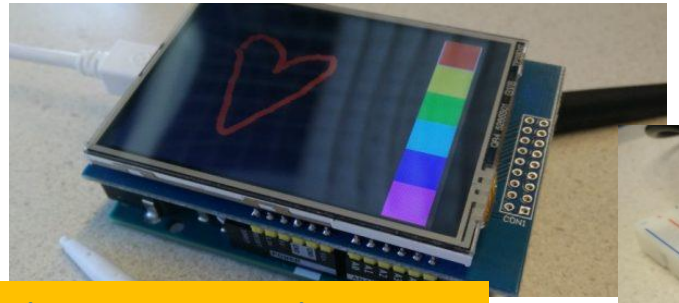
Arduino lessons

learn.adafruit.com/series/learn-arduino

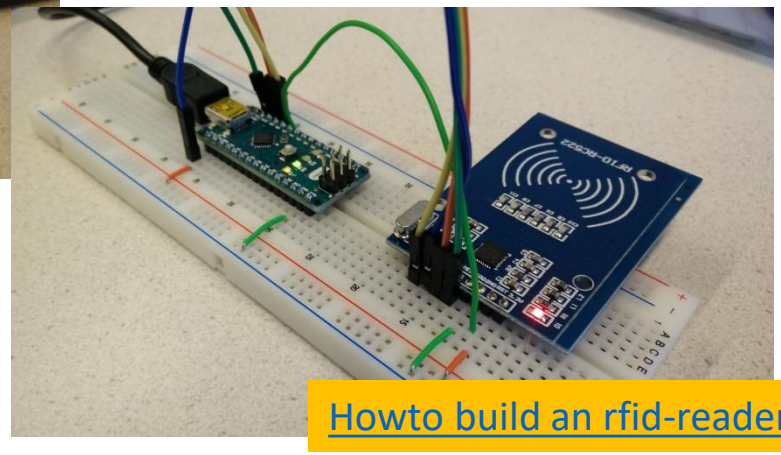
Learn more on Arduino programming, with lots of examples at Application Development (APPDEV) course

Eg. in APPDEV Lesson LegoPracticalSession3, learn more about: use of displays and how to use a temperature sensor





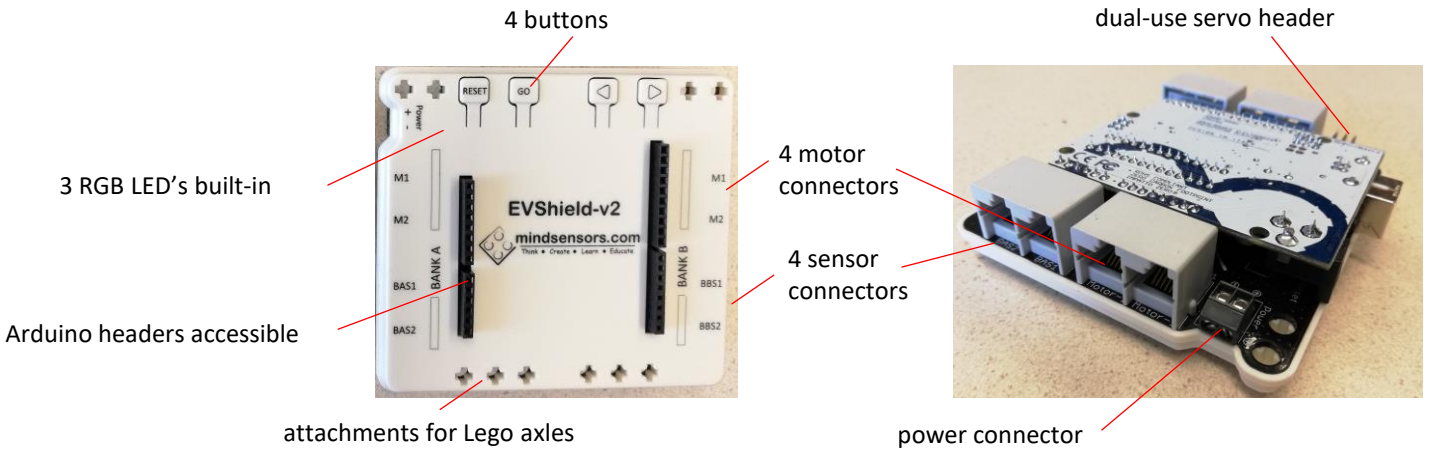
Touch screen examples: keypad, paint, image viewer and calculator



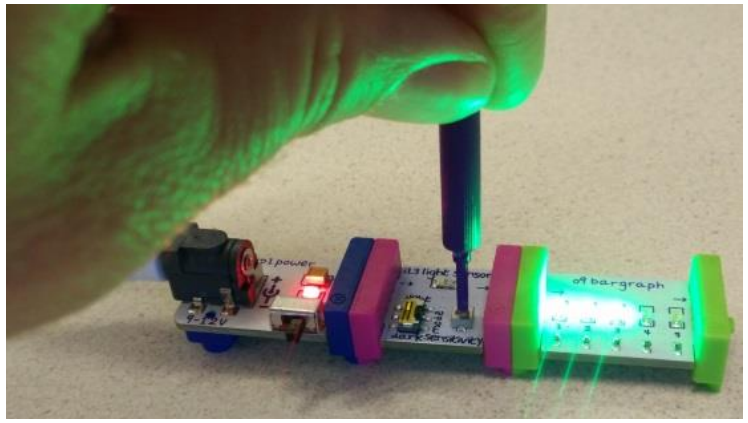
Howto build an rfid-reader

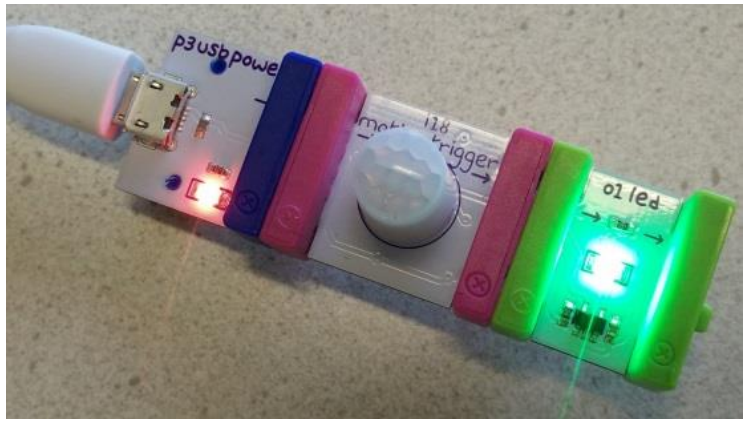
EVShield: control Lego Mindstorms with Arduino

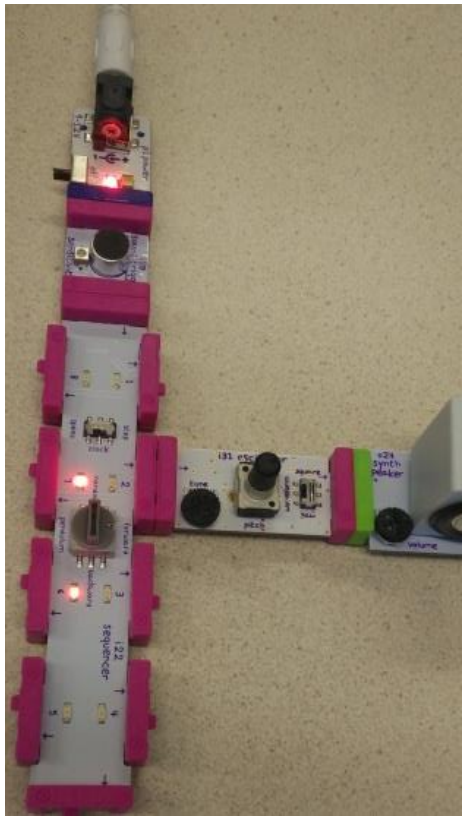
- Connect (almost) all Lego motor's and sensors

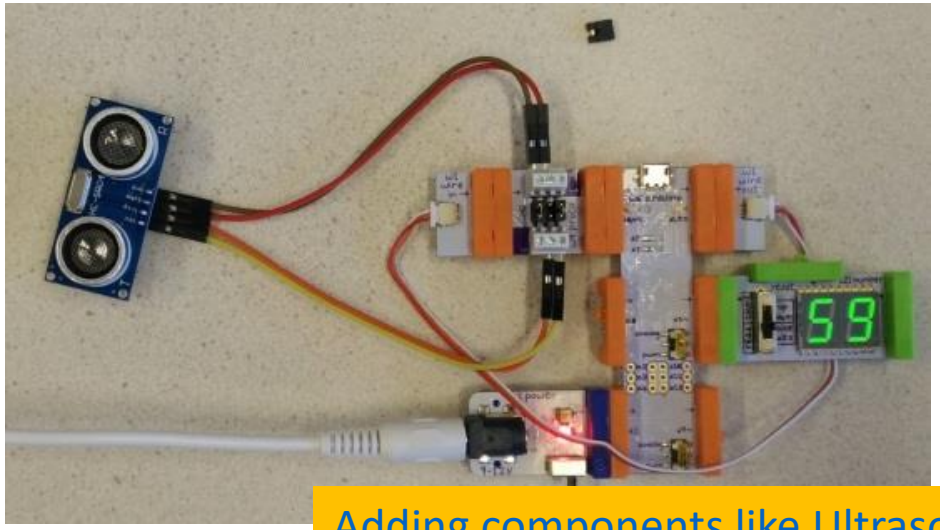


LittleBits Examples

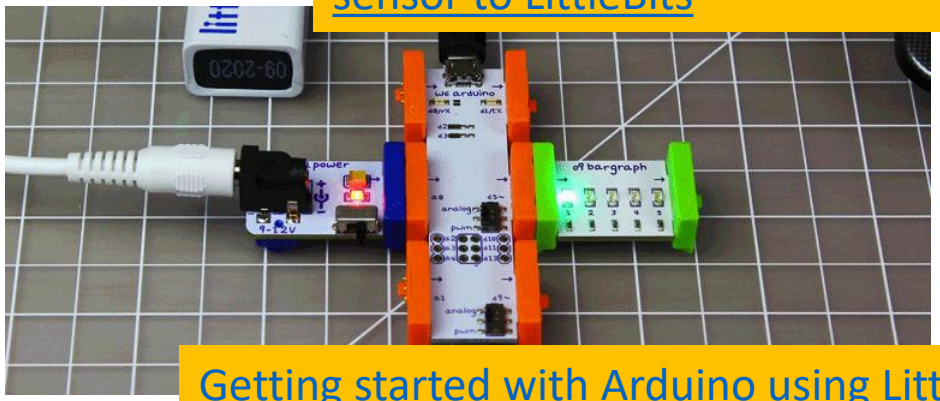








[Adding components like Ultrasonic sensor to LittleBits]

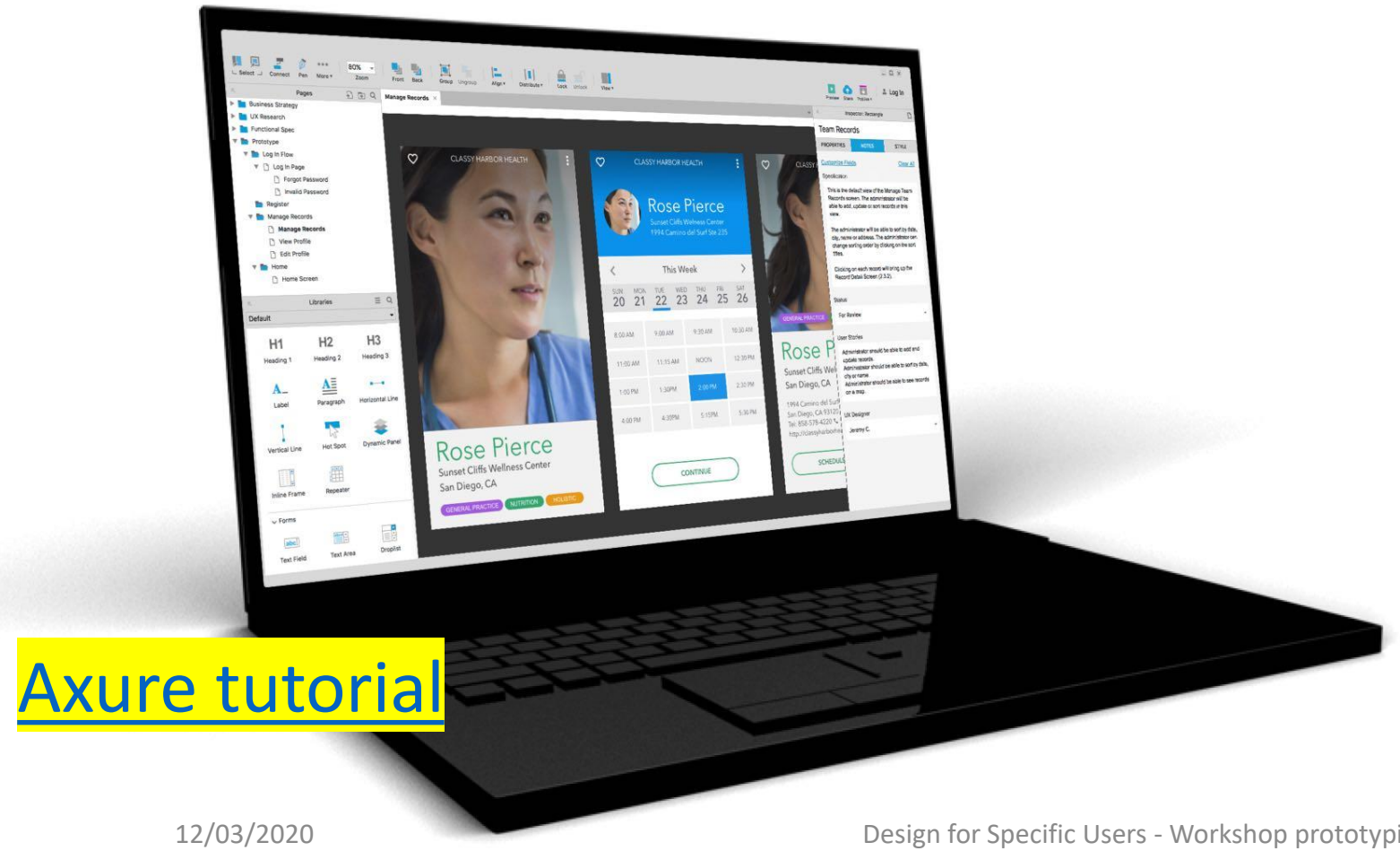


[Getting started with Arduino using LittleBits]

[LittleBits resources & interaction examples]

App mockups

Start creating a prototype (or mock-up) as soon as possible
... and involve the user from **day 1**!



Tools

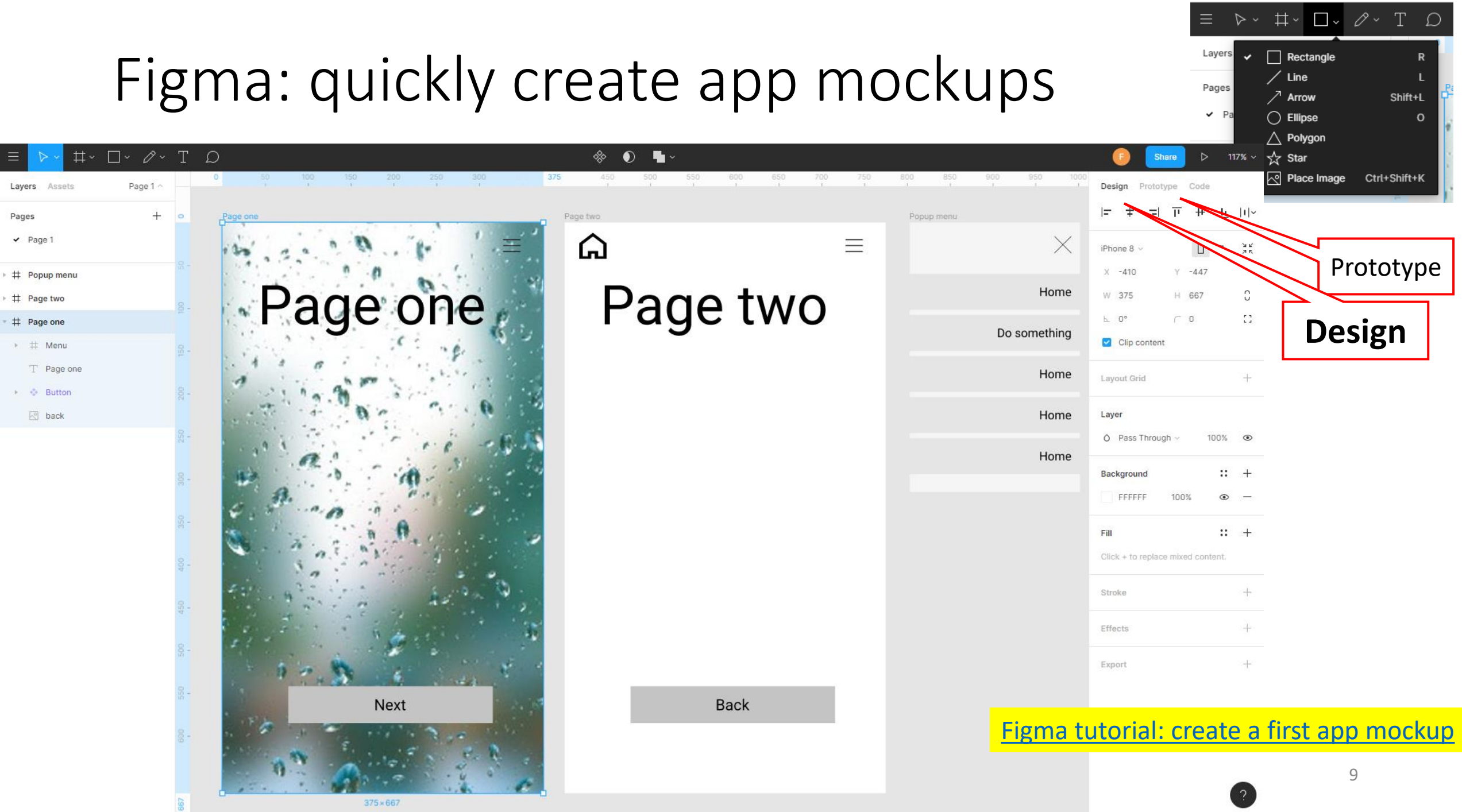
- Sketch (Mac only)
- InVision (Online, free with limitations)
- Figma (Online, free with few limitations)
- Axure (license available from teacher)
- Adobe XD (free)

Mock-up = a prototype (of a user interface) for usability tests

Axure tutorial

Learn more about Userinterface Design

Figma: quickly create app mockups



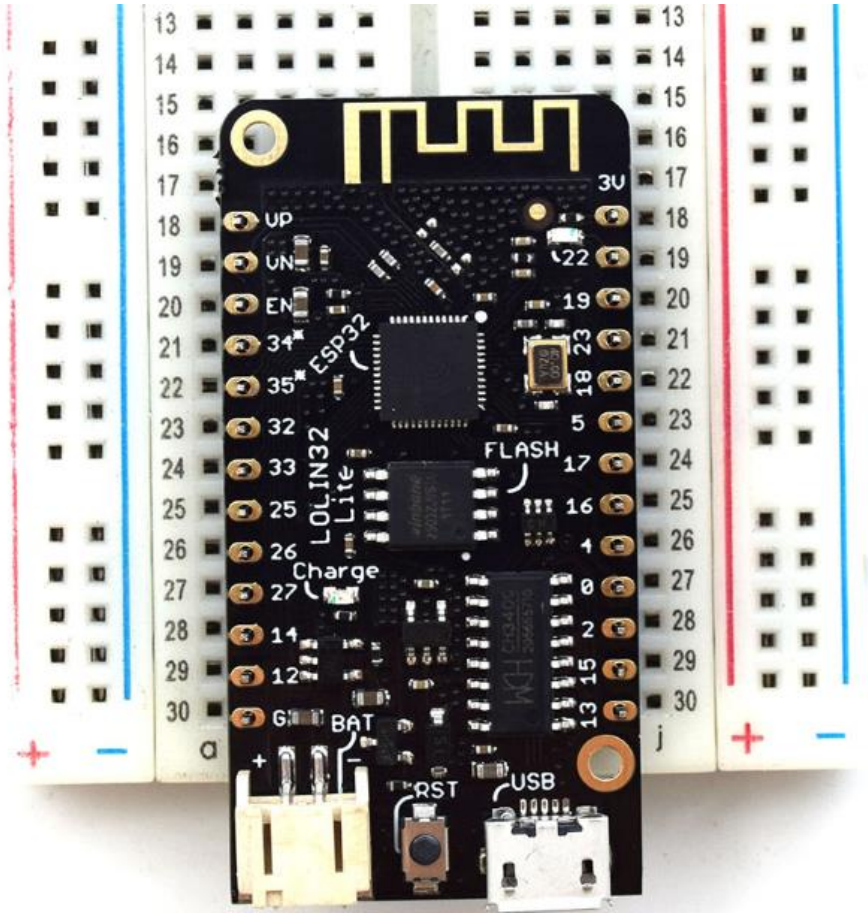
[Figma tutorial: create a first app mockup]

PHYSICAL PROTOTYPES

- Building connected sensors using an ESP module

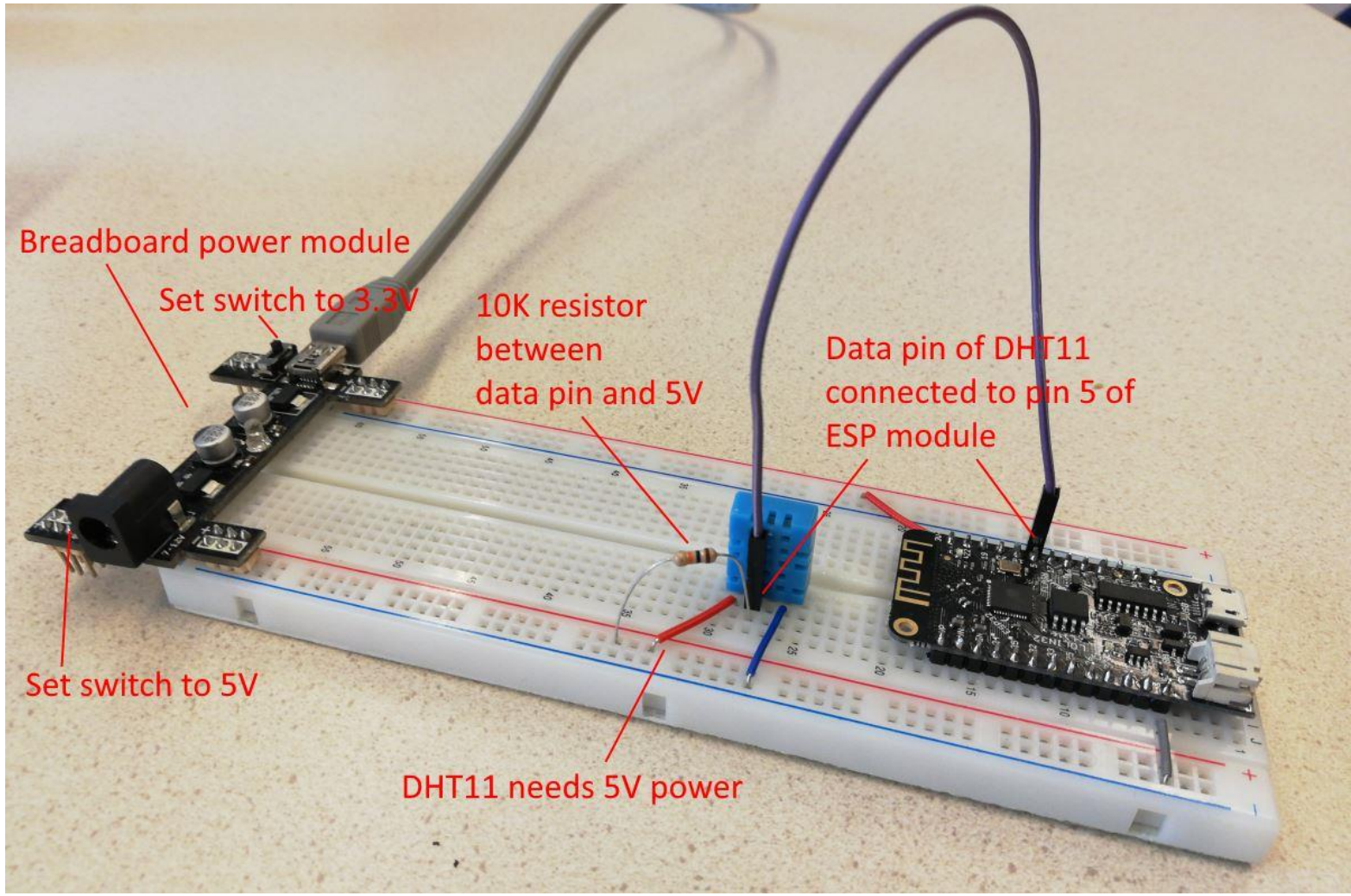
- What is an ESP module?
  - A small microcontroller (~ computer)
  - Can connect & control electronic circuits
  - Has Wifi (and sometimes Bluetooth)

- Combination of electronics & wireless communication



Connect DHT11 temperature sensor to ESP32 module

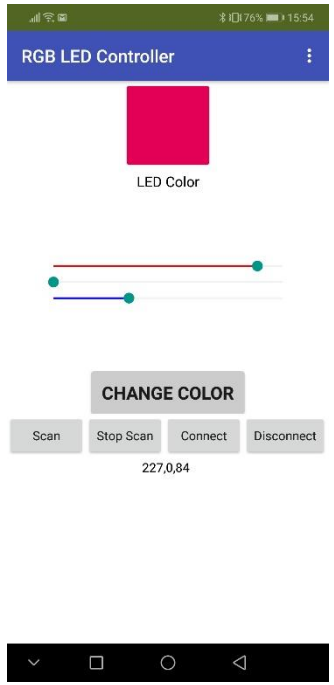
- Build this: "Tutorial: Create a connected sensor"
- It can:
  - Read temperature & humidity
  - Connect to WiFi
  - Store values online

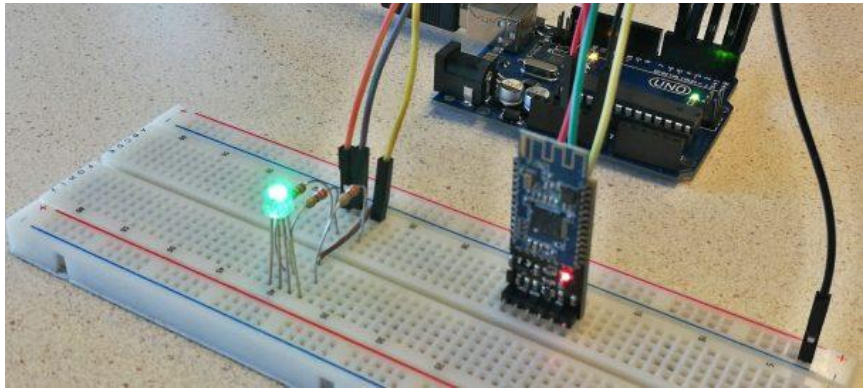


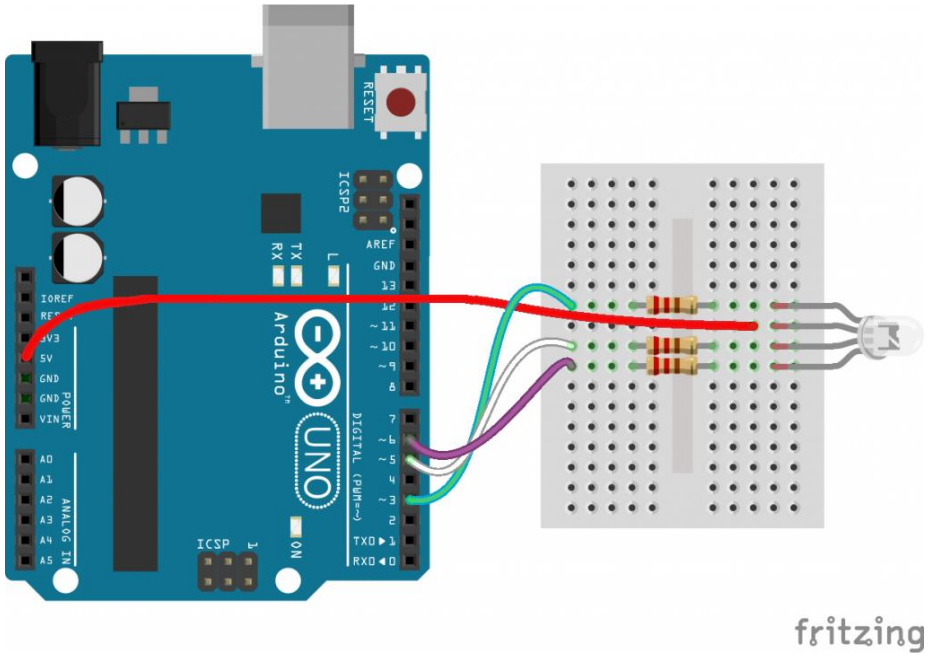
Alternative: use Blynk cloud service

Control an RGB LED from an Android App via Bluetooth

- Build this: "Control an RGB LED from an Android App via Bluetooth"
- It can:
  - Change color of LED from an App
  - Connect via Bluetooth to a phone







MOBILE APP DEVELOPMENT

- Types of Apps
  - Web (Webview)
  - Hybrid
  - Native
- Development environment
  - Online
  - On device (phone or tablet)
  - On desktop/laptop

Web Apps are coded in HTML/CSS/JavaScript. They are served through the internet and run through a browser.
+/- Access Native APIs
- Distribute through App Stores
+ Run on multiple platforms

Hybrid Apps are coded in HTML/CSS/JavaScript. They are run through an invisible browser (webview) that is packaged into a native application.
+ Access Native APIs
+ Distribute through App Stores
+ Run on multiple platforms

Native Apps are coded in the native language of the device (e.g. Objective-C for IOS, Java for Android). They are run directly on the device.
+ Access Native APIs
+ Distribute through App Stores
- Run on multiple platforms

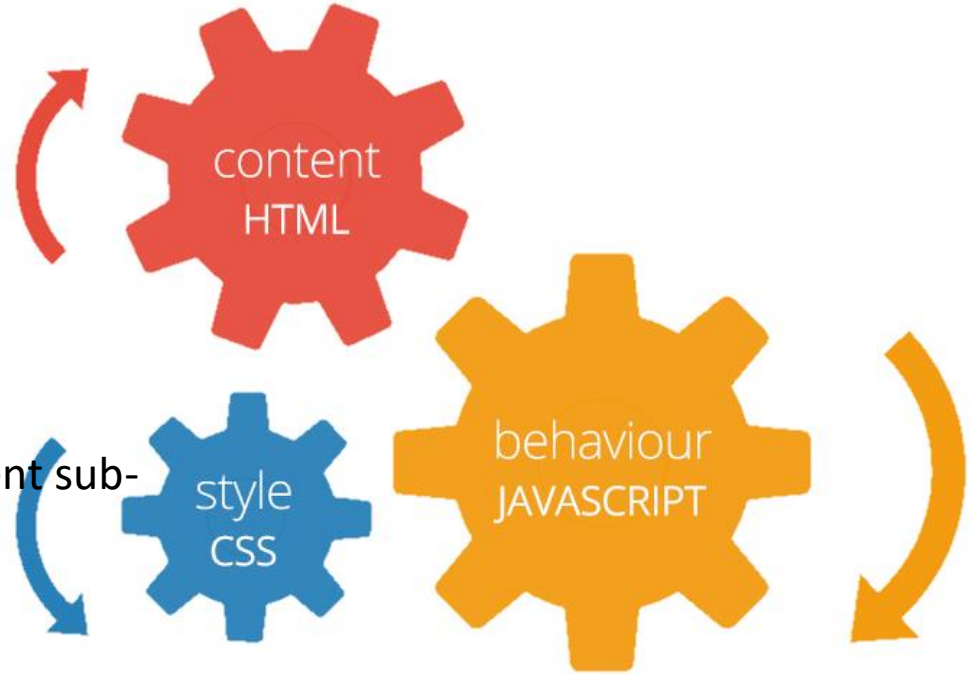
App generators

- App Inventor (online service)
- Sketchware (create Apps on your phone)
- Tasker+Tasker App Factory (Paid App)

Sketchware and App Inventor use ***Block Language*** programming

Graphical User Interface Builder

Hybrid Apps



- Lots of competition
- HTML/CSS/JavaScript based
- Often immature (version hell, poor documentation, different sub-platforms, mixed-up online info)

- Requires: intermediate programming knowledge, and web tech (HTML/CSS/JavaScript)
- Tools: Code editor/IDE e.g.: Atom, Visual Studio Code, Brackets
- Learn basics: course Web Tech, App Dev, w3schools

Hybrid Apps Frameworks

Based on Web Technology

- Ionic
- NativeScript (delivers Native App)
- PhoneGap (Adobe)
- React Native (delivers Native App)
- Xamarin (Microsoft, C#) (delivers Native App)
- jQuery Mobile

Ionic	NativeScript
Quick build & test	Build/test takes longer (wait each time you change)
Preview in browser or on Phone	Preview on Phone or Emulator
App: renders in Webview	Converts to native (small performance gain)
Create UI based on HTML/CSS	Create UI with own elements

Skill level: medium - pro

Native apps

developer.android.com
developer.apple.com

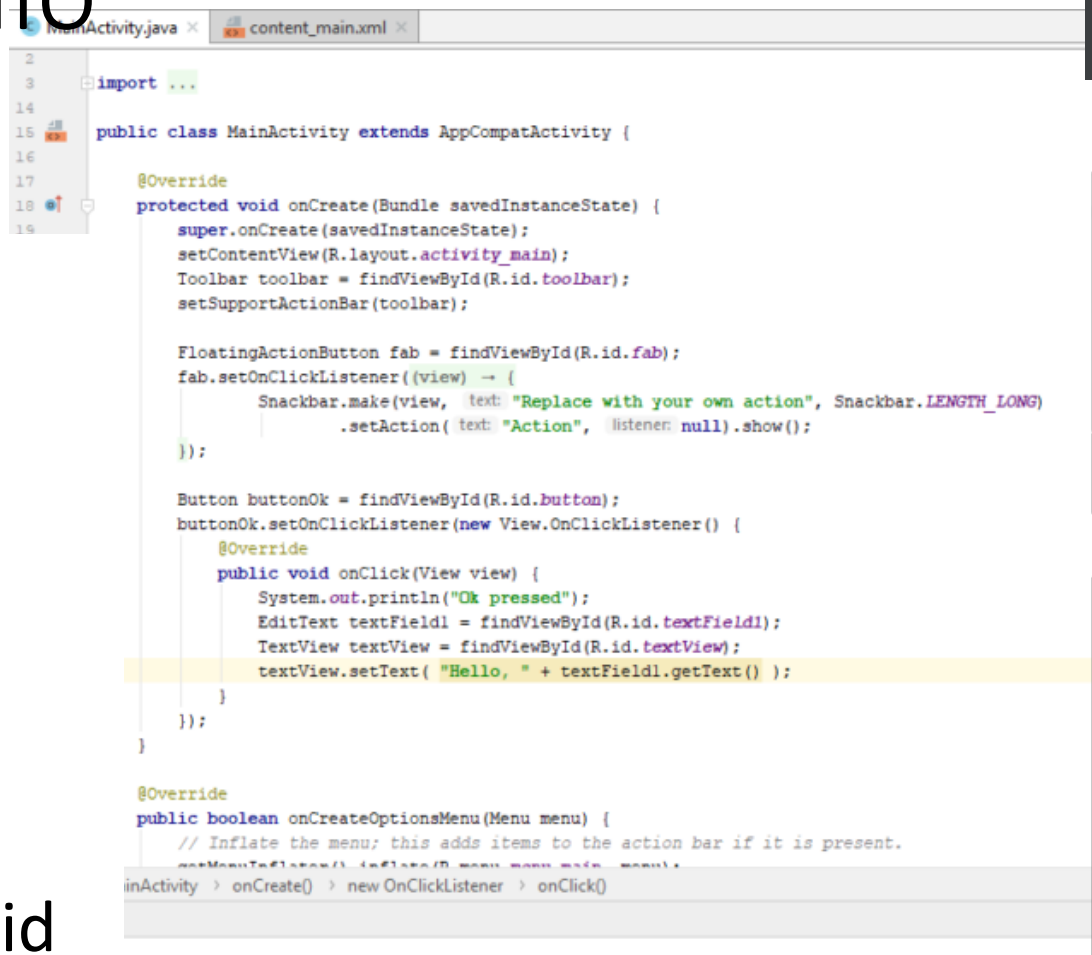
- Platform: IOS (Apple), Android (Google)
- Android:
  - Andoid Studio (Java)
- IOS
  - Xcode (Objective-C)
  - Swift

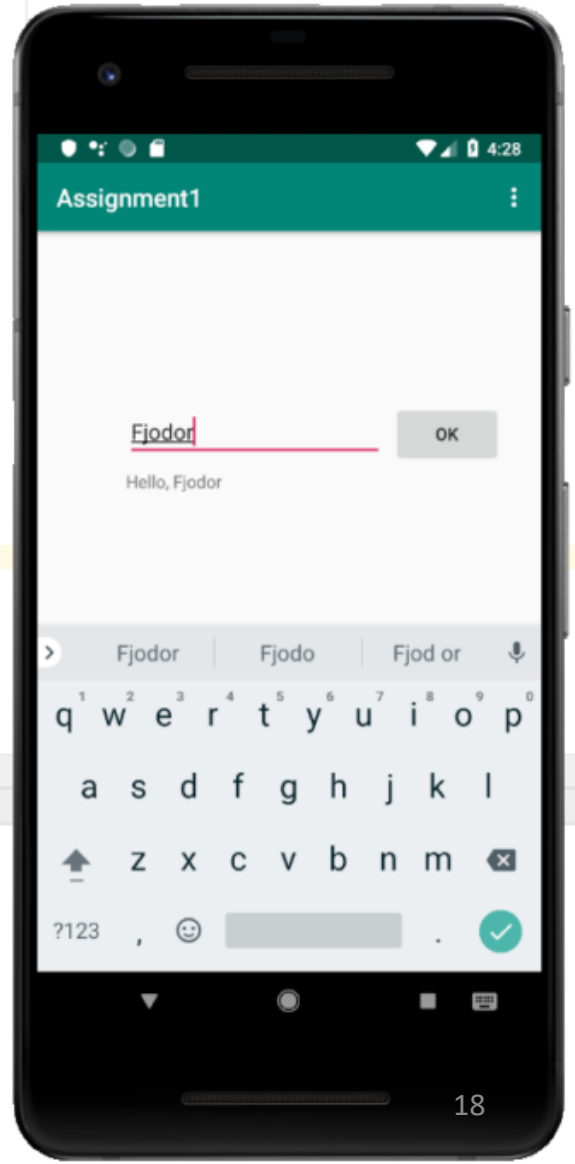
- Requires: programming knowledge, and preferably web tech
- Learn IOS: Everyone can code, App development with Swift

Android Studio (Java)

Skill level: medium - pro

- Tutorials:
  - Android Developer fundamentals
  - developer.android.com
  - more
- Requires: Java programming experience
- Learn basics of Android Studio:
  - Getting started
  - Create a first interactive userinterface





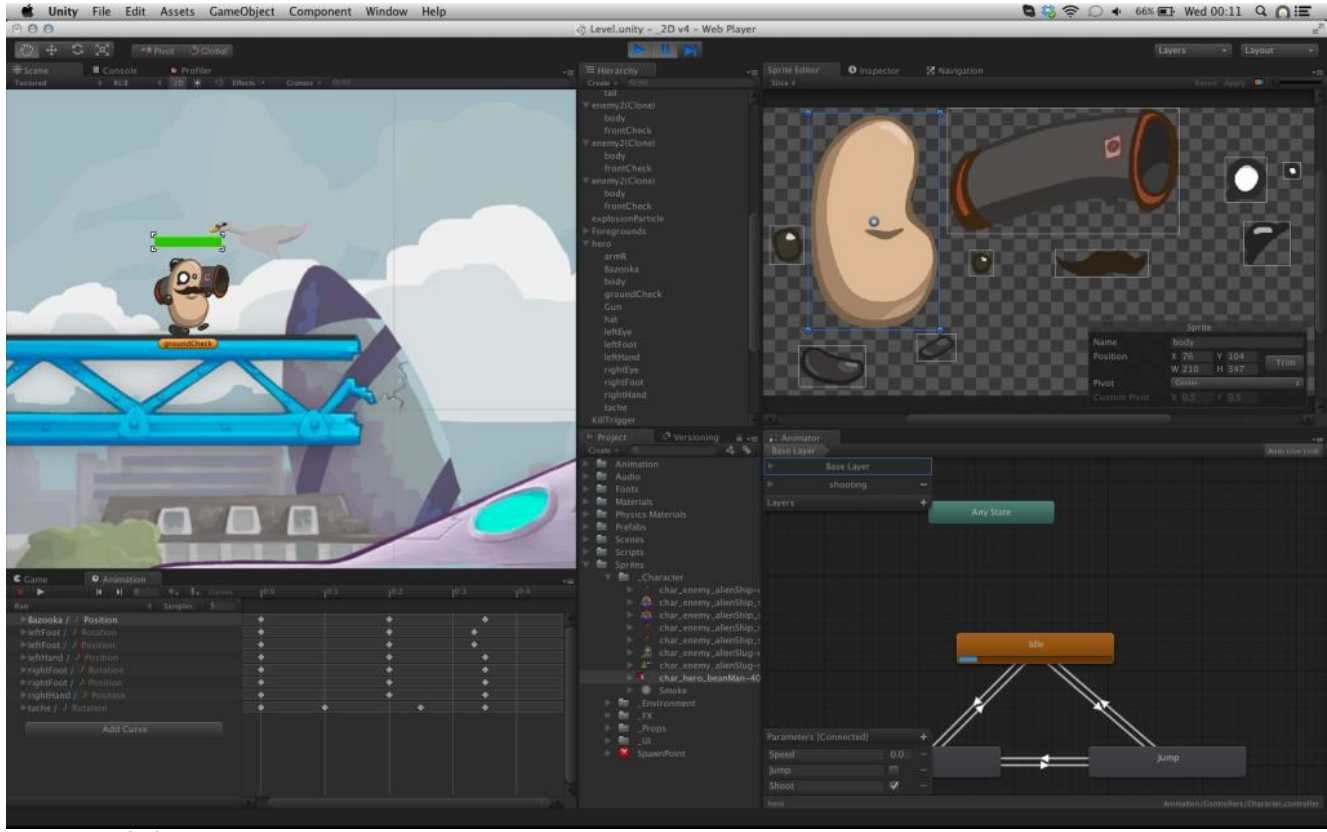
Publish in App store (deploy)

- Requires:
  - IOS: register ($99) with developer.apple.com/programs (more info)
  - Android: register (for free) with play.google.com/apps/publish
- Alternative: **test without publishing**
  - Android: distribute generated .apk to testers
  - IOS: have testers get a build via TestFlight: register participants on your team

Game engines

- Combine with Native App:
  - Unity: Mobile solutions
  - Unreal: Mobile Development
- Combine with Hybrid framework:
  - Phaser.io:
    - Deploy with Cocoon.io
    - Combine with: Ionic, PhoneGap

Learn more about game development:
emanueleferonato.com
html5gamedevs.com
gamedev.net



Intro to App Inventor

ai2.appinventor.mit.edu

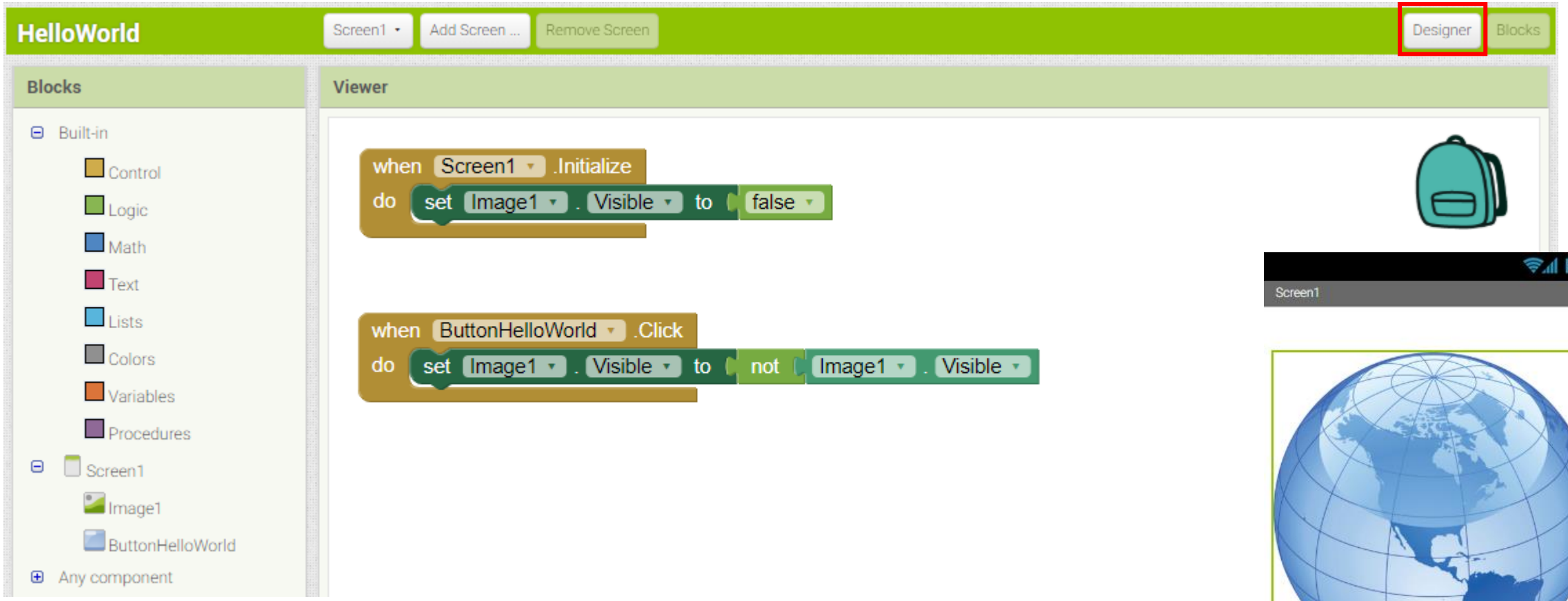
Program

[App Inventor tutorial](#)
[App Inventor 2: Create your own Android Apps](#)

Programming using Blocks

ai2.appinventor.mit.edu

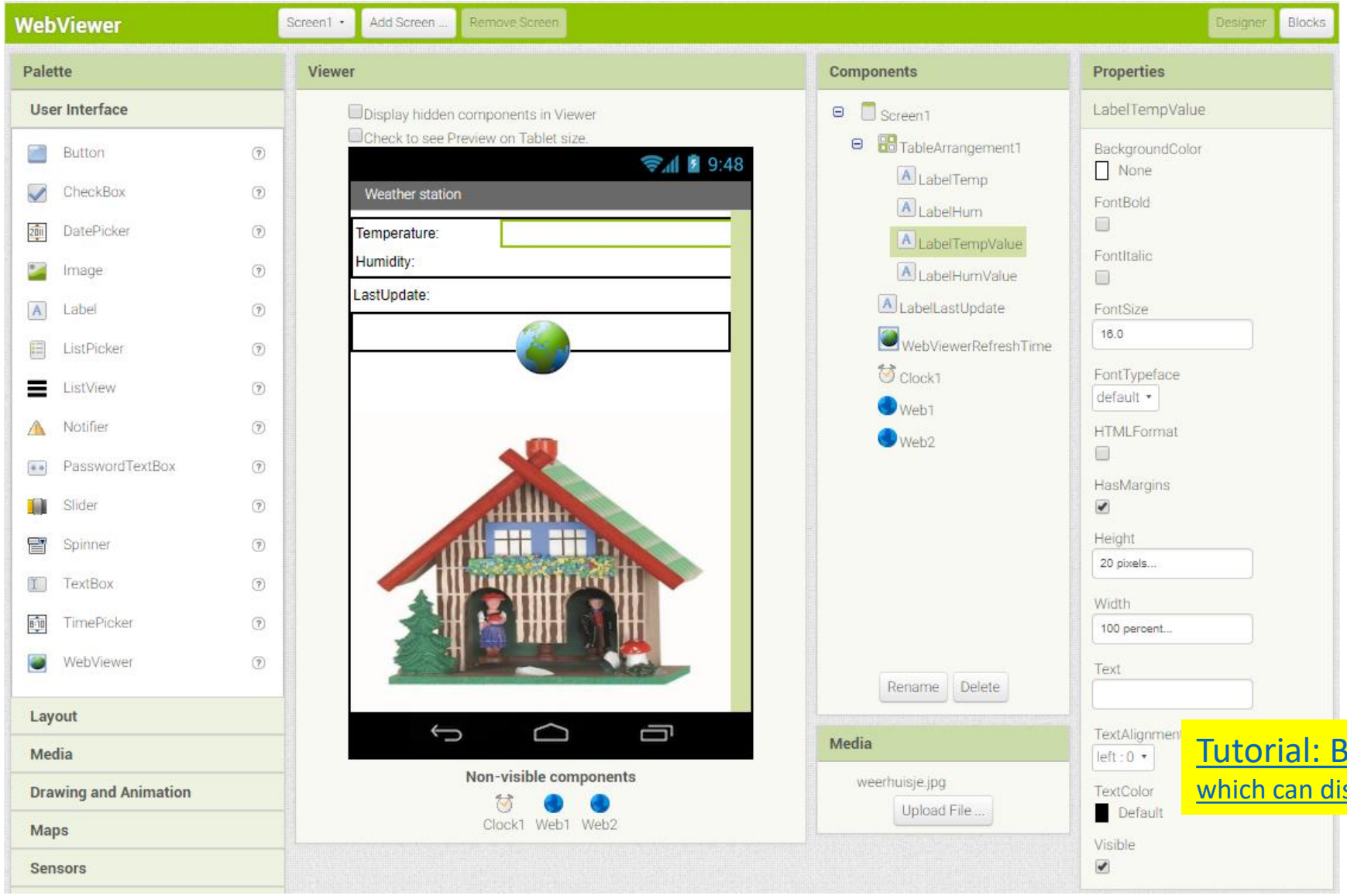
Designer

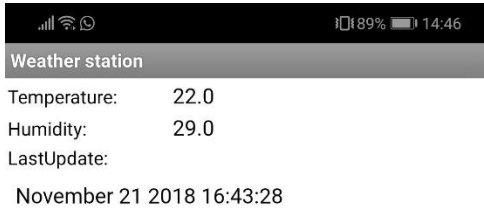


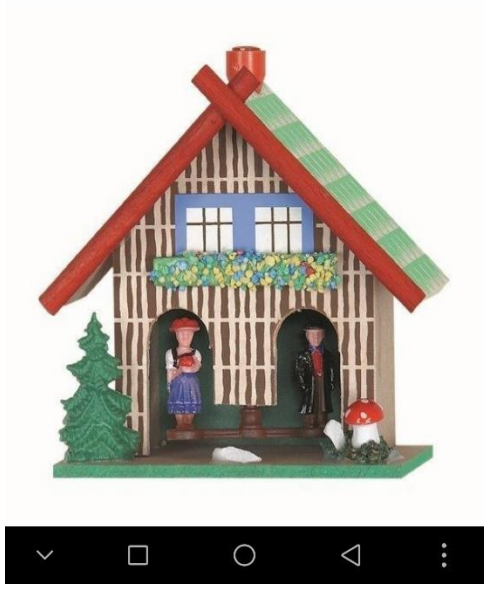
App Inventor tutorial
App Inventor 2: Create your own Android Apps

ai2.appinventor.mit.edu

Tutorial: Creating a connected App







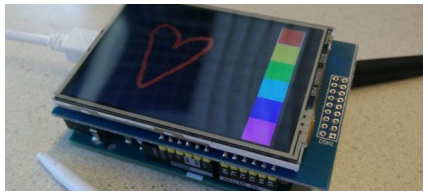
[Tutorial: Build an App with App Inventor](#)
[which can display values of a connected sensor](#)

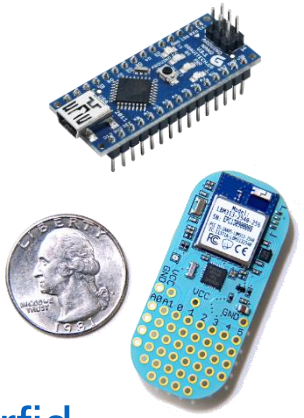
BORROW
MATERIALS

- Arduino** + EL-kit*
- LittleBits kit

Check out the contents of the SmartProducts (M4) kit here

- Arduino Leonardo/Nano/Wemos D1, ESP32, Wifi module, Bluetooth module, Blue bean
- Touchscreen, displays
- Motors, vibration motor
- Mp3 module, speakers, sd-card
- Sensors: distance, color, accelerometer/gyro, temp, rfid, rotation (in Lego motor)
- RGB-leds





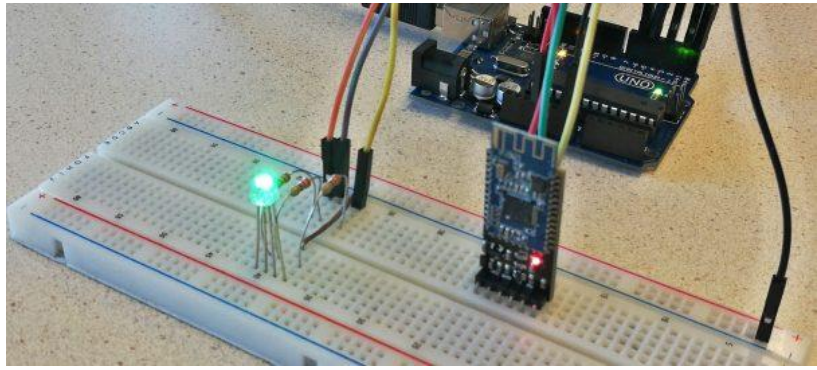
Tip: buy cheap components @ STORES Stores-shop, Educafé, Zilverling (Starbucks)

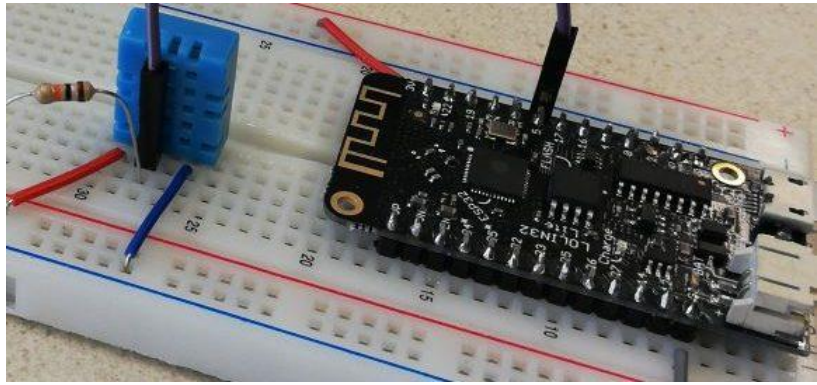
This presentation & tutorials available at: vanslooten.com/workshop

* contains breadboard, wires, Arduino Nano, power module, buttons, leds
** contains Arduino Uno EVShield, ESP Wifi module, 2 sensors, battery holder etc

LET'S DO…

- Practice building a prototype:
  - Create a connected sensor
  - Control an RGB LED from an Android App via Bluetooth
- Create a ‘virtual’ prototype
  - Do Axure or Figma tutorial
- Build an App
  - Creating a connected App using App Inventor
  - Create an App with Android Studio





Contact: f.vanslooten@utwente.nl, W241

This presentation & tutorials available at: vanslooten.com/workshop
Use examples and references to tutorials in this presentation

QUESTIONS?

vanslooten.com/workshop

f.vanslooten@utwente.nl
W241