WRITING MOBILE APPS FOR INVESTIGATIVE STUDY OF PHYSICS

Session 2

Objectives

- This course aims to introduce how Cloud Computing can be used to facilitate e-Learning in schools. After finishing the course, participants should be able to:
  - use common free-of-charge development tools or platforms, such as the web-based Apps Inventor or other easily accessible development tools, to develop Mobile Apps for smartphones or tablet PCs so that these devices could be used as data loggers for Physics experiments;
  - acquire basic programming skills of writing simple Mobile Apps to invoke the embedded sensors, record and export data for Physics demonstrations / experiments by using free-of-charge Mobile Apps development tools or platforms mentioned in (a);

Objectives

- Understand the properties of the many sensors embedded in mobile devices such as GPS, compass, accelerometers, etc. and master the basic techniques to program these sensors, activate them for use in Physics experiments, and extract and export data for in-depth analysis;
- design some investigative experiments with worksheets and sample measurements for Physics by using the custom-made Mobile Apps developed by the teachers;
- exploit the potential of students and develop their creativity by guiding them to write Mobile Apps for investigative study of Physics or applications of Physics to improve living; and
- appreciate how knowledge and skills related to science (in particular Physics), technology and mathematics can be appropriately integrated in learning and in applications in the real world.

Rundown

- Discussion of Homework
- Additional Programming Skills (From Session 1)
- Exporting Data from Mobile Apps
- Bluetooth Programming
- Connecting with other devices from App Inventor
- Doing experiments with Mobile Apps and other devices
- Other possibilities
- Conclusion

2.1 Discussion of Homework

- What did you planed?
- Any difficulties encountered?

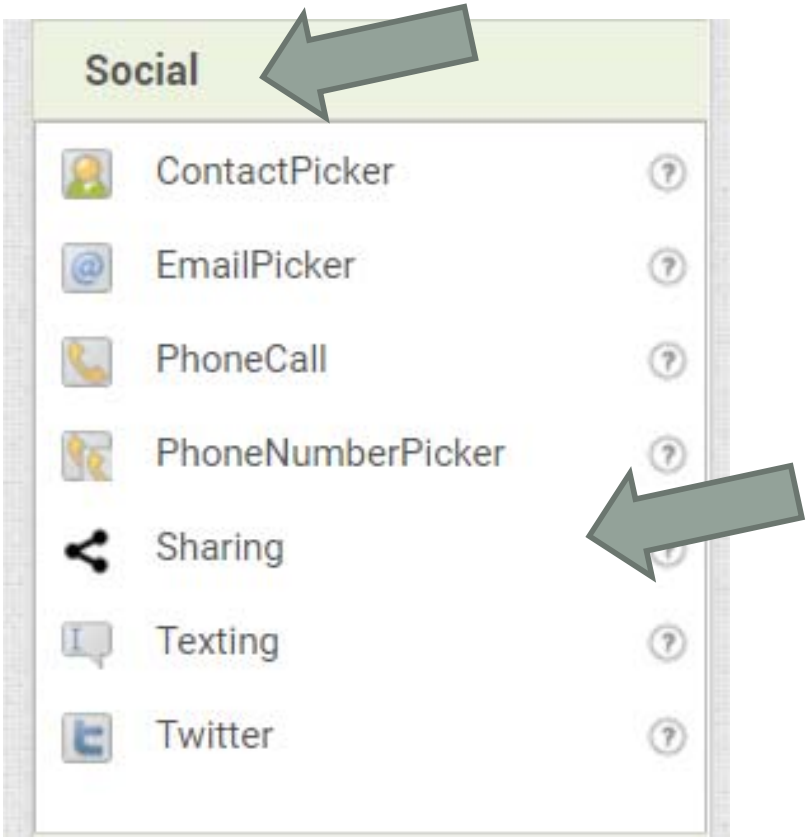
Additional Programming Skills

- From Session 1
  - IF-THEN-ELSE
  - Exporting to APK file
  - Calling Procedures from CAITE_CUHK_TEMPLATE1

- We will refer back to the slides in Session 1

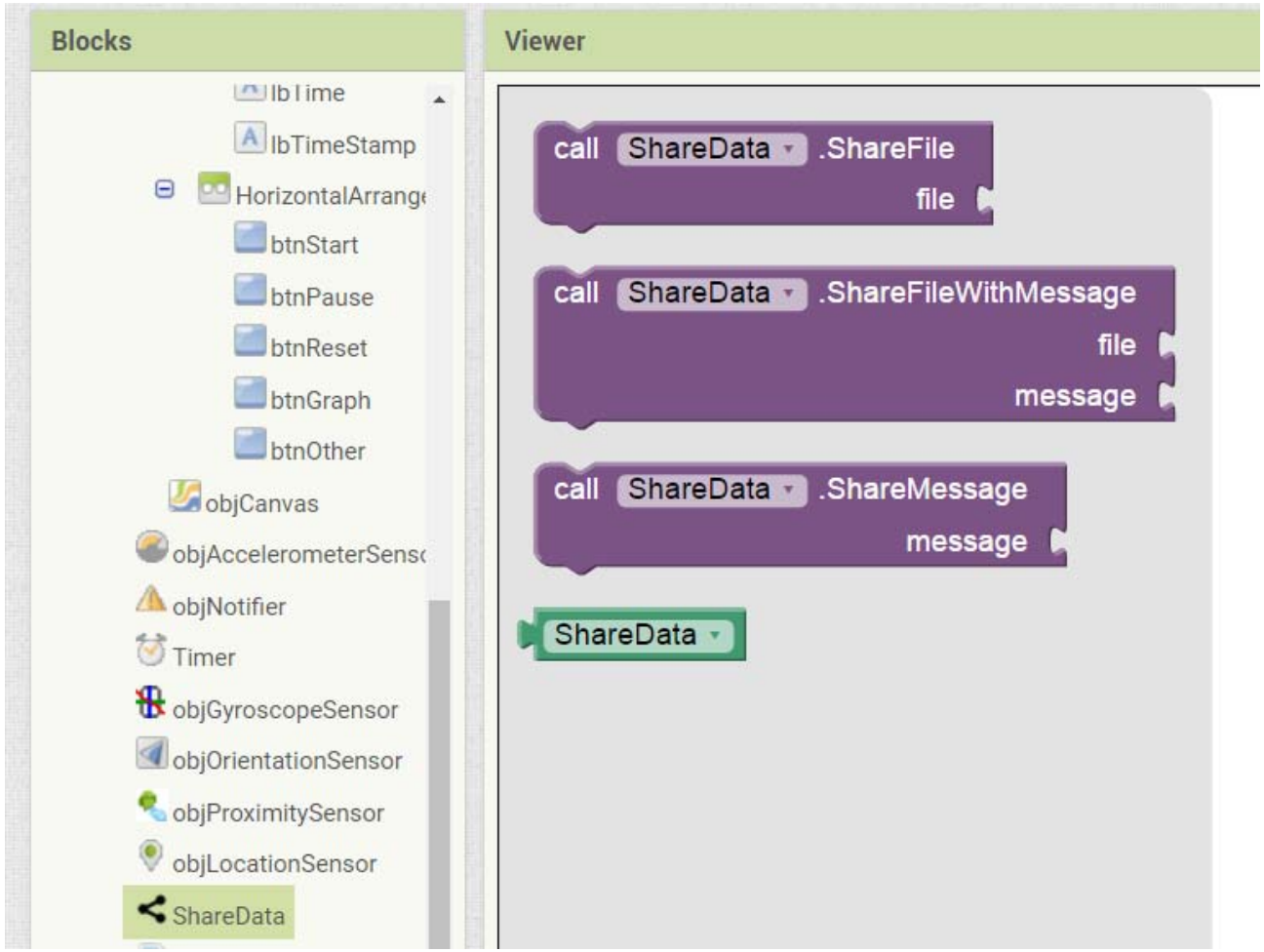
Exporting Data from Mobile Apps

- Data in App Inventor (e.g. Photos taken, video taken etc) through the Sharing component



Exporting Data from Mobile Apps

- The Sharing Component comes with 3 simple procedure



Exporting Data from Mobile Apps

- TWO precautions on using the Sharing component
  - your mobile phone should have related apps setup and configured
    - Example supported Apps:
      - Google Drive, Gmail, Whatsapp etc
  - you must configure the filename carefully
    - The file path can be taken directly from other components such as the Camera or the ImagePicker
    - but can also be specified directly to read from storage. Be aware that different devices treat storage differently, so a few things to try if, for instance, you have a file called arrow.gif in the folder Appinventor/assets, would be:

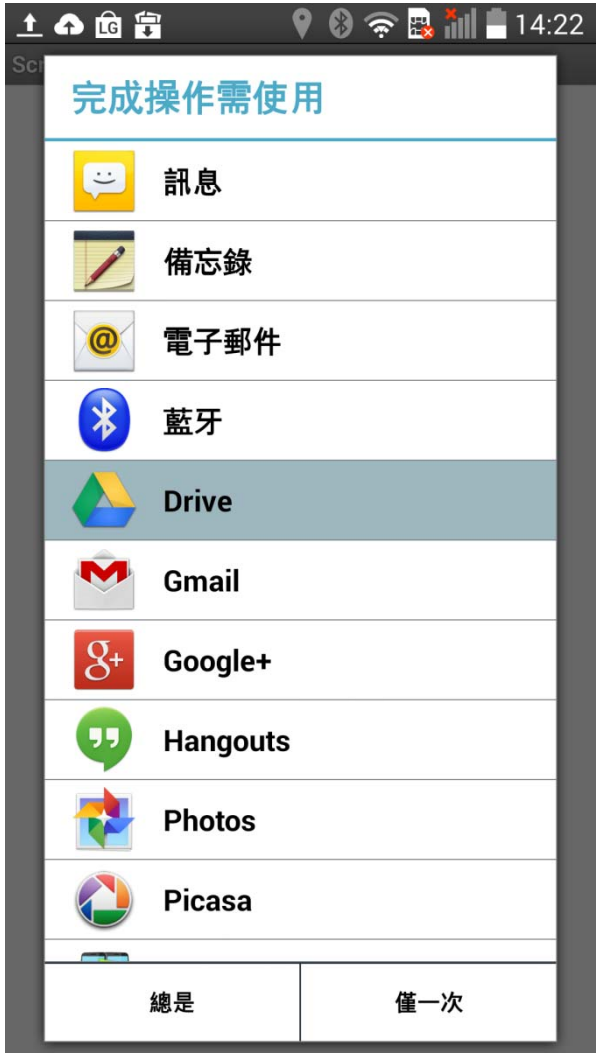
      file:///sdcard/Appinventor/assets/arrow.gif
    -

Exporting Data from Mobile Apps

- If you are using the CAITE_CUHK_TEMPLATE**2**
  - you may call the procedure exportData to save the collected sensor values and share the saved file (CSV) to Google Drive or Email in 1 step

Exporting Data from Mobile Apps

- Example Sharing Component in Action

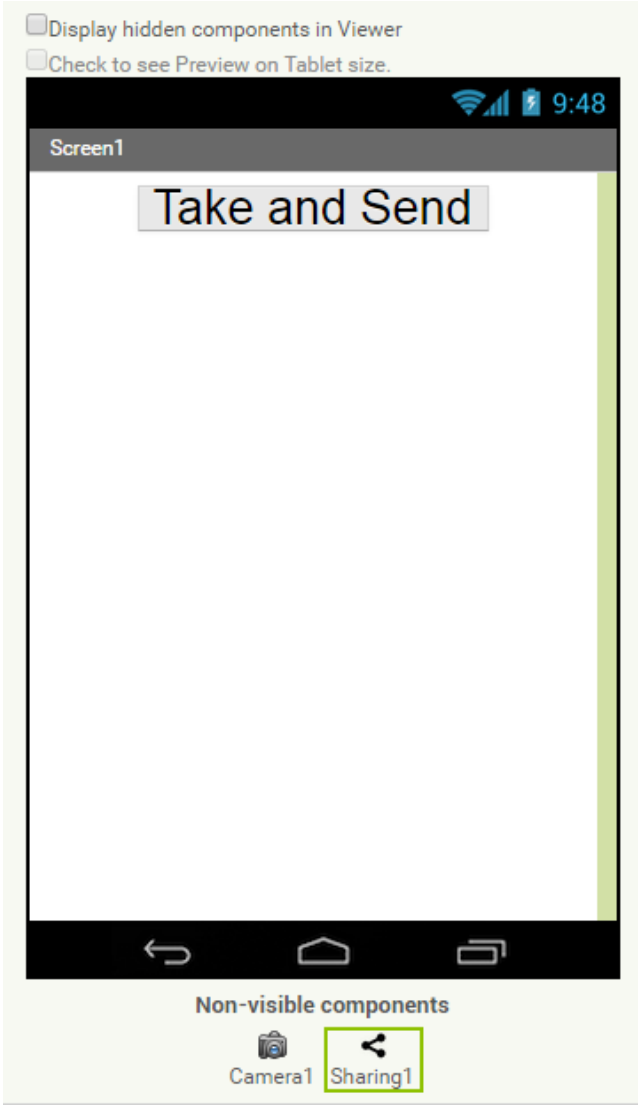


Exporting Data from Mobile Apps

- Let's build a simple example application:
  - Take a photo using the camera
  - Save the photo
  - Share the photo using the Sharing Component

Exporting Data from Mobile Apps

- Step 1:
  - Design a very simple interface
    - 1 button
    - 1 Sharing Component
    - 1 Camera Component (from the Media category)
  - Rename the component if you want

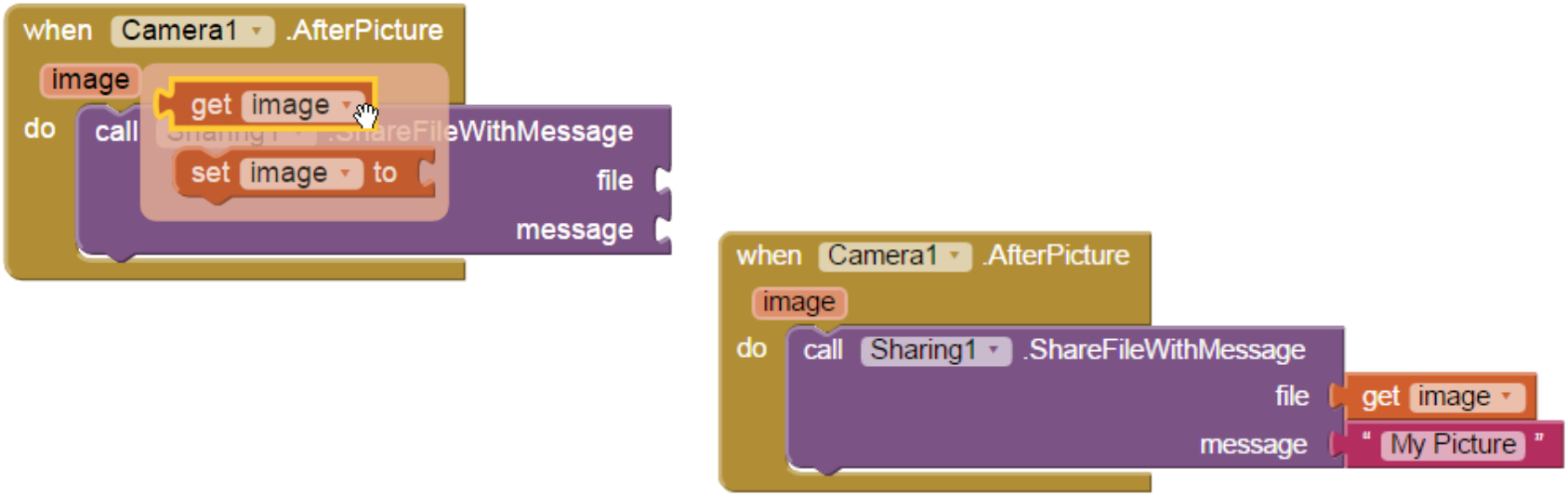


Exporting Data from Mobile Apps

- Create the event program for the button
- By making use of the TakePicture procedure of the Camera component, we can ask the Camera component to take picture when the button is clicked.

Exporting Data from Mobile Apps

- The Photo taking process may actually take a long time
- We have to make use of the event AfterPicture of the Camera component to share the photo to others
- The image variable store the FILENAME of the photo taken by the Camera component

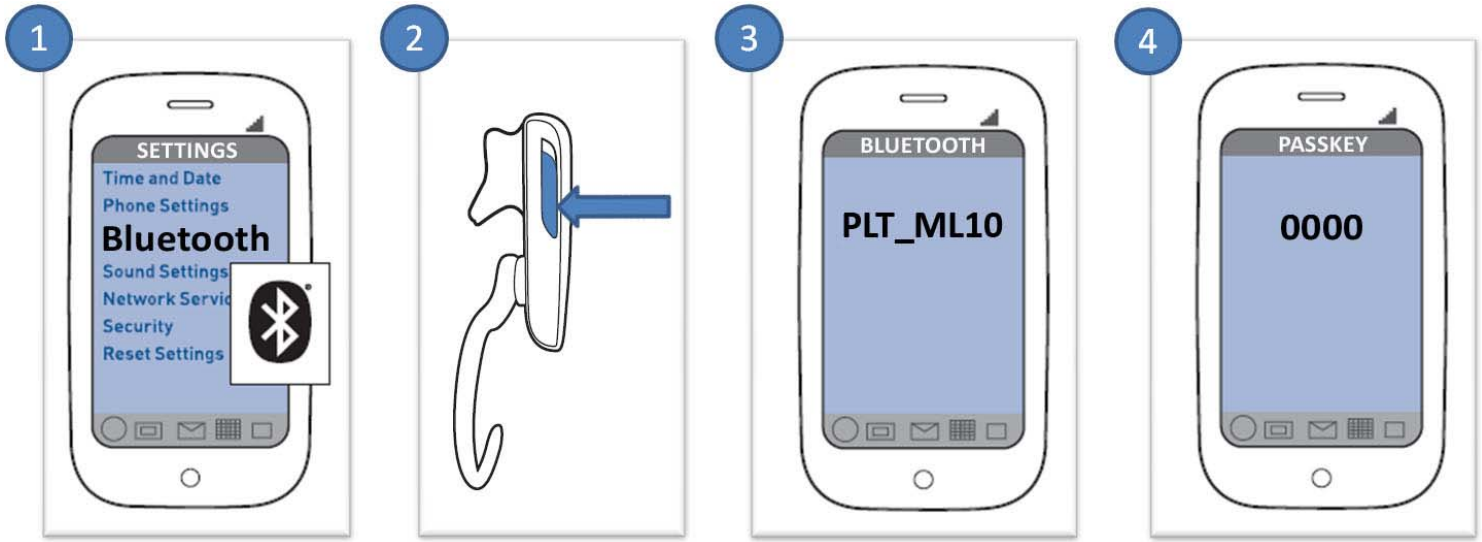


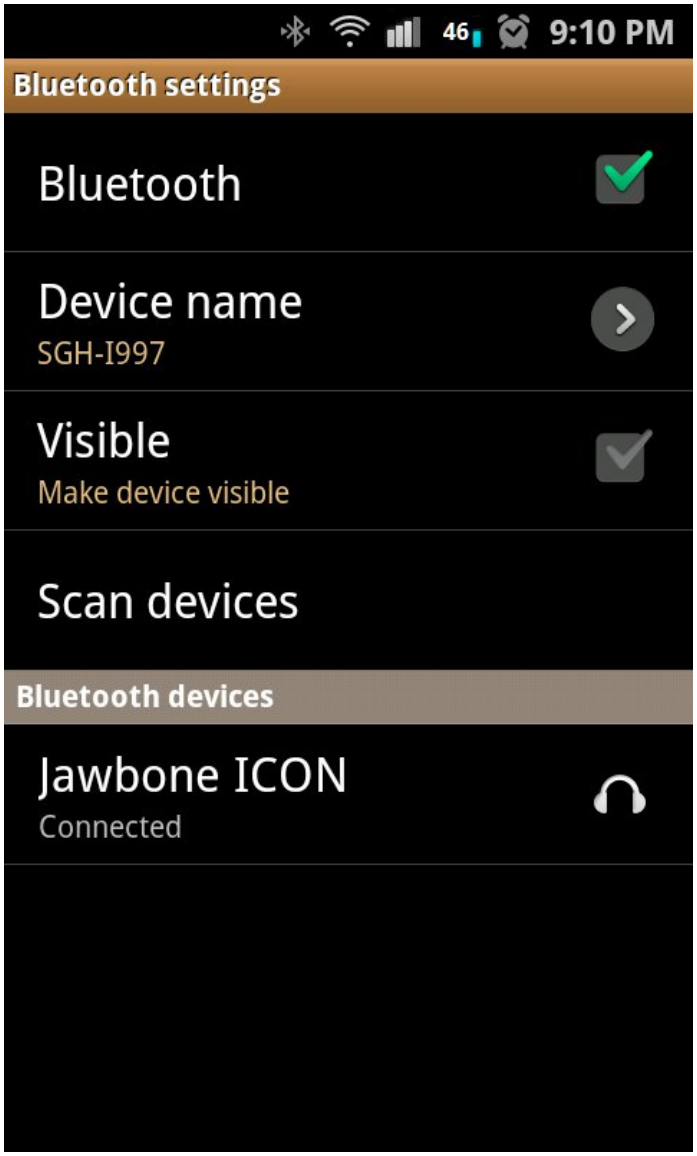
Try without building your own?

- You may download the file myCamera.aia from the course web page and try it out if you do not want to build the app from scratch

2.3 Bluetooth Programming

- Bluetooth is a wireless technology for connecting devices together through short distance (<10m)
- It can be used to connect to other devices from within App Inventor
- Before connecting to other devices, pairing must be correctly done

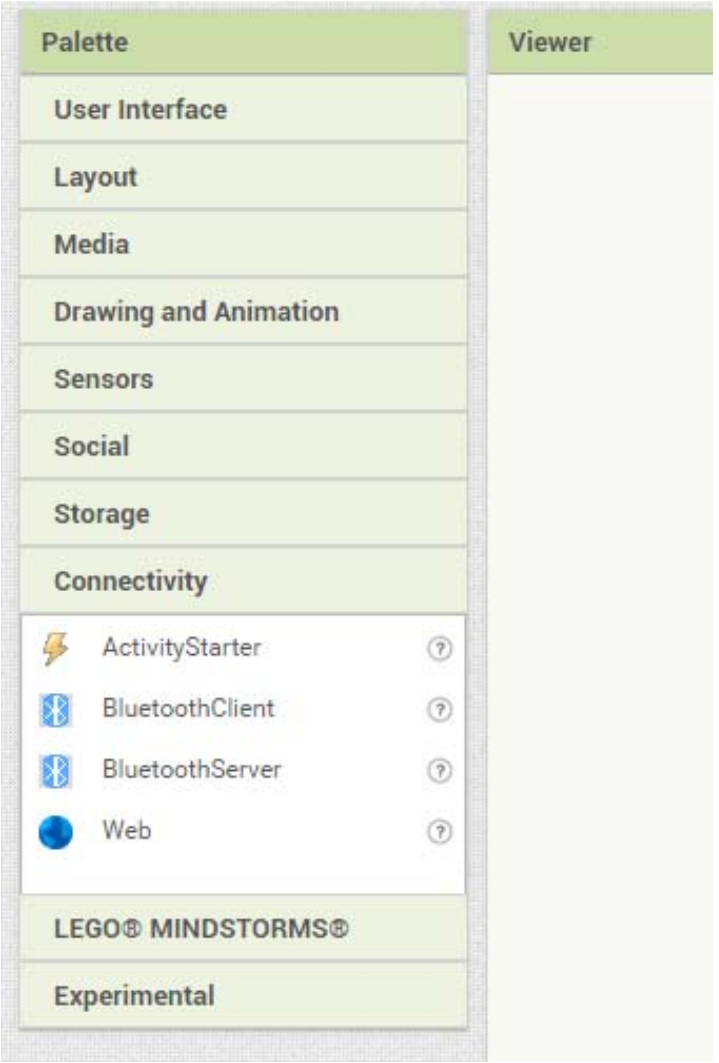



9:10 PM
Bluetooth settings
Bluetooth
Device name
SGH-I997
Visible
Make device visible
Scan devices
Bluetooth devices
Jawbone ICON
Connected


1:45 PM
Bluetooth pairing request
Type PIN to pair with "Scanfob-20675ADC6F". (Try 0000 or 1234.)
0000
OK
Cancel
ALT
ABC
Done

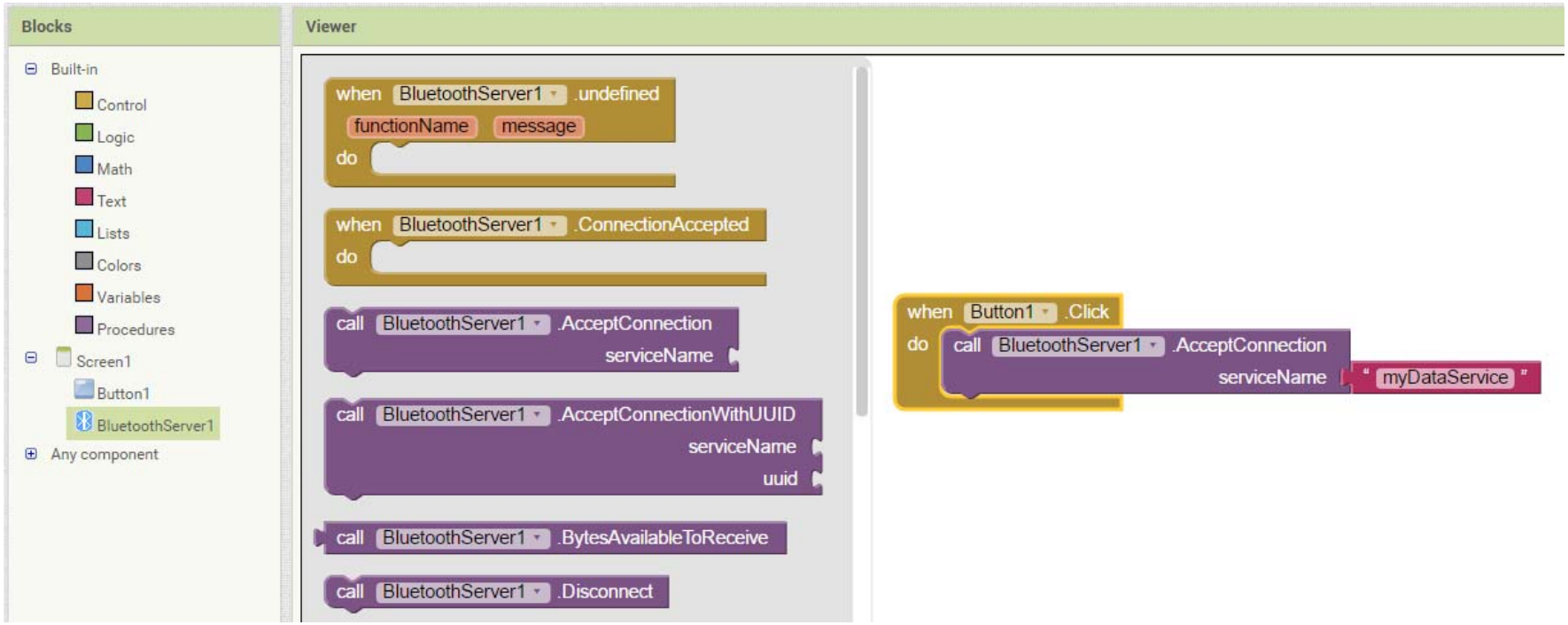
2.3 Bluetooth Programming

- Two components are used to make Bluetooth connection
- Bluetooth Client
  - Component used to connect to other devices waiting for connection
- Bluetooth Server
  - Component used to wait for incoming Bluetooth connection



2.3 Bluetooth Programming - Server

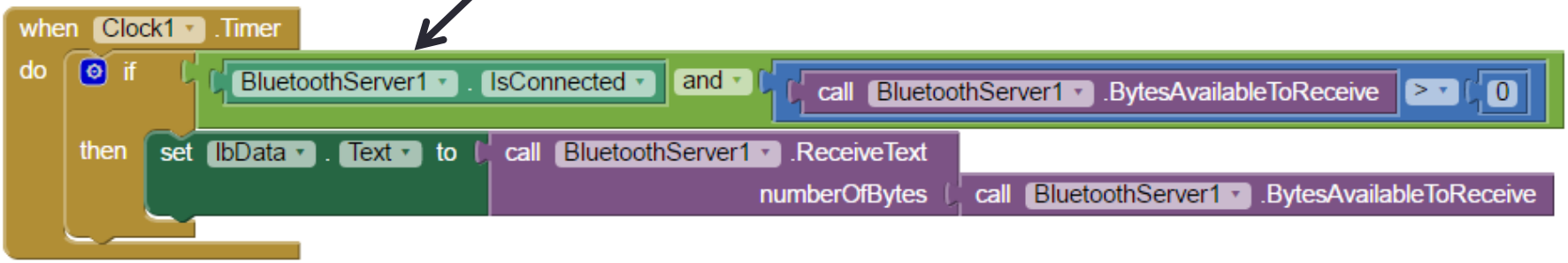
- App Inventor may start to accept connection from other Bluetooth devices by using the AcceptConnection Block from the Bluetooth Server



2.3 Bluetooth Programming - Server

- Once connected, the Bluetooth server may send or receive data
- A Clock can be used to check if there are any data available and retrieve it from the Bluetooth Server

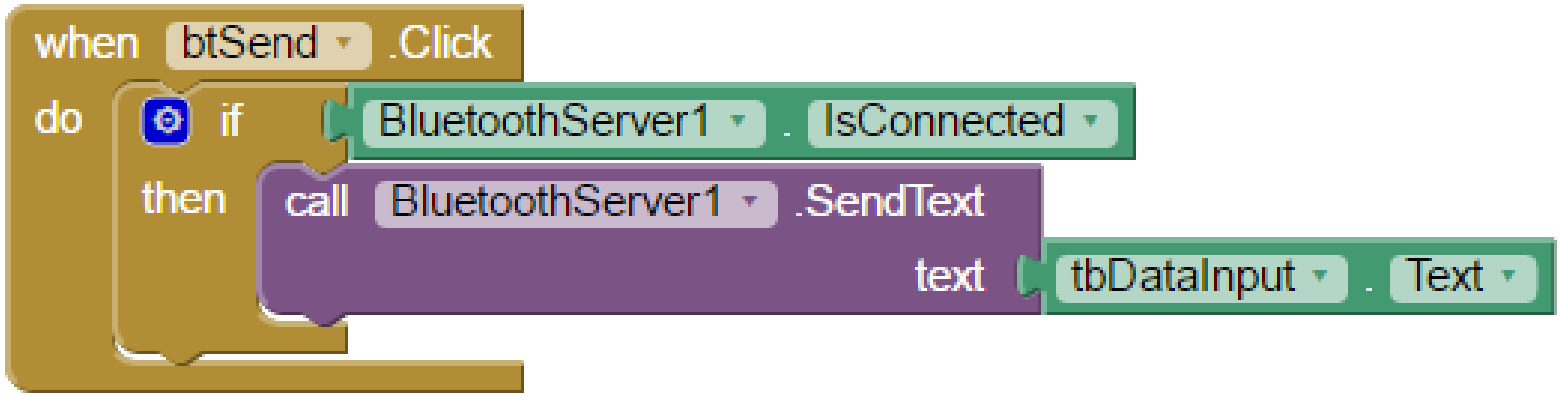
If condition is used to ensure that data receiving is done only when the Bluetooth connection is working and there is available data



The Bluetooth Server component provides the ReceiveText procedure for receiving data from the other side

2.3 Bluetooth Programming - Server

- To Send data to the connected client, we can use the SendText Procedure from the BluetoothServer component

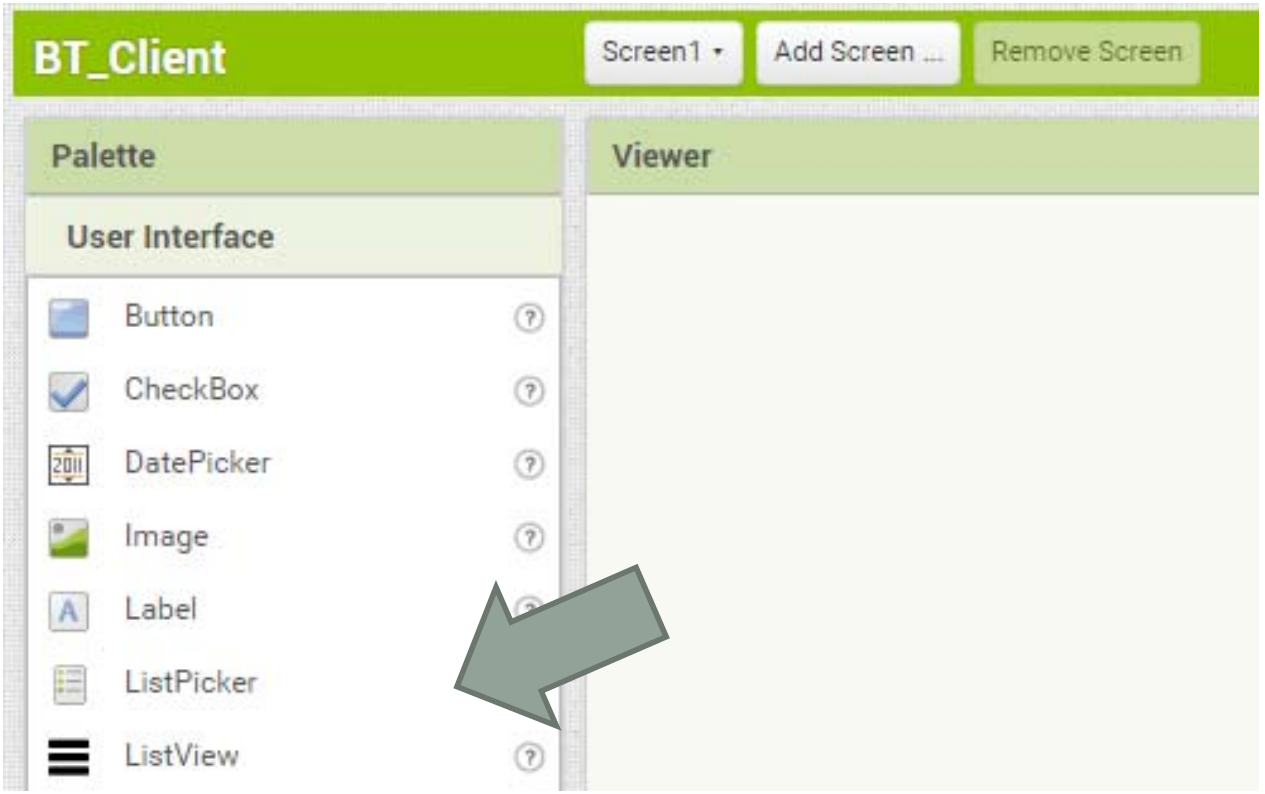


2.3 Bluetooth Programming - Client

- The Bluetooth Server app will not work on its own, we need another App to connect to it – the Bluetooth Client
- The BluetoothClient component works similarly to BluetoothServer component. The ONLY difference is that the BluetoothClient will initiate the connection to the BluetoothServer by using its address

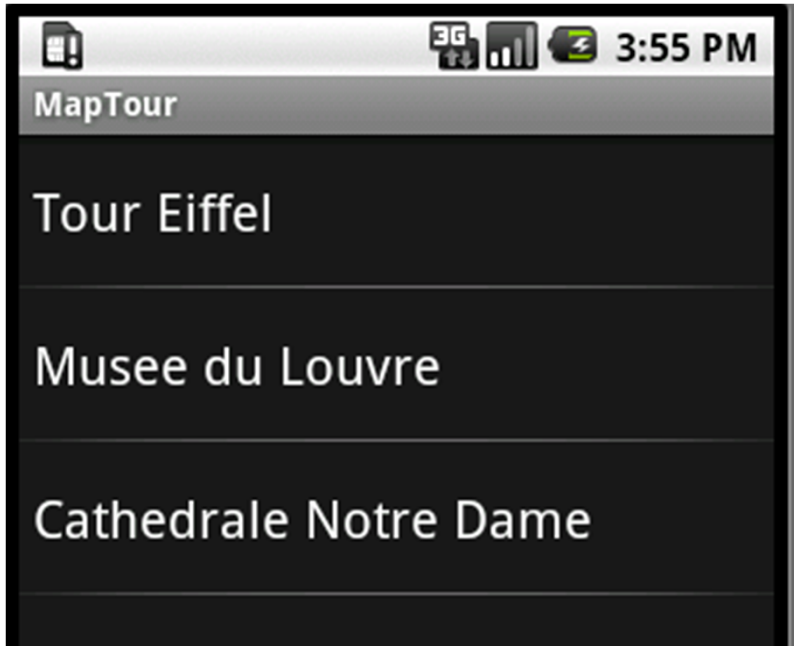
2.3 Bluetooth Programming - Client

- To make the choice of devices easier, we will make use of the ListPicker component from the UI category



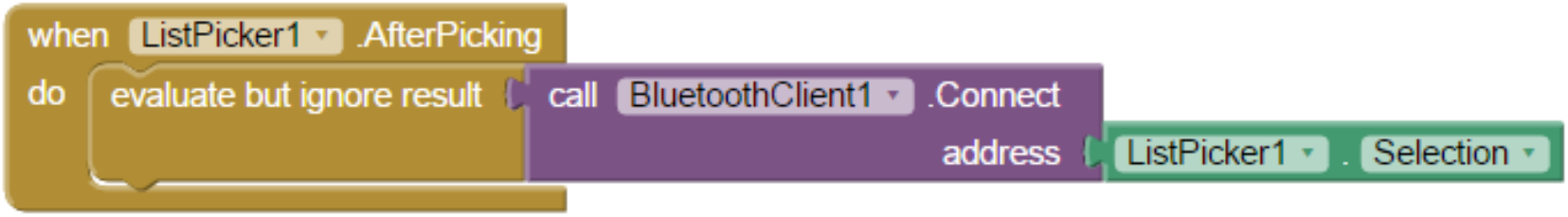
2.3 Bluetooth Programming - Client

- The ListPicker looks like a button, except that it will shows a list of choices when you click on it



2.3 Bluetooth Programming - Client

- The ListPicker comes with TWO useful events
  - BeforePicking
    - Populate the choices of names and address from the BluetoothClient discovered addresses
  - AfterPicking
    - Connects to the BluetoothServer selected by the user using the address from the list



Let's try

- You may download the following files and try it out.
  - BT_Server.aia
  - BT_Client.aia
- You cannot run both applications together, please find a partner

2.4 ADVANCED SENSORS EXTENSION AND OTHER CONTROL UNITS

Limitation of the sensors on the mobile phone

- Some sensors are not available in App Inventor
- Physical limitation: difficult to locate the position of sensors inside the mobile
- Limited outputs to control physical units in different experiments

Introduction to the Arduino platform and related electronics

- Arduino: microcontroller
- Essentially low-powered computers on a chip
- Digital Inputs/ Outputs (5V / 3.3V)
- Analog Input (10-bit analog to digital convertor)
- Analog Output with PWM (pulse width modulation)
- Control “things”
- Programmed before running

Introduction to the Arduino platform and related electronics

- Popularity of Arduino:
- Low cost
- Open-source hardware design
- Easy-to-use integrated development environment (IDE) to program it with
- Plug-in shields that add useful features to a basic Arduino board

Using Arduino as inexpensive dataloggers

- What kind of data the Arduino can collect?
- Digital timing:
- sub-millsecond timing resolution

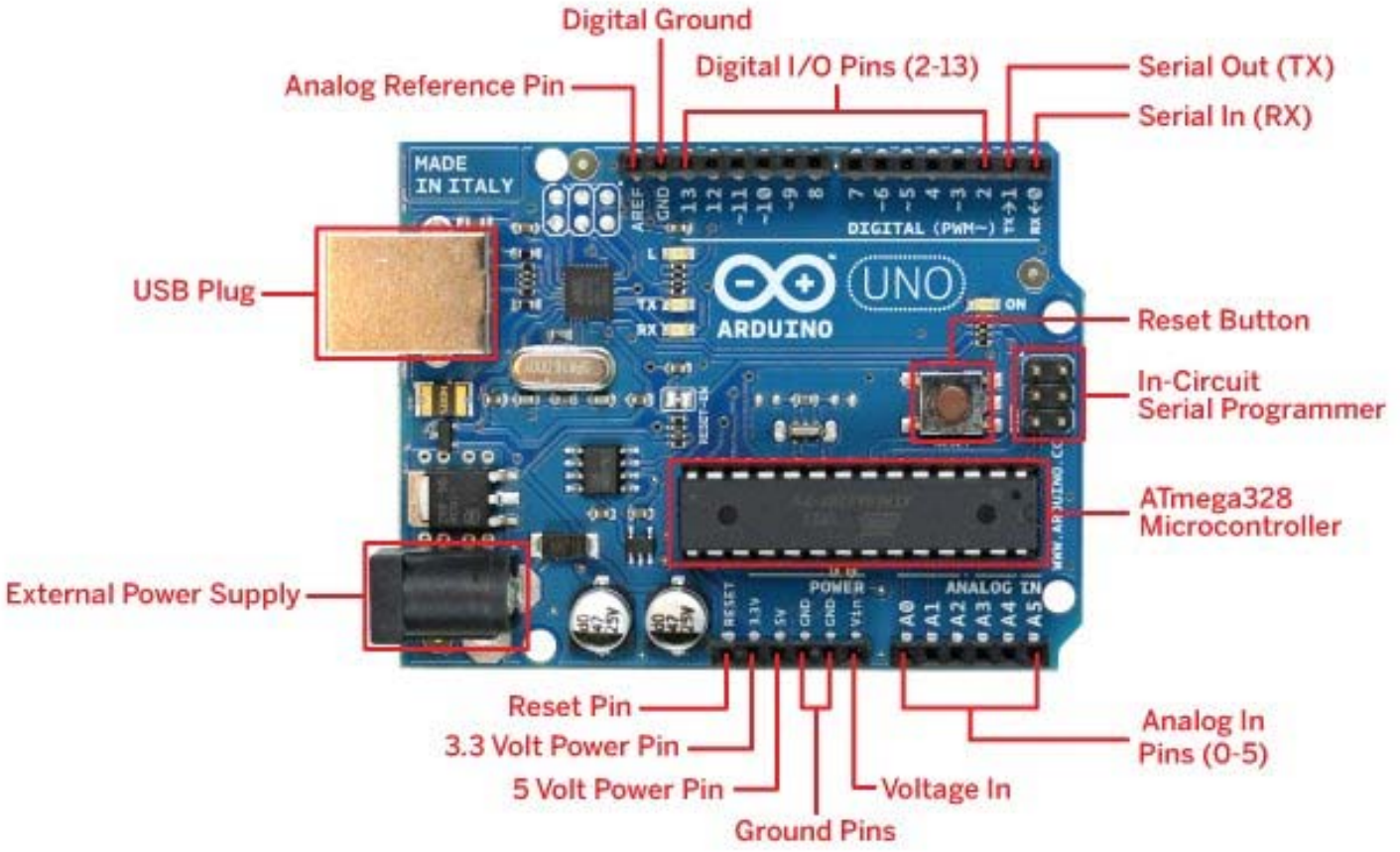
- Direct Analog Measurements:
- Built-in 10 bit ADCs on ATMega chip with limited resolution 1 part in 1024
- Sufficient measurements with MEMS accelerometer, light sensors, sound level meters, potentiometer-based angular measurements, analog Hall sensors, low-precision analog measurements

- Indirect Measurements:
- Built-in serial and I2C communications allow to communicate with other instruments (serial) and chips (I2C)
- Increase the measurement capabilities of device (e.g. atmospheric changes)

Arduino Uno R3

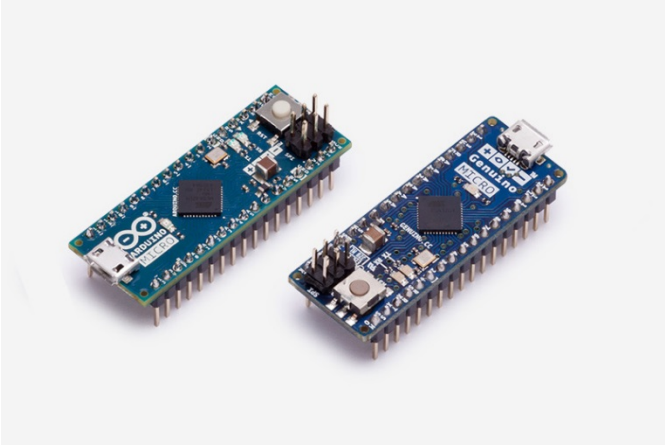
- Atmega 328 microprocesser
- 14 digital input/ output
- 6 analog inputs
- 16-MHz crystal oscillator
- USB connection

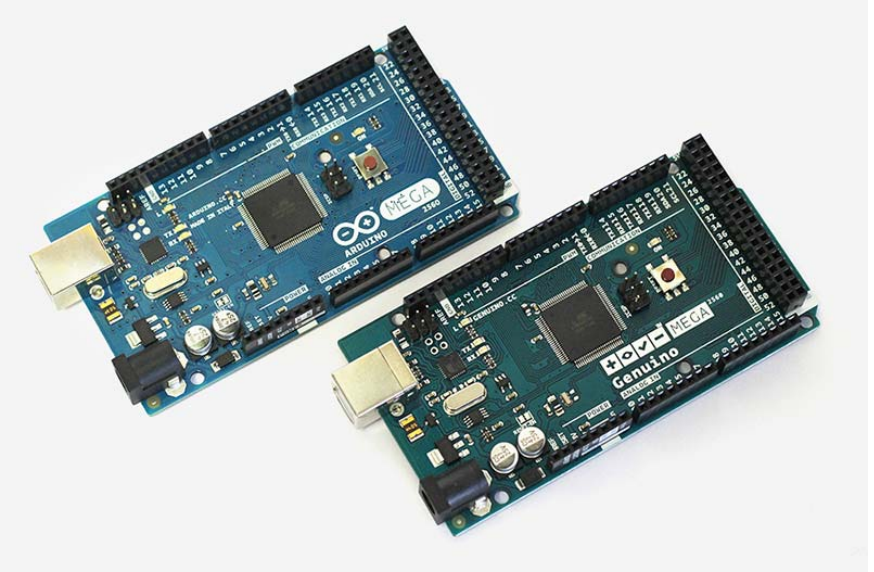
Arduino Uno R3



Different types of Arduino

- Mega : 54 digital I/O pins, 16 analog Inputs
- Micro : the smallest board
- MKR1000: WiFi connection





EXPERIMENTS AND REAL-WORLD APPLICATIONS WITH MOBILE APPLICATIONS

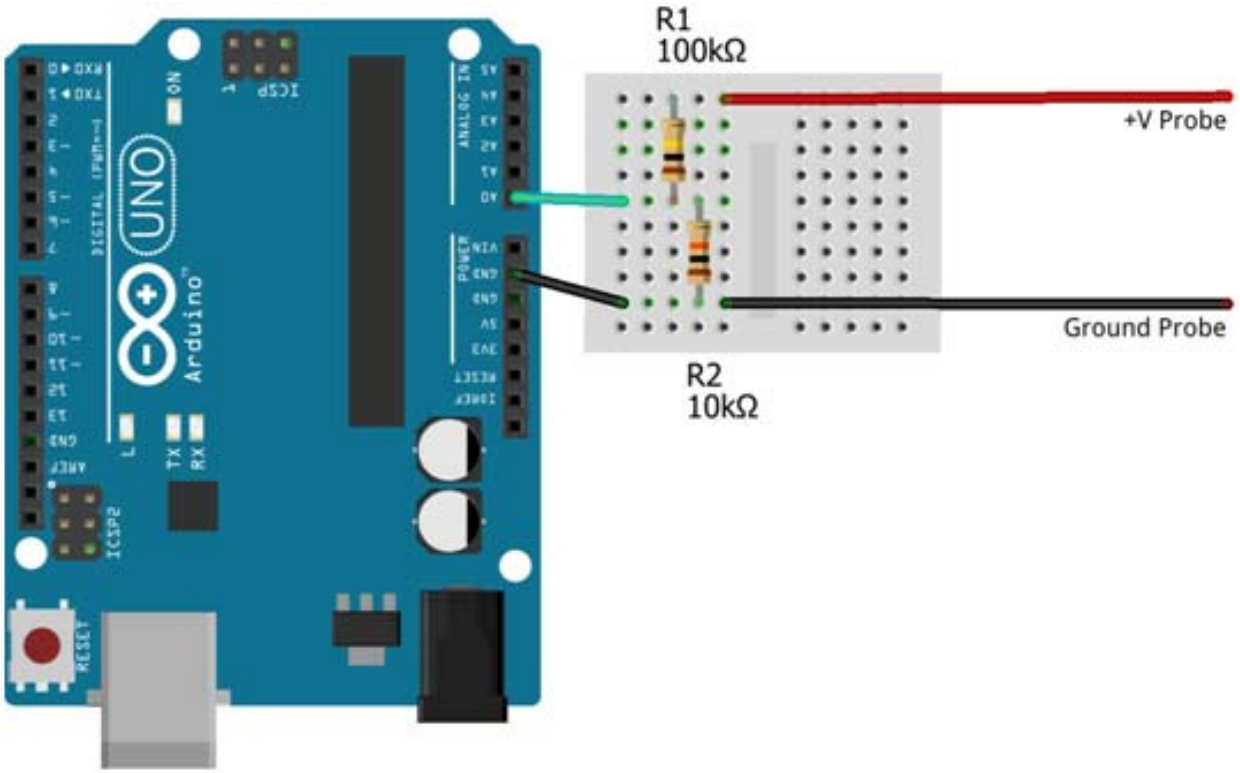
Experiments and Real-World Applications with Mobile Applications

- Arduino control unit as a blackbox
- Run the program to collect the experiment data only
- Writing the code is not taught in this course
- Through Bluetooth to communicate Arduino

Experiment 1: Make a digital voltmeter using an Arduino

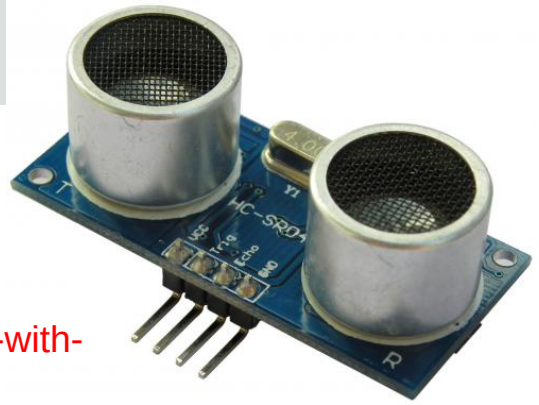
Objective	**(a) To measure the voltage up to 5V** **(b) To measure the voltage up to 30V**
Apparatus	(a) Direct measurement from analog input (b) 100kΩ resistor, 10 kΩ resistor or other combination of resistors to make potential divider LCD display (optional)
Arduino specification	Analog input pins that connect to an analog-to-digital converter (ADC) Arduino ADC is a ten-bit converter: output value will range from 0 to 1023

Experiment 1: Make a digital voltmeter using an Arduino



Experiment 2:Investigation of Hooke's law & S.H.M.

Objective	**(a) To determine the spring constant of a spring** **(b) To investigate the s-t graph of SHM motion**
Apparatus	HC-SR04 ultrasonic distance sensor
Sensor specification	Working Voltage: DC 5V Working Current: 15mA Working Frequency: 40Hz Max Range: 4m Min Range: 2cm
	It can be used to measure the speed of sound.

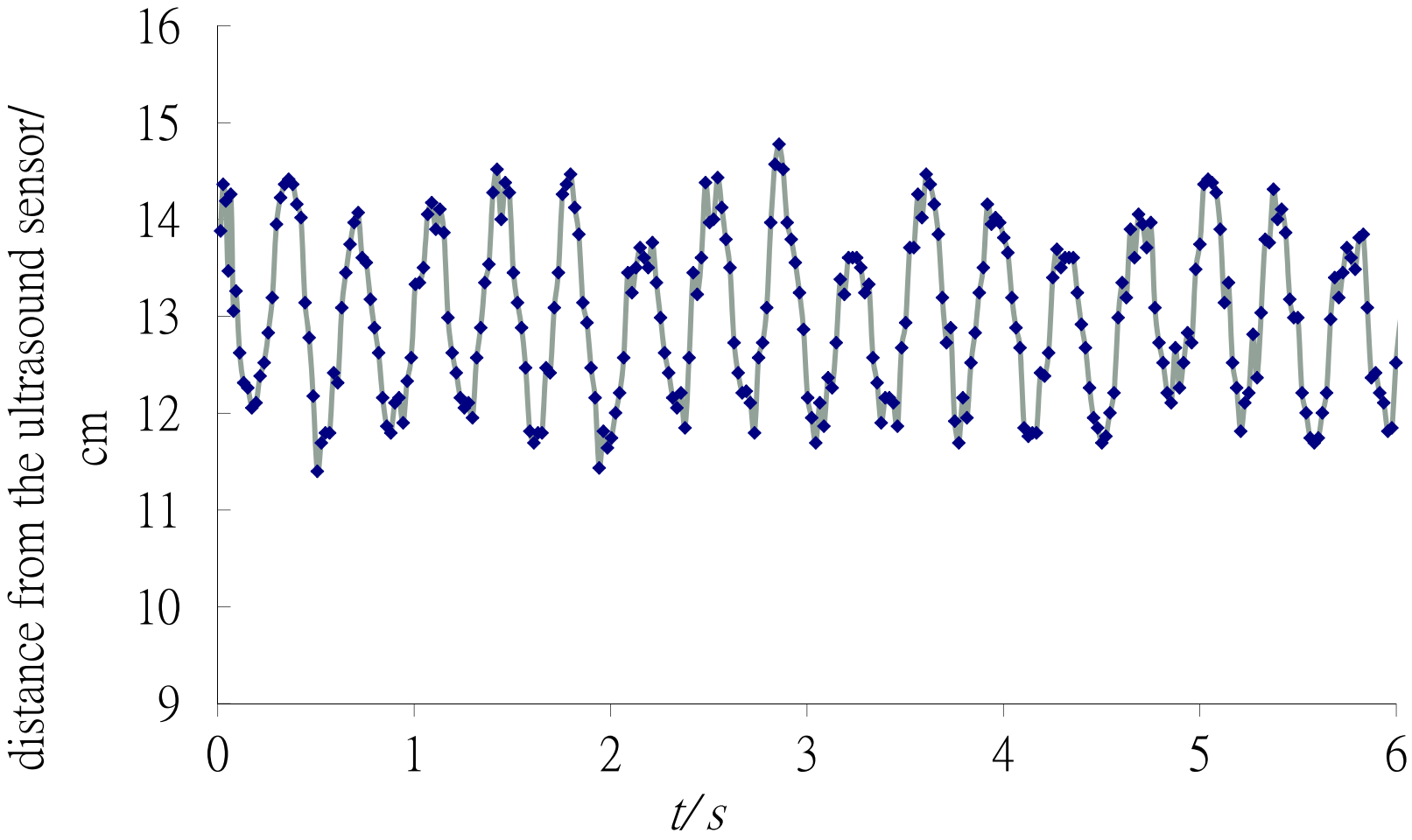


http://www.electroschematics.com/8902/hc-sr04-datasheet/

http://www.becker.edu/wp-content/uploads/2014/06/SHM.pdf

http://www.toptechboy.com/arduino/lesson-17-measuring-the-speed-of-sound-with-arduino-and-ultrasonic-sensor/

Experiment 2:Investigation of Hooke's law & S.H.M.



Experiment 2:Investigation of Hooke's law & S.H.M.

```
const int trigPin = 7;
const int echoPin =8;

void setup() {
  Serial.begin (9600);
}

void loop() {
  unsigned int echo_time;
  float distance;
  unsigned long time;

  pinMode(trigPin, OUTPUT);
  digitalWrite(trigPin, LOW);
  delayMicroseconds(10);
  digitalWrite(trigPin, HIGH);
  delayMicroseconds(10);
  digitalWrite(trigPin, LOW);

  pinMode(echoPin, INPUT);
  echo_time = pulseIn(echoPin, HIGH);
  time=micros();

  distance = echo_time/2.0*0.0343;
  distance = sqrt(distance*distance - 1.3*1.3);
  Serial.print(time/1000000.0, 6);
  Serial.print("\t");
  Serial.println(distance, 6);
  delay(10);
}
```

Experiment 2:Investigation of Hooke's law & S.H.M.

```
const int trigPin = 7;
const int echoPin =8;

void setup() {
  Serial.begin (9600);
}

void loop() {
  unsigned int echo_time;
  float distance;
  unsigned long time;

  pinMode(trigPin, OUTPUT);
  digitalWrite(trigPin, LOW);
  delayMicroseconds(10);
  digitalWrite(trigPin, HIGH);
  delayMicroseconds(10);
  digitalWrite(trigPin, LOW);
  pinMode(echoPin, INPUT);
  echo_time = pulseIn(echoPin, HIGH);
  time=micros();

  distance = echo_time/2.0*0.0343;
  distance = sqrt(distance*distance - 1.3*1.3);
  Serial.print(time/1000000.0, 6);
  Serial.print("\t");
  Serial.println(distance, 6);
  delay(10);
}
```

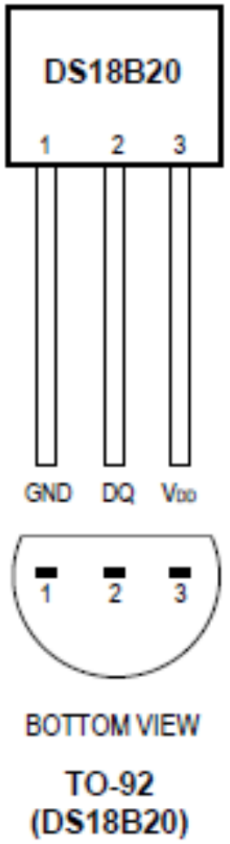
Experiment 3: Radiation absorption

Objective	Compare the absorption of radiation on different colour surfaces
Apparatus	Dallas DS1820/ DS18B20 thermometer Shiny can Black painted can Light bulb

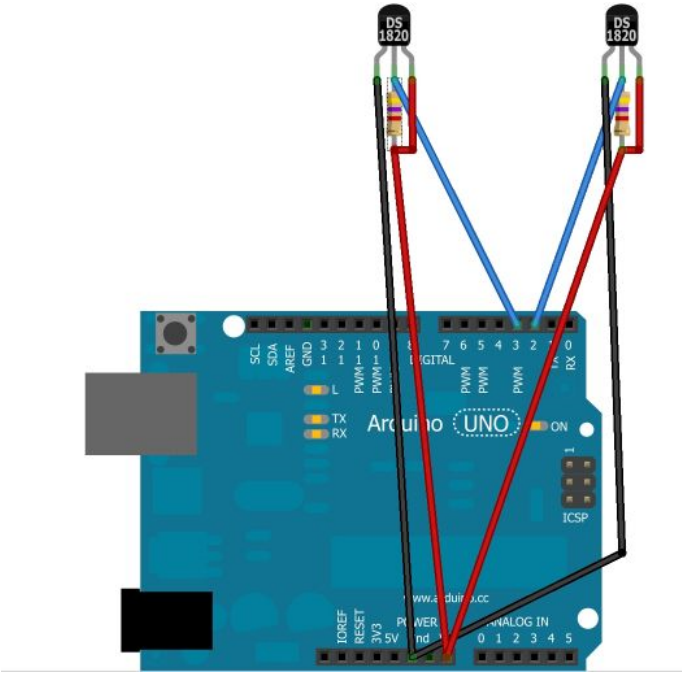
http://www.fisicayarduino.com.ar/

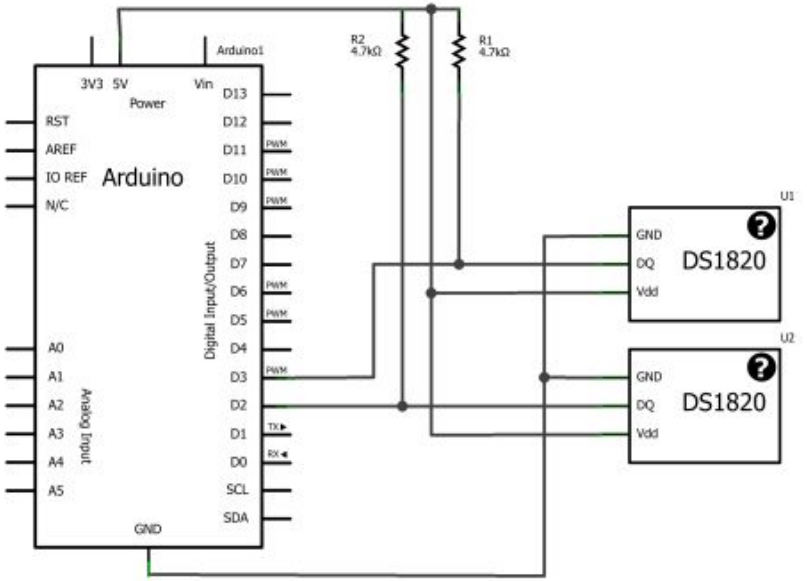
Experiment 3: Radiation absorption

- DS18B20 digital thermometer
- Only one port pin for communication
- Measures temperature from -55 ^{0}C to +125 ^{0}C
- A unique 64-bit serial code stored on-board ROM

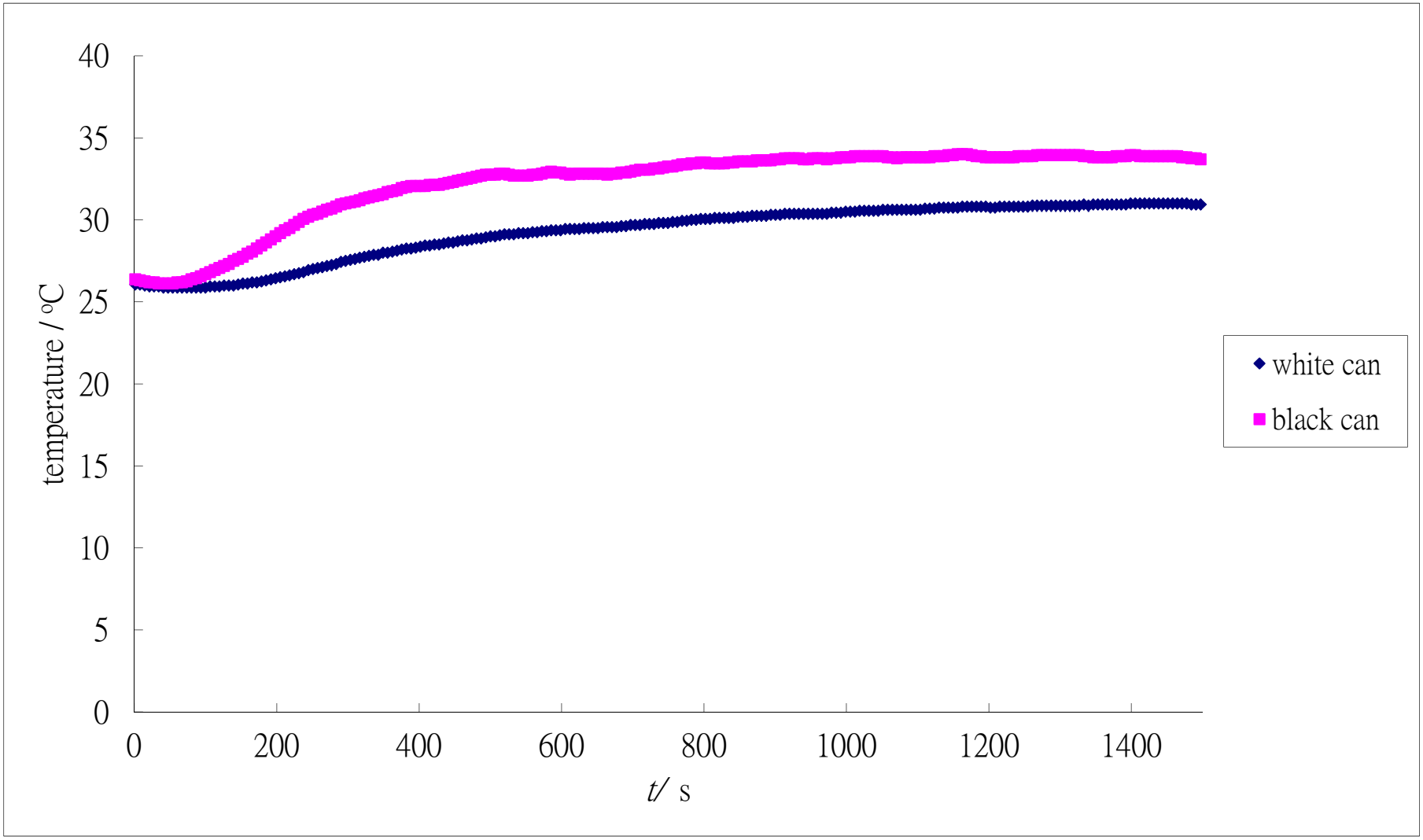


Experiment 3: Radiation absorption





Experiment 3: Radiation absorption



Experiment 3: Radiation absorption

```
#include <OneWire.h>
#include <DallasTemperature.h>

#define TWHITE 3
#define TBLACK 2

int start;

OneWire busWhite(TWHITE);
OneWire busBlack(TBLACK);

DallasTemperature sensorWhite(&busWhite);
DallasTemperature sensorBlack(&busBlack);

void setup(void)
{

  Serial.begin(9600);

  sensorWhite.begin();
  sensorBlack.begin();

  start = millis();
}
```

```
void loop(void)
{
  sensorWhite.requestTemperatures();
  sensorBlack.requestTemperatures();

  Serial.print(millis()-start);
  Serial.print(";");
  Serial.print(sensorWhite.getTempCByIndex(0));
  Serial.print(";");
  Serial.println(sensorBlack.getTempCByIndex(0));

  delay(5000);
}
```

2.4 App Inventor and Arduino

- We can receive data from Arduino Devices.
- We have prepared a number of Arduino devices, you may use the BT_Client App to connects to them and obtain data!
- Further programming is required to turn them into useful Dataloggers

2.5 Trying out with BT and Arduino

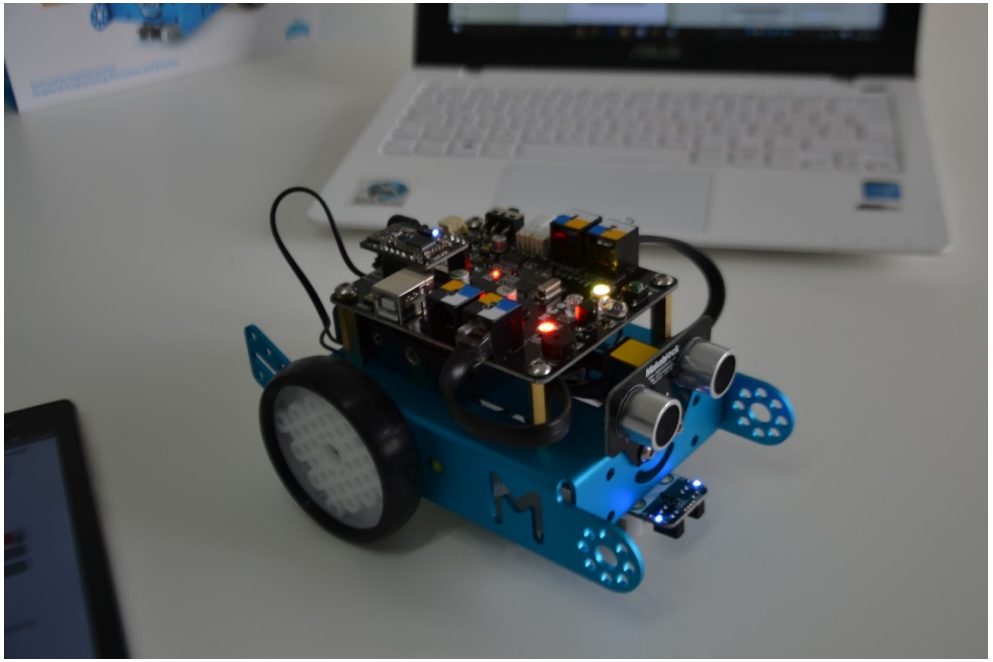
- Please pair your mobile phone / tablets with the Bluetooth Devices you can search
- All PASSKEY are 0000
- Walk around and try different setup

2.6 Other Possibilities

- Besides Arduino, other devices can also be connected with Bluetooth
  - LEGO NXT

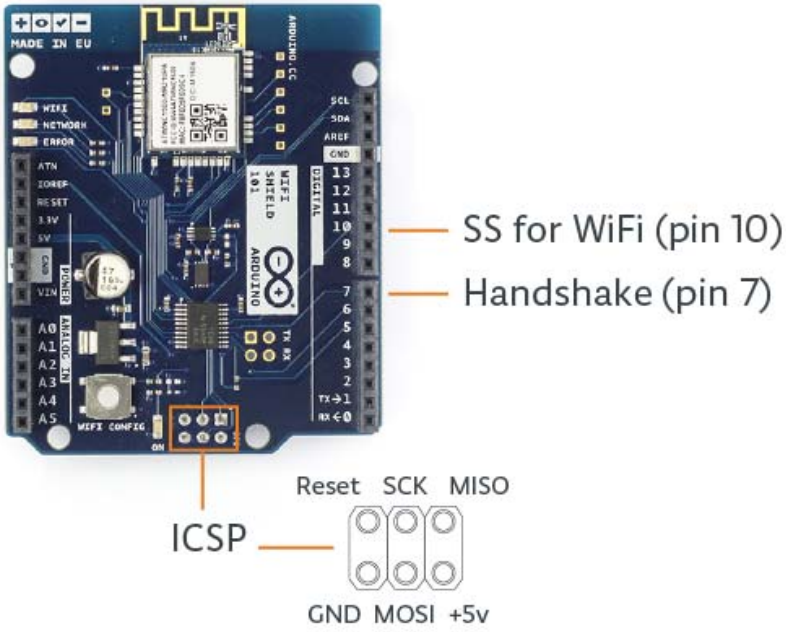
2.6 Other Possibilities

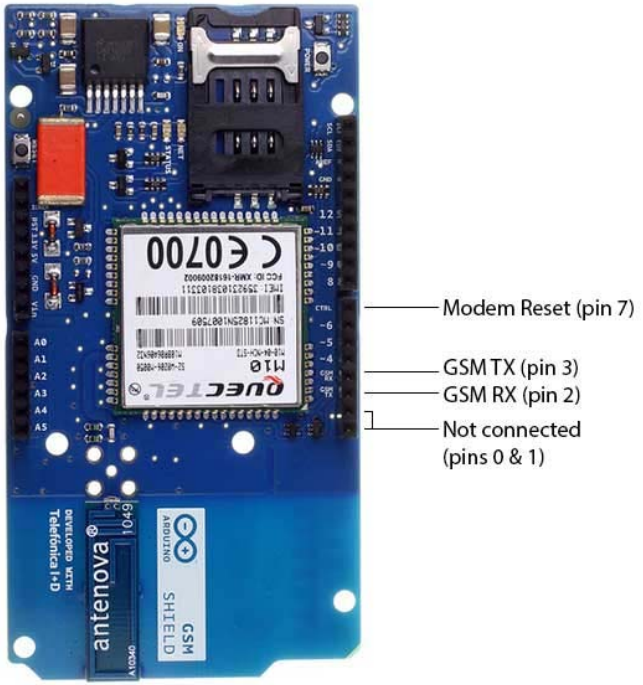
- App inventor + mBot
- http://webtoolsreview.blogspot.hk/2016/04/programming-mbot-with-app-inventor-2.html



2.6 Other Possibilities

- Arduino + WiFi/ GSM:
- Arduino WiFi Shield 101/ Arduino GSM Shield
- https://www.arduino.cc/en/Guide/ArduinoWiFiShield101
- https://www.arduino.cc/en/Guide/ArduinoGSMShield

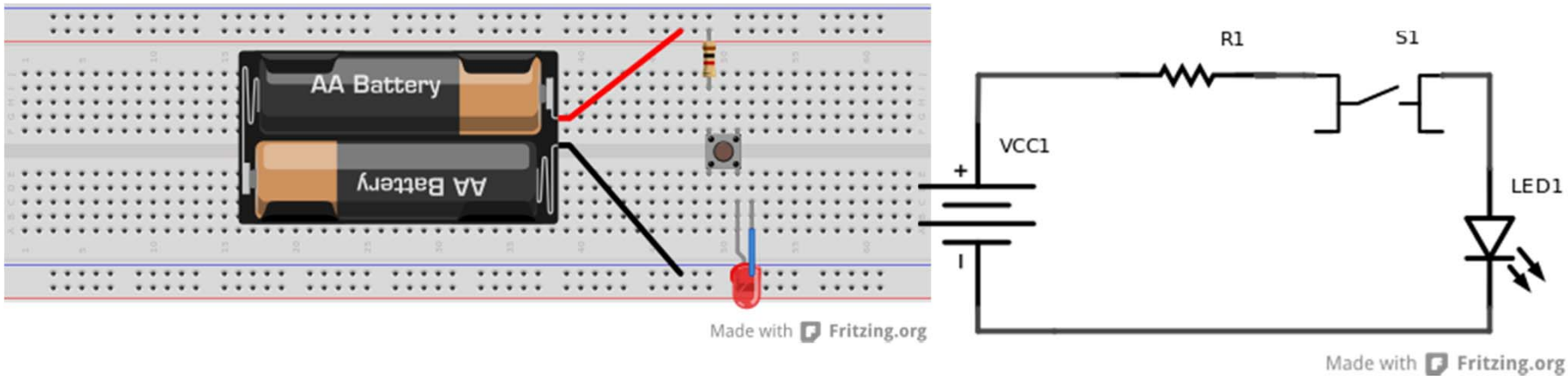




2.6 Other Possibilities

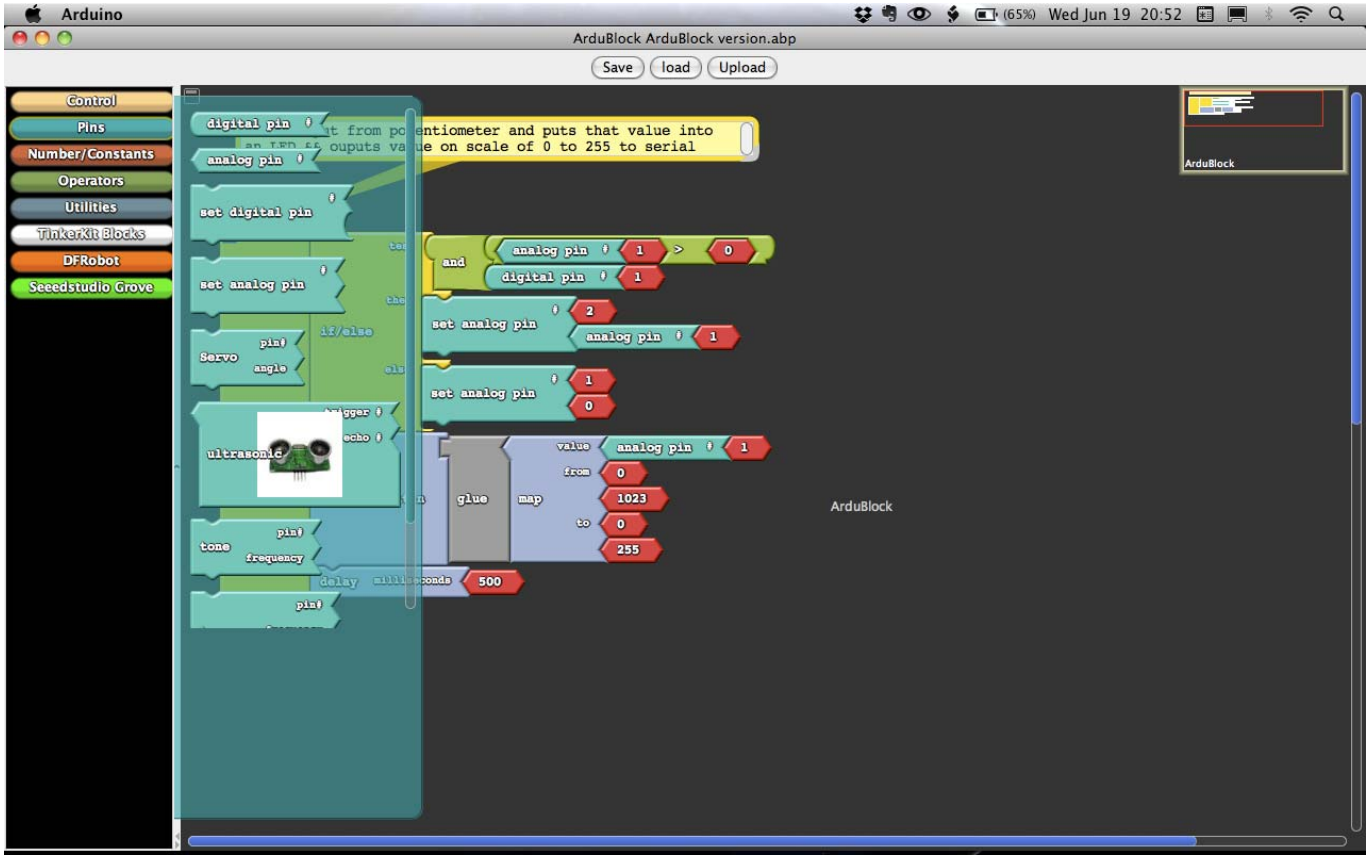
Fritzing:

- An open source for the design of electronics hardware.
- To support designers ready to move from experimenting with a prototype to building a more permanent circuit.
- http://fritzing.org/home/



2.6 Other Possibilities

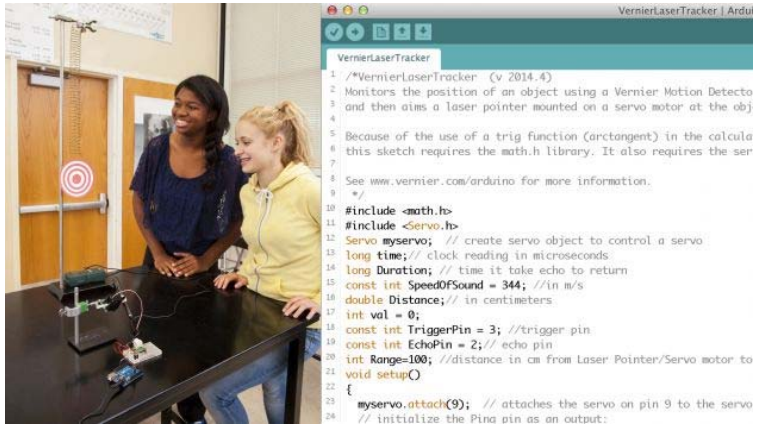
- A Block Language for Arduino:
- http://blog.ardublock.com/



2.6 Other Possibilities

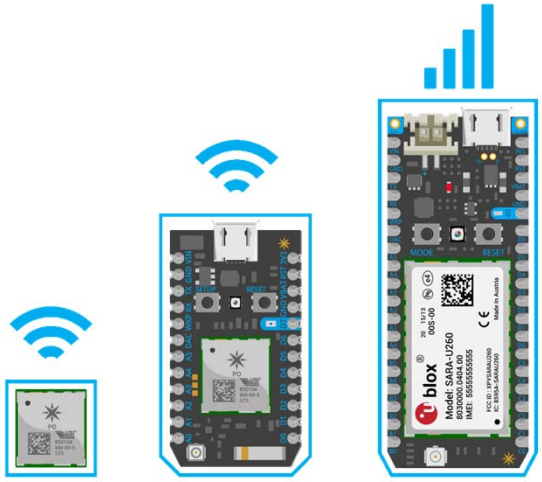
- Physics lab Sensors + Arduino
- Vernier Arduino Interface Shield
- http://www.vernier.com/engineering/arduino/

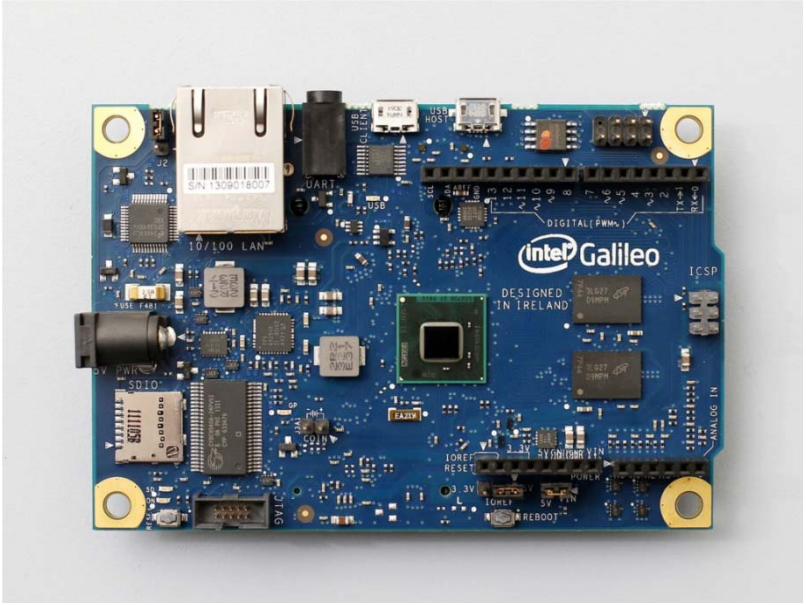
Vernier
Arduino Interface Shield



2.6 Other Possibilities

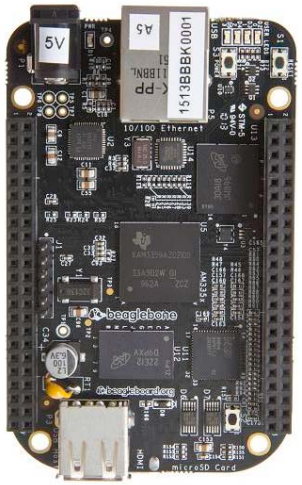
- Other microcontrollers/ Computers:
  - Intel + Arduino: Intel Galileo, Intel Edison
  - https://www.arduino.cc/en/ArduinoCertified/IntelGalileo
  - Photon(WiFi), Electron(2G/3G):
  - https://www.particle.io/
  - Grove, LinkitOne, etc.:
  - http://www.seeedstudio.com/





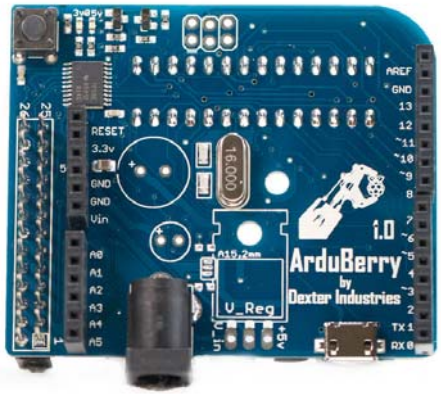
2.6 Other Possibilities

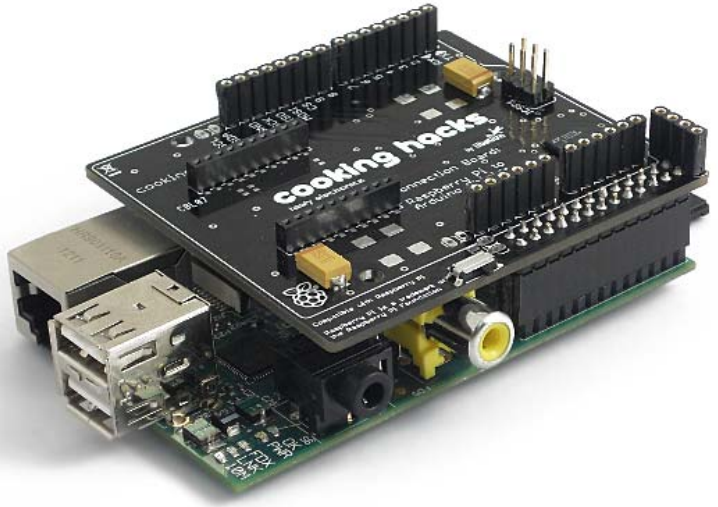
- Card-sized single board computer:
- Raspberry Pi
- https://www.raspberrypi.org/
- BeagleBones
- http://beagleboard.org/bone



2.6 Other Possibilities

Raspberry Pi+ Arduino:
Arduberry
http://www.dexterindustries.com/Arduberry/
Raspberry Pi to Arduino Shields Connection Bridge
https://www.cooking-hacks.com/documentation/tutorials/raspberry-pi-to-arduino-shields-connection-bridge/





Conclusion

- App Inventor may not be a very precise and accurate experimental tool
- But it is handy, accessible and available
- It is covered in KS3 ICT in some schools
- It can be used to further connects to other devices such as Arduino which is a very very interesting platform for possible investigative study and STEM projects

Reference

- http://arduino.cc
- http://freeduino.org
- http://phys.csuchico.edu/~eayars
- http://www.rugged-circuits.com/10-ways-to-destroy-an-arduino/
- https://www.adafruit.com/
- https://www.sparkfun.com/
- http://www.seeedstudio.com/