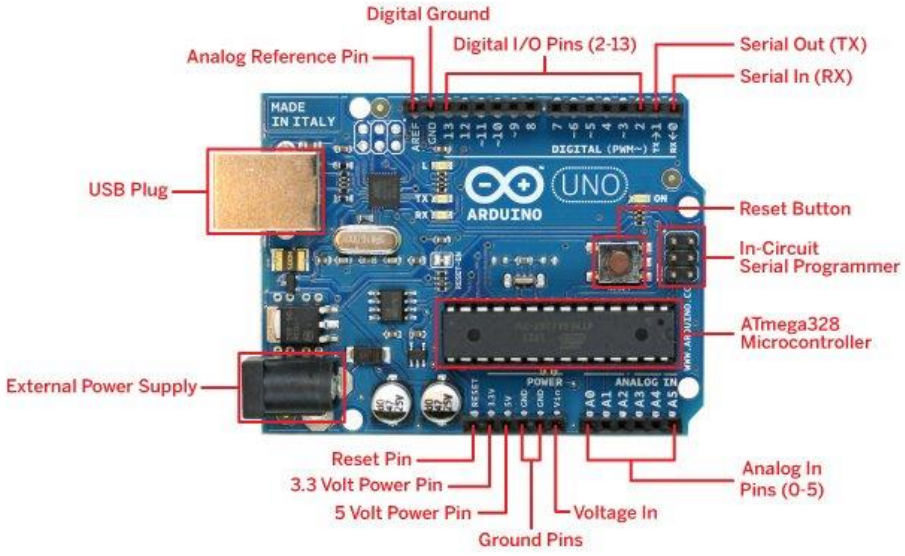
Introduction to Arduino

The laboratorial component of the CU Internet of things consists on a set of exercises and challenges to complete using Arduino (https://www.arduino.cc/). On the next Figure you can see the different pins available on the device.



Links for Arduino:

- https://www.allaboutcircuits.com/tools/resistor-color-code-calculator/
- http://www.audioacustica.com.br/exemplos/Valores_Resistores/Calculadora_Ohms_Resistor.html
- https://www.arduino.cc/en/Tutorial/HomePage?from=Main.Tutorials
- https://thestempedia.com/tutorial-hub/arduino/
- https://thestempedia.com/tutorials/arduino-ide/

TinkerCad Simulator

The exercises must be solved using TinkerCad (https://www.tinkercad.com) simulator. For that, you need to create a free account.

The exercises can be solved based on circuits built previously. For that, on the circuit options, chose “Duplicate” and change the name of the duplicated circuit.

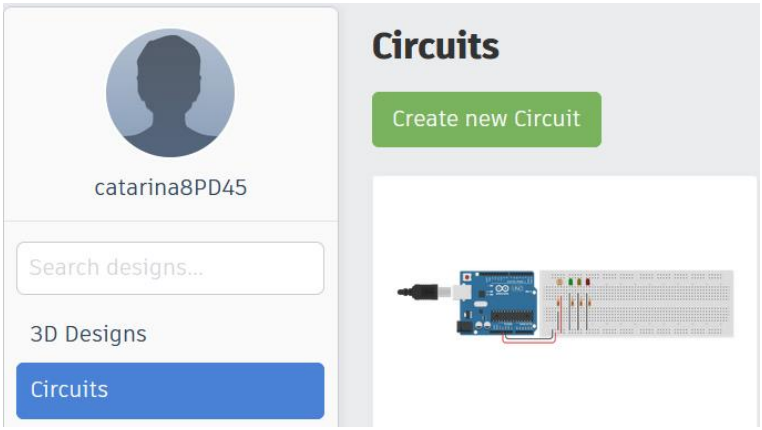
Arduino IDE

The IDE can be obtained in https://www.arduino.cc/download_handler.php and installed. The configuration will be performed on the first use.

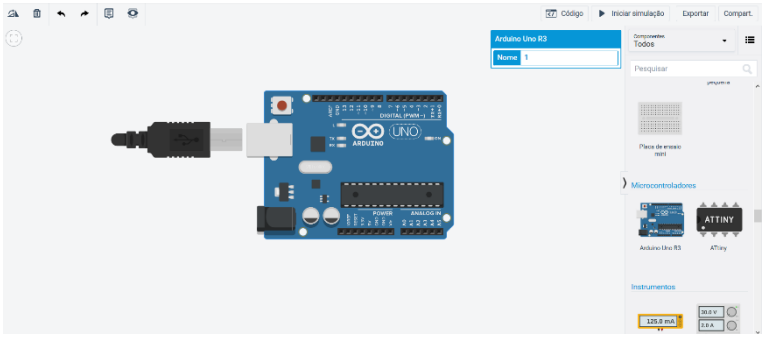
Exercises –TinkerCad simulator

1. Blinking LEDs

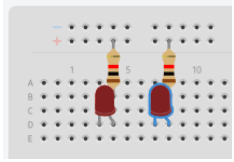
1. Circuits > Create new circuit



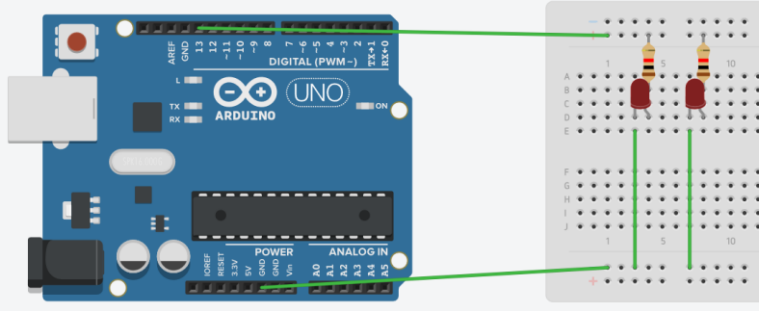
2. In “Components” select “All”
3. Select “Arduino UNO R3” and drag it to the work pane



4. Drag to the working pane the following components:
   - 1 breadboard
   - 2 resistors
   - 2 LEDs
5. Drag the resistors to the breadboard so that one of each resistor’s end is connected to the positive (+) row
6. LEDs have a longer “leg” (the “crooked” one) that should be connected to the positive pole. Set the LEDs so that the longer leg is connected to the other end of the resistors.



7. Create connections as shown on the next figure



8. Open “Code”
9. Instead of “Blocks”, select “Text”
10. Click “Start simulation”. **What happens?**
11. Change the delay values to 100. **What happens?**
12. Change the last delay back to 1000. **What happens?**
13. Save the cicuit: click on the TinkerCad logo (top left corner). On the circuit options (gear), select “Properties”, change the project name to “First test” and save the changes.

2. Blinking LEDs 2

Do you know Queen’s “We Will Rock You” (https://youtu.be/-tJYN-eG1zk)? Change the code so that the LEDs blink according to its intro.

3. Blinking LEDs 3

Change the circuit and the code so that, instead of blinking simultaneously, the LEDs will blink alternately for one second each.

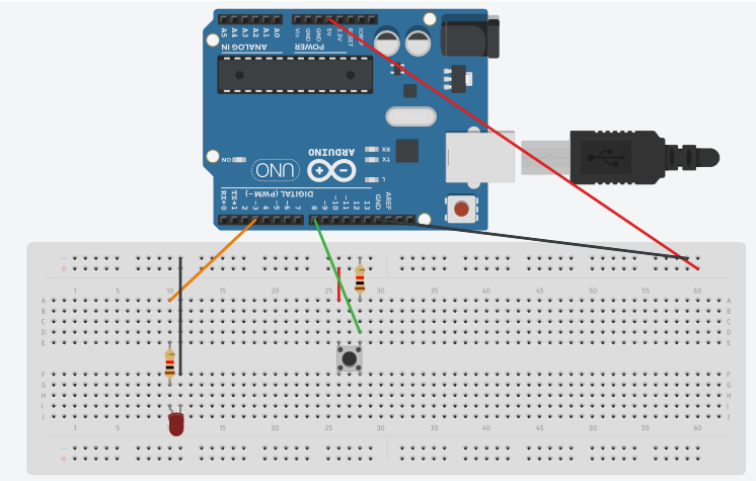
4. Blinking LEDs 4

Starting with the result of exercise 2, change the code so that the LEDs will blink following “We Will Rock You” intro rhythm, and one of the LEDs is the “tum” and the other one is the “ta”.

5. Use button to turn LED ON and OFF

1. Create a new circuit
2. Drag the components:
   - 1 breadboard
   - 1 Arduino UNO R3
   - 1 LED
   - 2 Resistors
   - 1 Button

3. Setup the circuit:



4. Insert the following C code:

```
/* LED controlado por botão */

// set pinos
const int buttonPin = 8; // número do pino a que está ligado o botão
const int ledPin = 3; // número do pino a que está ligado o LED

// set variables
int estado = 0; // variável para leitura do botão
int guarda_estado = LOW; // variável para armazenar valores do botão

void setup() {
    pinMode(ledPin, OUTPUT); // define o pino do Led como saída do Arduino
    pinMode(buttonPin, INPUT); // define pino do pushbutton como entrada do Arduino
}

void loop(){
    estado = digitalRead(buttonPin); // le o estado botão: ligado (HIGH) ou desligado (LOW)
    if (estado == HIGH) { // verifica se o botão está pressionado
        guarda_estado = !guarda_estado; // inverte valor do estado
        delay(500); //esperera o tempo de 500ms para evitar que haja várias vezes alterações
    }

    if (guarda_estado == HIGH) {
        digitalWrite(ledPin, HIGH); // liga o led
    } else {
        digitalWrite(ledPin, LOW); // desliga o led
    }
}
```

Comments translation (by line):
1: LED controlled by button
3: set pins
4: number of the pin where the button is connected
5: number of the pin where the LED is connected
8: variable to read the button (estado = state)
9: variable to store button's state
12: defines the LED pin as output
13: defines the button pin as input
17: read the button state: on (HIGH) or off (LOW)
18: check if the button is pressed
19: switches state
20: waits 500 milliseconds to avoid several changes
24: turns the LED on
26: turns the LED off

5. Start and test the simulation

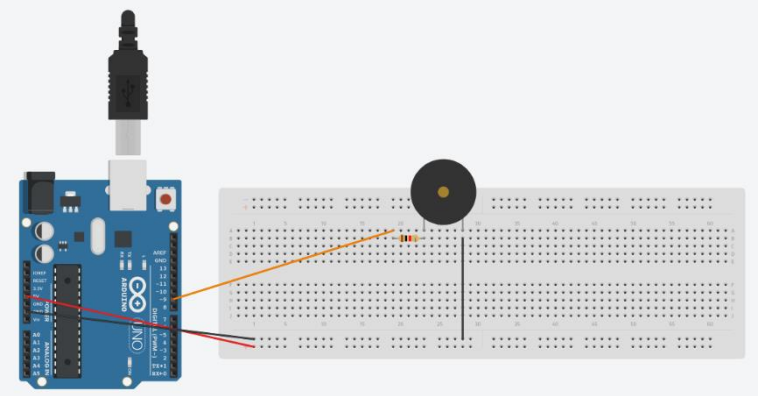
6. Using Buzzer

Piezo speakers (buzzers) A "piezo buzzer" is basically a tiny speaker that you can connect directly to an Arduino. ... Connect one pin (it doesn't matter which one) to the Arduino's ground (Gnd) and the other end to digital pin 8. From the Arduino, you can make sounds with a buzzer by using tone.

Documentation: https://www.arduino.cc/reference/en/language/functions/advanced-io/tone/

Buzzer emits sound intermittently

1. Circuit:



2. Code:

Texto | 1 (Arduino Uno R3)

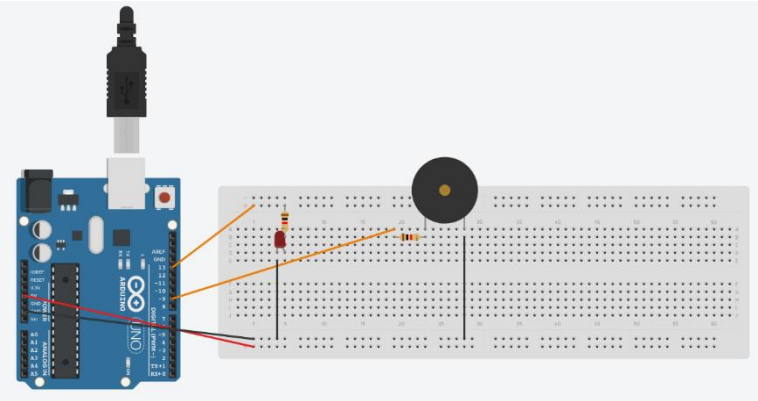
```
const int buzzer = 9;

void setup(){
  pinMode(buzzer, OUTPUT);
}

void loop(){
  tone(buzzer, 1000); // Send 1KHz sound signal...
  delay(1000);        // ...for 1 sec
  noTone(buzzer);     // Stop sound...
  delay(1000);        // ...for 1sec
}
```

3. Start and test the simulation

7. Using Buzzer and LED

Buzzer emits sound intermittently, while LED blinks

1. Circuit:



2. Code:

Texto | 1 (Arduino Uno R3)

```
1  const int buzzer = 9;
2  //completar: definir o pino do led
3
4
5  void setup(){
6    pinMode(buzzer, OUTPUT);
7    // completar: definir o pino do led como sendo de output
8  }
9
10 void loop(){
11
12   tone(buzzer, 1000);
13   // completar: o que tem de acontecer aqui?
14   delay(1000);
15   noTone(buzzer);
16   // completar: o que tem de acontecer aqui?
17   delay(1000);
18
19 }
```

Comments translation (by line):
2: complete: define the LED pin
7: complete: define the LED pin as output
13: complete: what needs to happen here?
16: complete: what needs to happen here?

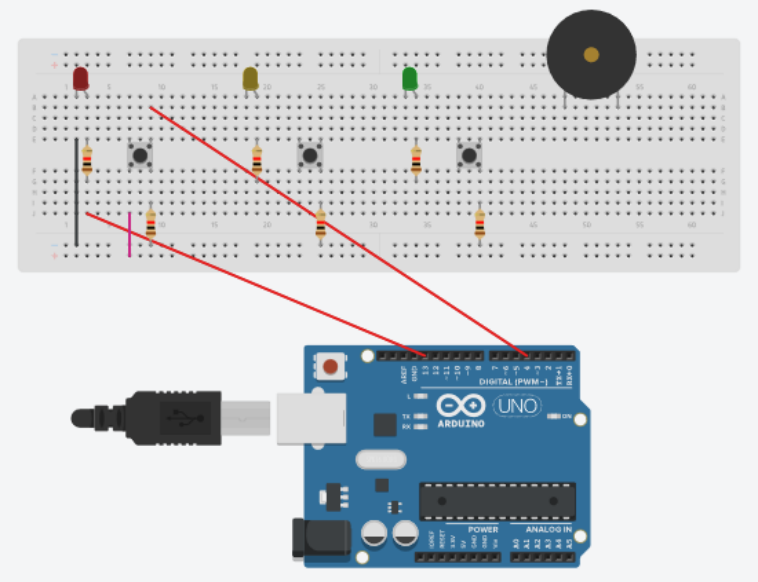
3. Complete the spaces marked with "// completar" with the code needed
4. Start and test the simulation

8. Blinking piano

Objective: Create a circuit where each button turns on an LED and plays a different musical note.

1. Components:
   - 1 Breadboard
   - 1 Arduino UNO R3
   - 3 LEDs
   - 6 Resistors
   - 3 Buttons
   - 1 Buzzer

2. Partial circuit (only with connections for the first LED and button)



3. Complete the circuit
4. Partial code (only for the first LED and button)

Texto — 1 (Arduino Uno R3)

```
const int ledpino1 = 13;
const int botaoA = 4;
int estado_botaoA = 0;
// completar: leds 2 e 3, botões B e C
const int som = 9;
int tom = 0;

void setup() {
    pinMode(ledpino1, OUTPUT);
    pinMode(botaoA, INPUT);
    // completar: leds 2 e 3, botões B e C
    pinMode(som, OUTPUT);
}

void loop() {
    estado_botaoA = digitalRead(botaoA);
    // completar: botões B e C

    if(estado_botaoA && !estado_botaoB && !estado_botaoC) {
        tom = 100;
        digitalWrite(ledpino1, HIGH);
    }
    // completar: botões B e C (mudar tom), leds 2 e 3

    if(tom > 0) {
        digitalWrite(som, HIGH); // Liga buzzer
        delayMicroseconds(tom); // Espera o tempo proporcional ao comprimento de onda da nota musical em milisegundos
        digitalWrite(som, LOW); // Desliga buzzer
        delayMicroseconds(tom); // Espera o tempo proporcional ao comprimento de onda da nota musical em milisegundos
        tom = 0; // Reset do tom para zero, para sair do loop while e nao tocar o som constantemente
        digitalWrite(ledpino1, LOW);
        // completar: leds 2 e 3
    }
}
```

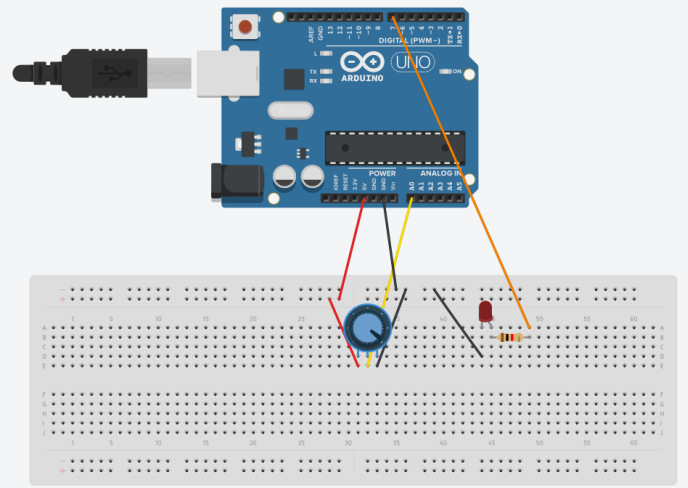
Comments translation (by line):
2: complete: define the LED pin
7: complete: define the LED pin as output
13: complete: what needs to happen here?
16: complete: what needs to happen here?

5. Complete the spaces marked with "// completar" with the code needed
6. Start and test the simulation

9. Analog Input + LED

Objective: use an analog input to set the LED

1. Circuit



2. Code

```
Texto                                                    1 (Arduino Uno R3)

1  int led = 7;
2
3  void setup() {
4    Serial.begin(9600);
5    pinMode(led, OUTPUT);
6  }
7
8  void loop() {
9    int sensorValue = analogRead(A0);
10   int newValue = map(sensorValue, 0, 1023, 0, 255);
11   analogWrite(led, newValue);
12
13   // print the results to the Serial Monitor:
14   Serial.print("sensor = ");
15   Serial.print(sensorValue);
16   Serial.print("\t output = ");
17   Serial.println(newValue);
18
19   delay(1000);
20 }
```

3. Start and test the simulation

10. Analog Input + LED + Buzzer

Use the analog input to control the LED's intensity and the frequency in which the buzzer beeps.

11.Temperature Sensor and LCD 16x2

Use a temperature sensor to measure the present temperature and show the result on an LCD 16x2

Documentation:

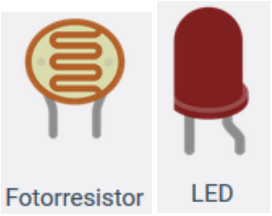
- Temperature sensor:
  - https://www.instructables.com/id/TMP36-Temperature-Sensor-Arduino-Tinkercad/
  - https://www.bc-robotics.com/tutorials/using-a-tmp36-temperature-sensor-with-arduino/
    - Note: the sensor's output isn't the temperature (in ºC). To convert it, the expression on "TMP36 Temperature Sensor With Arduino in Tinkercad" can be used (supposing that the sensor is connected to analog output A0):

```
celsius = map(((analogRead(A0) - 20) * 3.04), 0, 1023, -40, 125);
```

      - Map function: https://www.arduino.cc/reference/en/language/functions/math/map/
- LCD 16x2: https://www.tinkercad.com/things/dGiMrTYFn9R-copy-of-16x2-lcd-display/editel?sharecode=mnqd_76wiTlxLtJsoFUPIPDmBtywokO1j6thV38e8KA

12.Light sensor and LEDs

Use a light sensor (photoresistor) to measure the present luminosity and, according to the value, turn on a different LED: green (low luminosity: < 300), yellow (average: <600), or red (high).

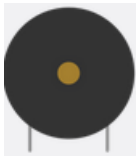


Documentation:

- Photoresistor: https://blog.tinkercad.com/2018/08/15/light-sensor-photoresistor-with-arduino-in-tinkercad-circuits
- LED: https://www.instructables.com/id/Blink-an-LED-With-Arduino-in-Tinkercad/

13.Gas sensor and Buzzer

Use a gas sensor to measure the gas concentration and, according to the value, beep a different sound (for example, half the value obtained by the sensor)

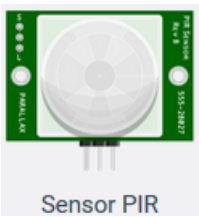


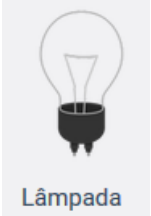
Documentation:

- Gas sensor
    - o https://www.tinkercad.com/things/7S1QneIskY8-copy-of-gas-sensor/editel?sharecode=MjEuHGF7JDcU0CepZEmqa2t5FWN5YaavLPZCvmtCgqw
- Buzzer
    - https://www.tinkercad.com/things/d4URlmkqGYu-piezo
    - Função tone: https://www.arduino.cc/reference/en/language/functions/advanced-io/tone/

14.Movement sensor and lamp

Use a PIR sensor to detect movement and, when detected, a lamp should be turned on.



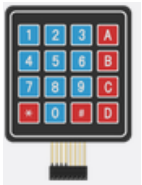


Documentation:

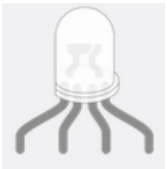
- PIR sensor: https://www.tinkercad.com/things/kXM4u4GVFDd-copy-of-if-else-pir-sensor-leds/editel?sharecode=ItjLpEMR6_YSbZnT8LwZs5viorl9SBRtG-Jcon6o454
- Lamp: https://www.tinkercad.com/things/3W0m9blpRrd-copy-of-turn-on-an-ac-lamp-using-arduino-5v-relay-and-motion/editel?sharecode=IV9HRpSHwx95U_9E2u7ogN-bufKNT_IceXOdAphD81o

15.4x4 keypad and RGB LED

Use a 4x4 keypad to change an RGB LED color, according to the pressed key.



Teclado 4x4


LED RGB

The colours must be set on the RGB LED according to the following table:

Key	Red	Green	Blue
1	0	0	0
2	255	255	255
3	255	0	0
A	0	255	0
4	0	0	255
5	255	255	0
6	0	255	255
B	255	0	255

Key	Red	Green	Blue
7	192	192	192
8	128	128	128
9	128	0	0
C	128	128	0
*****	0	128	0
0	128	0	128
#	0	128	128
D	0	0	128

Documentation:

- 4x4 keypad: https://www.tinkercad.com/things/8iDg0ZuguN8-using-keypad-4x4-with-arduino
- RGB LED: https://www.tinkercad.com/things/a8KMGOddc8F-copy-of-rgb-led/editel?sharecode=D5X48FVf3lJGAr1Xjd6Dcp5mNcfSz2tjnAKeHPJjEE4

16.4x4 keypad

Use a 4x4 keypad to insert a code that should be shown on the serial monitor. The codes may contain characters A, B, C and D and numbers 0 – 9. There is no minimum or maximum number of characters. The codes start with * and end with #. The code inserted should be shown on the serial monitor.

17.4x4 keypad and LCD

Use a 4x4 keypad to insert a code that should be shown on the LCD. The codes may contain characters A, B, C and D and numbers 0 – 9. There is no minimum or maximum number of characters. The codes start with * and end with #. The code inserted should be shown on the LCD.

18.Infrared remote

Use infrared remote and sensor. The key pressed on the remote should be shown on the serial monitor.

19.Infrared remote and LEDs

Use infrared remote and sensor. Depending on the key pressed, a different LED should be turned on:

- 1: green
- 2: blue
- 3: yellow
- Any other: red

Besides, when the red LED is turned on, the serial monitor must show the cause of the error (example: “Error: pressed 4 key”)

Exercises – Arduino board

1. Blinking LEDs

1. Setup the circuit from exercise **1** of the previous section using the hardware available on the lab
2. Connect the circuit to your computer
3. Open the IDE and check if Arduino has been recognised:
   - Tools > Board “Arduino/Genuino Uno”
   - Port > Serial Port - COM3 or COM1
4. Select the Blink example: File > Examples > 01.Basics > Blink
5. On the new opened window, clock “Verify” to see that there are no errors
6. Click “Send” to see the circuit working

2. Turn LED on and off with button

Repeat exercise **5** of the previous section using the hardware available on the lab and the Arduino IDE on your computer.

3. Analog Input

Repeat exercise **9** of the previous section using the hardware available on the lab and the Arduino IDE on your computer.

Besides these, you can reproduce any of the exercises from the previous section using the hardware available on the lab and Arduino IDE on your computer.