Arduino

Yarelis D. Acevedo
Arianna H. Colón
Tiahra N. Avilés
AGENDA

1. What is Arduino?
   What is an Arduino? Why Arduino? How can I use it and implement it on the lab?

2. Arduino Software
   How Arduinos are programmed

3. How to get started?

4. Hands On
   Using Tinkercad simulator
What is a microcontroller

To answer this enter to menti.com
Code: 5829 2707
What is a microcontroller?

A microcontroller (MCU for microcontroller unit) is a small computer on a single metal-oxide-semiconductor (MOS) integrated circuit (IC) chip. A microcontroller contains one or more CPUs (processor cores) along with memory and programmable input/output peripherals.
Big computers vs small computers

Dell Precision T1500

- **CPU Speed**: 2.93GHz
  - Quad-core!
- **RAM**: 16GB
- **Storage**: 2TB
- **I/O**:
  - USB, Firewire, Serial, PS/2, RJ-45, Audio, etc.

Atmel ATMEGA328P

- **CPU Speed**: 20MHz
  - Not Quad-core 😊
- **RAM**: 2KB
- **Storage**:
  - 32KB Program Memory
  - 1KB EEPROM
- **I/O**:
  - Up to 23 generic I/O
    - 6 of them ‘analog-capable’
    - UART/SPI/I²C, etc.
What is Arduino?

What is an Arduino? What is the purpose? How can I use it and implement it on the lab?
A brief story. The Arduino project began in 2005 as a tool for students at the Interaction Design Institute Ivrea in Ivrea, Italy, aiming to provide a low-cost and easy way for novices and professionals to create devices that interact with their environment using sensors and actuators.
Arduino Microcontroller

- Open-source electronics platform based on easy-to-use hardware and software.
- Are able to read inputs - light on a sensor, a finger on a button, or a Twitter message - and turn it into an output - activating a motor, turning on an LED, publishing something online.
What is it used for?

- Physical Computing projects / research
- Interactive Installations
- Rapid Prototyping
What can I do?

**Sensors**
- Push buttons, touchpads, tilt switches
- Variable resistors (Sliders, Volume knobs)
- Photoresistors (sensing light)
- Thermistors (temperature)
- Ultrasound (proximity range finder)

**Actuators**
- Lights, LED’s
- Motors
- Speakers
- Displays (LCD’s)
Types of Arduinos
What is the difference between them?
# Types of Arduino

<table>
<thead>
<tr>
<th>Arduino Board</th>
<th>Processor</th>
<th>Memory</th>
<th>Digital I/O</th>
<th>Analogue I/O</th>
</tr>
</thead>
<tbody>
<tr>
<td>Arduino Uno</td>
<td>16Mhz ATmega328</td>
<td>2KB SRAM, 32KB flash</td>
<td>14</td>
<td>6 input, 0 output</td>
</tr>
<tr>
<td>Arduino Due</td>
<td>84MHz AT91SAM3X8E</td>
<td>96KB SRAM, 512KB flash</td>
<td>54</td>
<td>12 input, 2 output</td>
</tr>
<tr>
<td>Arduino Mega</td>
<td>16MHz ATmega2560</td>
<td>8KB SRAM, 256KB flash</td>
<td>54</td>
<td>16 input, 0 output</td>
</tr>
<tr>
<td>Arduino Leonardo</td>
<td>16MHz ATmega32u4</td>
<td>2.5KB SRAM, 32KB flash</td>
<td>20</td>
<td>12 input, 0 output</td>
</tr>
</tbody>
</table>
Arduino Uno

Digital output
0, 1 Serial Port

In circuit serial programming

ATMEL Microcontroller

Analog input

Power supply

Power input

USB port
The Arduino can input and output analog signals as well as digital signals.

An analog signal is one that can take on any number of values, unlike a digital signal which has only two values: HIGH and LOW.

We will talk about this later!
How Arduino is programmed?

Using a software called Arduino IDE
This example code is in the public domain.

http://www.arduino.cc/en/Tutorial/Blink
*/

// the setup function runs once when you press reset or power the board
void setup() {
    // initialize digital pin LED_BUILTIN as an output.
    pinMode(LED_BUILTIN, OUTPUT);
}

// the loop function runs over and over again forever
void loop() { 
    digitalWrite(LED_BUILTIN, HIGH);  // turn the LED on (HIGH is the voltage level)
    delay(1000);
    digitalWrite(LED_BUILTIN, LOW);   // turn the LED off by making the voltage LOW
    delay(1000);                      // wait for a second
}

Arduino Software

How Arduinos are programmed
The Arduino Integrated Development Environment - or Arduino Software (IDE) - contains a text editor for writing code, a message area, a text console, a toolbar with buttons for common functions and a series of menus. It connects to the Arduino hardware to upload programs and communicate with them.
Arduino Language

- Simplified C/C++
- Based on the wiring project
  - http://wiring.org.co
- Peripheral libraries
  - LCD, sensors, 12C, etc.
## Useful functions

<table>
<thead>
<tr>
<th>Function</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>pinMode()</td>
<td>set pin as input or output</td>
</tr>
<tr>
<td>digitalWrite()</td>
<td>set a digital pin high/low</td>
</tr>
<tr>
<td>digitalRead()</td>
<td>read a digital pin’s state</td>
</tr>
<tr>
<td>analogRead()</td>
<td>read an analog pin</td>
</tr>
<tr>
<td>analogWrite()</td>
<td>write an “analog” PWM value</td>
</tr>
<tr>
<td>delay()</td>
<td>wait an amount of time</td>
</tr>
<tr>
<td>millis()</td>
<td>get the current time</td>
</tr>
</tbody>
</table>
Sketch

Declare variables at the top

Initialize
setup() - run once at the beginning
set pins

loop() - run repeatedly after setup()
Let’s see how a sketch for turning an LED light runs in an Arduino
Blinking LED

```cpp
int ledpin = 13;

void setup() {
  // initialize digital pin LED_BUILTIN as an output.
  pinMode(ledpin, OUTPUT);
}

// the loop function runs over and over again forever
void loop() {
  digitalWrite(ledpin, HIGH); // turn the LED on (HIGH is the voltage level)
  delay(5000); // wait for 5 seconds
  digitalWrite(ledpin, LOW); // turn the LED off by making the voltage LOW
  delay(5000); // wait for 5 seconds
}
```

Sketch uses 926 bytes (24%) of program storage space. Maximum is 12206 bytes.
Global variables use 9 bytes (0%) of dynamic memory, leaving 2099 bytes for local variables. Maximum is 2048 bytes.
Global variables

int ledPin = 13;
- LED connected to the control pin 13
setup()

pinMode(ledPin, OUTPUT);
loop()

digitalWrite(ledPin, HIGH);
delay(5000);
digitalWrite(ledPin, LOW);
delay(5000);
How to get started?
If you have the board
How to get started?

- Arduino board
  - USB cable
  - DC power supplies
- Download the Arduino’s software (Arduino IDE)
  - Read carefully
    - Instruccions to install and setup the Arduino board with the computer and software
  - Download the Arduino IDE software
    - http://www.arduino.cc
- Plug it in!
Plug in it into the computer

Writing and Downloading Code

Write sketch on PC

Download sketch to Arduino
Online
Tinkercad is a free, online 3D modeling program that runs in a web browser, known for its simplicity and ease of use. Since it became available in 2011 it has become a popular platform for creating models for 3D printing as well as an entry-level introduction to constructive solid geometry in schools.
Tinkercad

Make an account in tinkercad

It’s free!
Tinkercad Lesson Plans
Tinkercad lesson plans are ready to use online or in the classroom. Discover curriculum developed in partnership with teachers. Learn more

Circuits

Create new Circuit

Incredible Snicket
9 minutes ago
Private

Copy of Arduino Blink
11 minutes ago
Private

Copy of Blink (Blocks)
2 months ago
Private

Fabulous Stantia-Borwo
7 months ago
Private
Scroll down until you find the Arduino Uno
Find the resistor and the LED

Connect the resistor to the GND (Ground)

Connect the LED: Shorter leg with the resistor and longer leg to pin 12
Write the code

After writing the code, start the simulation here to see the circuit work with the Arduino!
// C++ code
void setup()
{
  pinMode(13, OUTPUT);
}

void loop()
{
  digitalWrite(13, HIGH);
  delay(1000); // Wait for 1000 millisecond(s)
  digitalWrite(13, LOW);
  delay(1000); // Wait for 1000 millisecond(s)
Button

```cpp
// constants won't change. They're used here to set pin numbers:
const int buttonPin = 2; // the number of the pushbutton pin
const int ledPin = 13; // the number of the LED pin

// variables will change:
int buttonState = 0; // variable for reading the pushbutton status

void setup() {  
  // initialize the LED pin as an output:
  pinMode(ledPin, OUTPUT);
  // initialize the pushbutton pin as an input:
  pinMode(buttonPin, INPUT);
}

void loop() {  
  // 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);
  }  
}
```
Moving a stepper motor
Moving a stepper motor using a potentiometer
Using a load cell
More projects

You can find more projects at:

https://create.arduino.cc/projecthub
Any Questions
Hands On
Challenge!!

LED Control Using a button
Challenge!!

LED Brightness Control Using a Potentiometer
Challenge!!

Scrolling LED