

# Simulador en Thinkercad el Arduino y ESP82

## (Junio de 2020)

Iván Zambrano

**Resumen - El poder desarrollar soluciones con un diseño electrónico computacional se ha convertido en uno de los modos más convencionales y funcionales de la actualidad, esto conlleva a poner más atención en los elementos de hardware los cuales tienen una gran importancia, un ejemplo es que estos puedan realizar su función sin ninguna dificultad que se deba a su fabricación. Gracias a varias nuevas plataformas que han venido surgiendo en los últimos años el tener que desarrollar desde cero sea simplificado enormemente**

**Índice de Términos - Tinkercad, Arduino, simulador,**

### I. INTRODUCCION

Gracias a las nuevas plataformas que se han creado, los desarrolladores han podido manejar el desarrollo y diseño de distintos proyectos con una mayor rapidez y eficiencia en lo que son los diferentes procesos que estos conllevan. Actualmente la adquisición de dichas plataformas sea vuelto mucho más fácil y barato, al punto de llegar a utilizar algunas plataformas de manera gratuita.

También en otro apartado de lo que se refiere hardware, los controladores como lo son los Arduinos Uno son los más utilizados por sus cualidades al momento de programarlos ya que su programación es estándar y eso ayuda a muchos desarrolladores al momento de generar interfaces para cualquier dispositivo.

Tinkercad es una plataforma online brindada por Autodesk, una de sus principales características es el hecho de poder utilizarla de forma gratuita, antes de ello es esencial la creación de una cuenta de dicha plataforma. Existen varios servicios que nos provee la plataforma por el que se la conoce en todo el mundo principalmente es ser una herramienta para el diseño de piezas en 3D, pero en este documento nos centraremos en otro servicio que igualmente ofrece la plataforma es cuál es la posibilidad de simular circuitos electrónicos como también el hecho de poder programar los mismos, especialmente en Arduino.

### II. HARDWARE

#### A. Arduino UNO

ARDUINO UNO es un microcontrolador el cual está basado en ATmega 328P. Tiene 14 entradas/salidas digitales, de las cuales 6 se pueden utilizar como salidas PWM (Modulación por ancho de pulsos) y otras 6 son entradas analógicas.

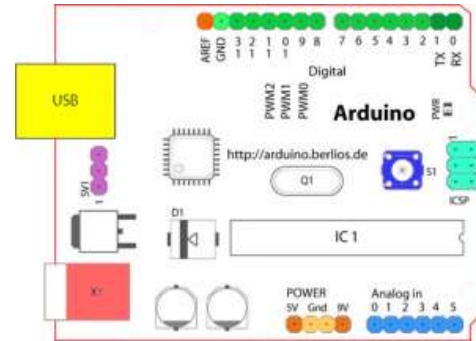


Fig 1 Conectores del Arduino UNO

1. Terminal de referencia analógica (naranja)
2. Tierra digital (verde claro)
3. Terminales digitales 2-13 (verde)
4. Terminales digitales 0-1/ E/S serie – TX/RX (verde oscuro) – Estos pines no se pueden utilizar como e/s digitales (digitalRead() y digitalWrite()). Botón de reinicio – S1 (azul oscuro)
5. Programador serie en circuito «In-circuit Serial Programmer» o «ICSP» (azul celeste). Terminales de entrada analógica 0-5 (azul claro)
6. Terminales de alimentación y tierra (alimentación: naranja, tierras: naranja claro)
7. Entrada de alimentación externa (9-12VDC) – X1 (rosa)
8. Selector de alimentación externa o por USB – SV1 (púrpura). En las versiones nuevas de Arduino la selección de alimentación es automática por lo que puede que no tengas este selector.
9. USB (utilizado para subir programas a la placa y para comunicaciones serie entre la placa y el ordenador; puede utilizarse como alimentación de la placa) (amarillo)

#### B. Tinkercad

Tinkercad es una plataforma de diseño de circuitos en línea gratuito, diseño en 3D y como plataforma de impresión 3D desarrollada y administrada por AUTODESK. Se lo utiliza para diseñar y simular circuitos electrónicos.

Tinkercad también permite generar el código C / C ++, por bloques, texto o mixta. Ésta última muestra las dos opciones (bloques y texto) simultáneamente. Pero comunmente se suele utilizar el código del mismo Arduino UNO. Para programar un Arduino, el lenguaje estándar es C++, aunque es posible programarlo en otros lenguajes. No es un C++ puro sino que es una adaptación que proviene de avr-libc que provee de una librería de C de alta calidad para usar con GCC en los microcontroladores AVR de Atmel y muchas funciones específicas para los MCU AVR de Atmel.

### C. ESP8266

*ESP8266, ésta es una placa de desarrollo de bajo coste, de tamaño reducido, el cual tiene la gran ventaja de conectarse a internet, contiene 1 puerto analógico, 9 pines GPIO con I2C y SPI, cuenta adicionalmente con 4MB de memoria FLASH (32 Mbit) (GrikoGeek, 2019). En otras palabras, es un chip integrado con conexión Wifi y compatible con el protocolo TCP/IP. El objetivo principal es dar acceso a cualquier microcontrolador a una red.*

#### Hardware

- Utiliza una CPU Tensilica L106 32-bit
- Voltaje de operación entre 3V y 3,6V
- Corriente de operación 80 mA
- Temperatura de operación -40°C y 125°C

#### Conectividad

- Soporta IPv4 y los protocolos TCP/UDP/HTTP/FTP
- No soporta HTTPS en un principio. Sí que lo hace mediante software tanto en cliente como servidor TLS1.2. La primera implementación está todavía en desarrollo.

### Proyecto: Temperatura del Clima

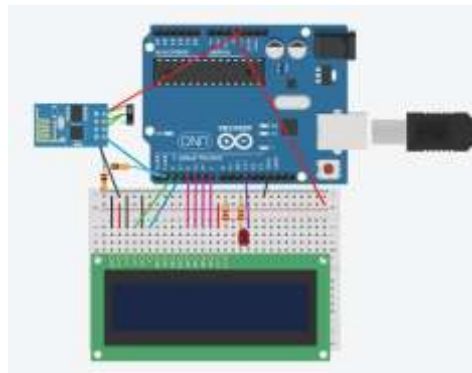


Fig. 2 Proyecto en Tinkercad

#### Código:

```
#include <LiquidCrystal.h>

LiquidCrystal lcd(2, 3, 4, 5, 6, 7);

int errorLED = 11;

String ssid = "Simulator Wifi"; // SSID to connect to
String password = ""; // Our virtual wifi has no password (so dont do your banking stuff on this network)

String host = "api.openweathermap.org"; // Open Weather Map API
const int httpPort = 80;
String uri =
"/data/2.5/weather?q=San%20Francisco,us&appid=1a702a15a2f46e405e61804cf67c0d30&units=imperial";

// the setup routine runs once when you press reset:
void setup() {
  // Setup LCD and put some information text on there
  lcd.begin(16,2);
  lcd.print("Temperature SF:");
  lcd.setCursor(0,1);
  lcd.print("F: ");

  pinMode(errorLED, OUTPUT); // init our red error LED

  // Start our ESP8266 Serial Communication
  Serial.begin(115200); // Serial connection over USB to
  computer
  Serial.println("AT"); // Serial connection on Tx / Rx port to
  ESP8266
  delay(10); // Wait a little for the
  ESP to respond
  if (!Serial.find("OK")) digitalWrite(errorLED, HIGH); // check if the ESP is
  running well

  // Connect to 123D Circuits Simulator Wifi
  Serial.println("AT+CWLAP=\"" + ssid + "\",\" + password + "\"");
  delay(10); // Wait a little for the
  ESP to respond
  if (!Serial.find("OK")) digitalWrite(errorLED, HIGH); // check if the ESP is
  running well

  // Open TCP connection to the host:
  Serial.println("AT+CIPSTART=\"TCP\", \" + host + "\", \" + httpPort);
  delay(50); // Wait a little for the
  ESP to respond
  if (!Serial.find("OK")) digitalWrite(errorLED, HIGH); // check if the ESP is
  running well

}

// the loop routine runs over and over again forever:
void loop() {
```

```

// Construct our HTTP call
String httpPacket = "GET " + uri + " HTTP/1.1\r\nHost: " + host + "\r\n\r\n";
int length = httpPacket.length();

// Send our message length
Serial.print("AT+CIPSEND=");
Serial.println(length);
delay(10); // Wait a little for the ESP to respond
if (!Serial.find(">")) digitalWrite(errorLED, HIGH); // check if the ESP is
running well

// Send our http request
Serial.print(httpPacket);
delay(10); // Wait a little for the ESP to respond
if (!Serial.find("SEND OK\r\n")) digitalWrite(errorLED, HIGH); // check if
the ESP is running well

while(!Serial.available()) delay(5); // wait until we receive the
response from the server

if (Serial.find("\r\n\r\n")){ // search for a blank line which defines the
end of the http header
  delay(5);

  unsigned int i = 0; //timeout counter
  String outputString = "";

  while (!Serial.find("\r\n\r\n")){ // find the part we are interested in.

  while (i<60000) { // 1 minute timeout checker
    if(Serial.available()) {
      char c = Serial.read();
      if(c==',') break; // break out of our loop because we got all we need
      outputString += c; // append to our output string
      i=0; // reset our timeout counter
    }
    i++;
  }

  lcd.setCursor(3,1); // set our LCD cursor to the correct position
  lcd.print(outputString); // push our output string to the LCD

}

delay(10000); // wait 10 seconds before updating
}

```

## IX. CONCLUSIÓN

- Las plataformas digitales se convierten en una gran herramienta las cuales nos ayudan a poder diseñar nuestros circuitos, en el caso de no poder conseguir alguno de ellos. Y como complemento se vuelve muy útil a la hora de guardar nuestros proyectos que por otro lado tendríamos que desarmar nuestro circuito para diseñar uno nuevo.
- La Plataforma TinkerCad resulta ser una buena herramienta a la hora de poder hacer circuitos los cuales podemos codificar con Arduino sin tener que salir de la misma.

## REFERENCES

- Troncoso, M., & Nicolás, D. (2018). Prototipo de seguimiento en tiempo real implementado en Arduino y microcontrolador ESP8266.
- Sánchez, E. (2012). Diseño de un sistema de control domótico basado en la plataforma Arduino. *Master's thesis. Escuela Técnica Superior de Ingeniería Informática. Universidad Politécnica de Valencia.*
- C. Vidal, C. Cabezas, J. Parra y L. López, (S.F.) "Experiencias prácticas con el uso del lenguaje de programación Scratch para desarrollar el pensamiento algorítmico de estudiantes en Chile " Formación Universitaria, vol. 8, págs. 23–32, 2015. [En línea]. Disponible: <http://www.redalyc.org/articulo.oa?id=373544191001>
- European-Schoolnet, (2015) Calculando nuestro futuro: programación informática y Codificación de prioridades, programas escolares e iniciativas en toda Europa. Bruselas, Bélgica.
- Vidal-Silva, C., Lineros, M. I., Uribe, G. E., & Olmos, C. J. (2019). Electrónica para Todos con el Uso de Arduino: Experiencias Positivas en la Implementación de Soluciones Hardware-Software. *Información tecnológica*, 30(6), 377-386.