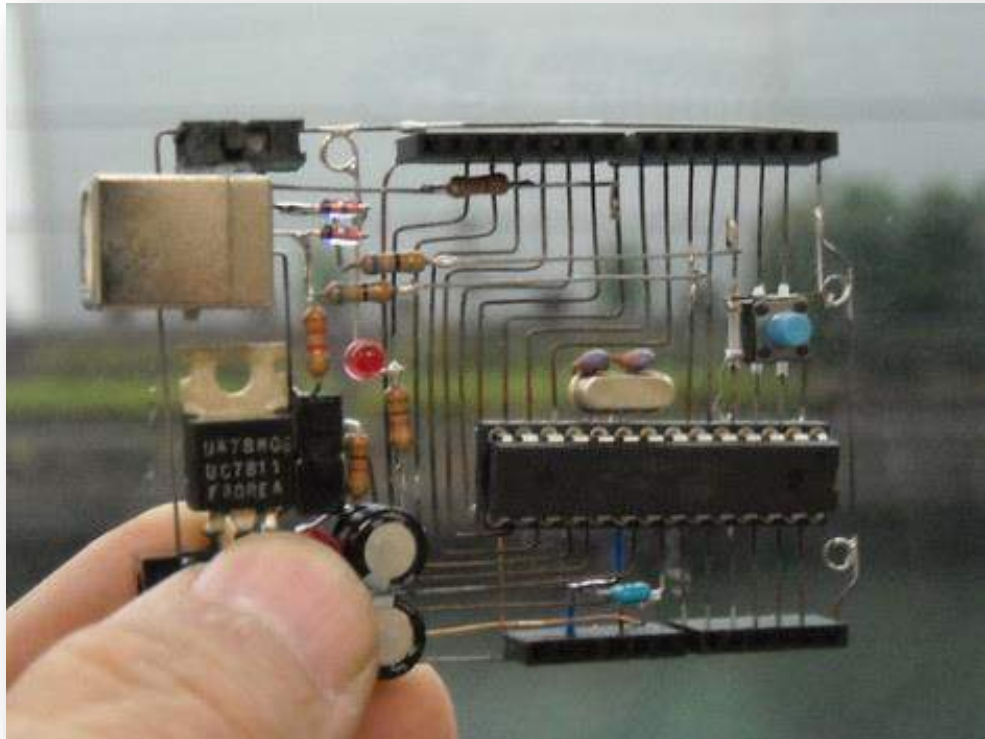


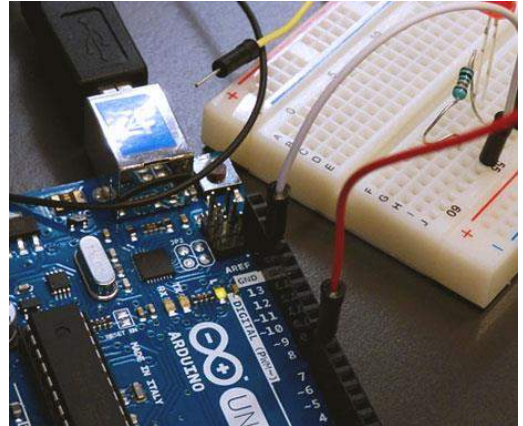
Manual de Introducción a la Electrónica



Introducción a la Electrónica Básica con el uso de herramientas modernas, didácticas y fáciles de conseguir.

¿Qué es la Electrónica?

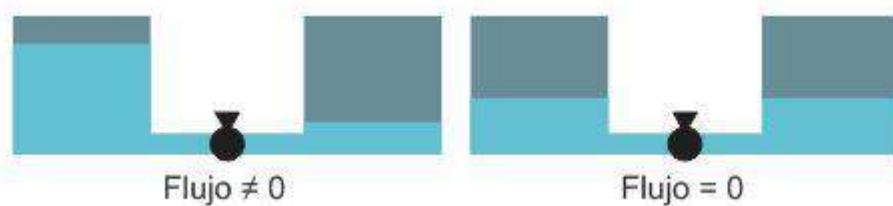
Es una ciencia que se encarga del estudio, control y aplicación de pequeños y grandes sistemas que nos permiten el tratamiento, almacenamiento e intercambio de información contenida en señales eléctricas. Hoy en día está presente en TODO, no hay nada que escape de la electrónica, desde lo más pequeño hasta lo más grande. Qué nos lleva a estudiar electrónica? Aprender cómo es que se mueve el mundo en base a esta ciencia, lograr hacer proyectos utilizables en nuestro día a día y sobre todo jugar!



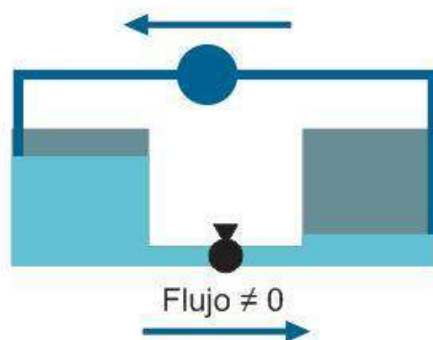
Conceptos claves para entender un poco más de Electrónica

Antes que nada daremos una breve explicación de la teoría básica necesaria, para entender ciertos conceptos claves de la Electrónica.

El voltaje, o diferencia de potencial, puede ser asimilado como la diferencia de nivel entre dos tanques conectados. Si no hay diferencia de nivel, no puede haber flujo. Lo mismo ocurre con la corriente. Si no hay diferencia de potencial, no hay corriente. Sin embargo, hay otro requisito, y es que el flujo sea permitido por las conexiones entre los tanques. Es decir, mientras nadie le permita el paso, a pesar de tener voltaje, la corriente será 0.



Ahora bien, contamos con una diferencia de potencial estable. Para que esto ocurra, usamos una fuente de voltaje, que puede ser asemejada a una bomba de agua que mantiene la diferencia de nivel, permitiendo un flujo constante por diferencia de potencial.



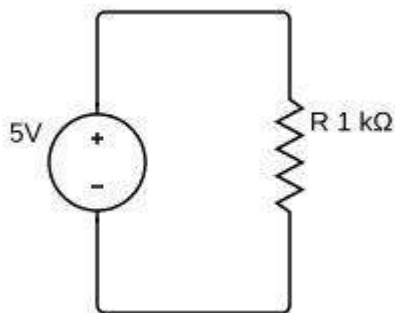
Sin embargo, la cantidad de agua que circula depende de la tubería. Mientras más angosta sea, menos agua podrá circular. Esto se asemeja a las **Resistencias**, que limitan el flujo de la corriente.

Es hora de ponerle unidades. La diferencia de potencial (diferencia de nivel) se mide en voltios (V); la intensidad de corriente (flujo de agua), en amperios (A) y la resistencia (el ancho de la tubería), en ohmios (Ω). La relación entre las variables anteriormente mencionadas es descrita por la famosa ley de Ohm:

$$\text{Voltaje} = \text{Corriente} \times \text{Resistencia}$$
$$V = I \times R$$

Puesto que un amperio es una unidad bastante grande de corriente, lo normal es trabajar con el miliamperio (mA).

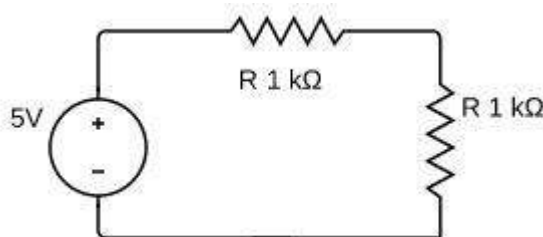
Realizaremos el cálculo correspondiente para el siguiente circuito:



Puesto que tenemos como datos el voltaje (5V) y la resistencia (1 K Ω), calcularemos la corriente, mediante la ley de ohm. Así de simple, obtenemos la respuesta: **5 mA** (o también 0.005 A, aplicar conversión de unidades, 1A = 1000 mA).

Siguiente concepto importante, la resistencia equivalente. Se sigue aplicando la ley de Ohm, pero con una consideración, en este caso:

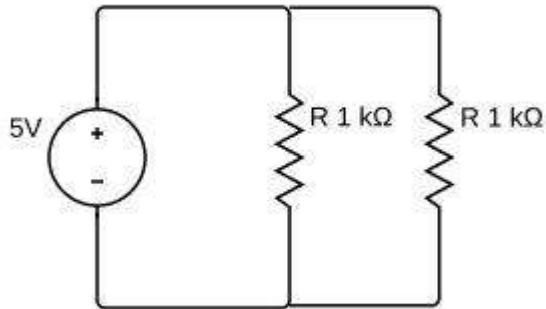
La resistencia equivalente de "n" resistencias en serie es igual a la suma de dichas resistencias



Entonces ambas resistencias pueden ser reemplazadas por su equivalente, es decir, una resistencia de 2K Ω . Realizando nuevamente los cálculos, obtenemos el nuevo valor de la corriente: 2.5 mA

$$R_e = R_1 + R_2 + R_3 + \dots + R_n$$

La resistencia equivalente de "n" resistencias en paralelo es igual a la inversa de la suma de sus inversas"



Entonces ambas resistencias pueden ser reemplazadas por su equivalente, es decir, una resistencia de 500Ω. Realizando nuevamente los cálculos, obtenemos el nuevo valor de la corriente: 10 mA.

$$R_e = \frac{1}{\frac{1}{R_1} + \frac{1}{R_2} + \frac{1}{R_3} + \dots + \frac{1}{R_n}}$$

Notamos aquí algo curioso, en ambos casos tuvimos la misma cantidad de resistencias, y estas fueron del mismo valor. La única diferencia fue la manera de conectarlas.

Sistemas Electrónicos

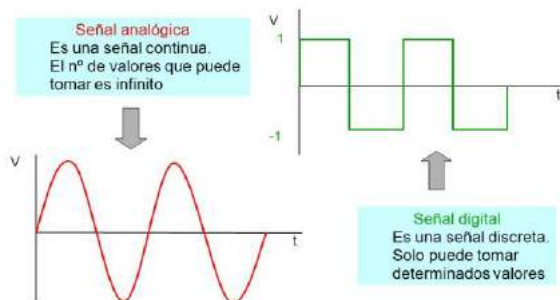
SISTEMAS ELECTRÓNICOS



Mejor explicado imposible! Gracias Wikipedia!

Digital vs Analógico

Señal Analógica y Señal Digital



Para ejemplificar esto tomaré de ejemplo un interruptor, solo tienes dos estados posibles (encendido y apagado), a esto lo llamamos Digital, tiene dos valores predeterminados. Un ejemplo de algo Analógico es el volumen de tu celular al escuchar música, puede tomar distintos valores, cuando subimos de volumen 3 a 4, hay una variación y aumenta el volumen, pero entre el 3 y 4 hay aún más divisiones y más divisiones.

Estudiar la electrónica puede llegar a ser todo un dolor de cabeza, sobre todo por lo amplio que es, además que requiere de mucho tiempo (aproximadamente... toda tu vida), dedicación, esfuerzo, sangre, sudor, lágrimas, electrocutadas y más electrocutadas. Entonces ¿Cómo aprenderemos electrónica? ¿Qué herramientas usaremos para aprender electrónica de una manera didáctica, sencilla y hasta fácil? La respuesta es sencilla: Arduino. Con esta herramienta vamos a aprender más sobre electrónica, y podremos montar nuestros pequeños proyectos de una manera práctica y sencilla.

¿Qué es el Arduino?

El Arduino es una plataforma de código abierto (open source), que nos permite crear objetos y entornos electrónicos de una manera didáctica y sencilla. Al ser una plataforma incluye software y hardware. Fue creado para que artistas, diseñadores, comunicadores y en general las personas puedan ingresar al mundo de la electrónica sin complicaciones.



Ventajas del Arduino:

1. BARATO: Es relativamente cómodo el precio del Arduino frente a otras plataformas similares, inclusive podemos armarnos nuestro propio Arduino!
2. OPEN SOURCE: Al hablar de Open Source nos referimos a que el código es libre para que cualquiera pueda usarlo. Es tanto Software Libre (código libre) como Hardware Libre (lo mencionado en la opción anterior).
3. MULTIPLATAFORMA: No solo es instalable en Windows, también en Linux y en Macintosh OSX. Esta es una ventaja clara frente a otros microcontroladores, que solo tienen software para Windows.
4. IDE SENCILLO: El IDE o software para programar en Arduino es muy fácil de entenderlo y usarlo.

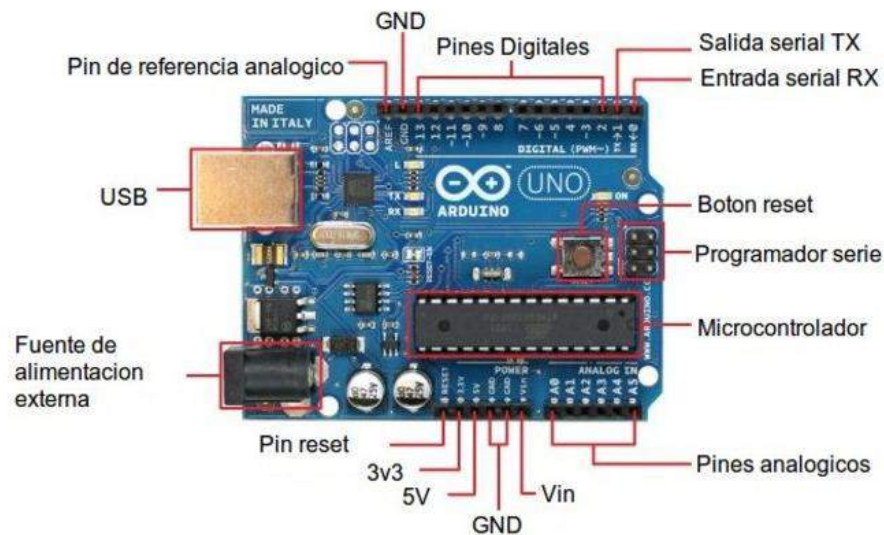


Existen arduinos para todos los gustos, de múltiples tamaños y muchas opciones extra de acuerdo a que queramos hacer con este.

Para nuestro entrenamiento utilizaremos el Arduino Uno, el más común y con el que se empieza a jugar :D

PD: Por si la curiosidad, el término Arduino significa GRAN AMIGO

Partes del Arduino:



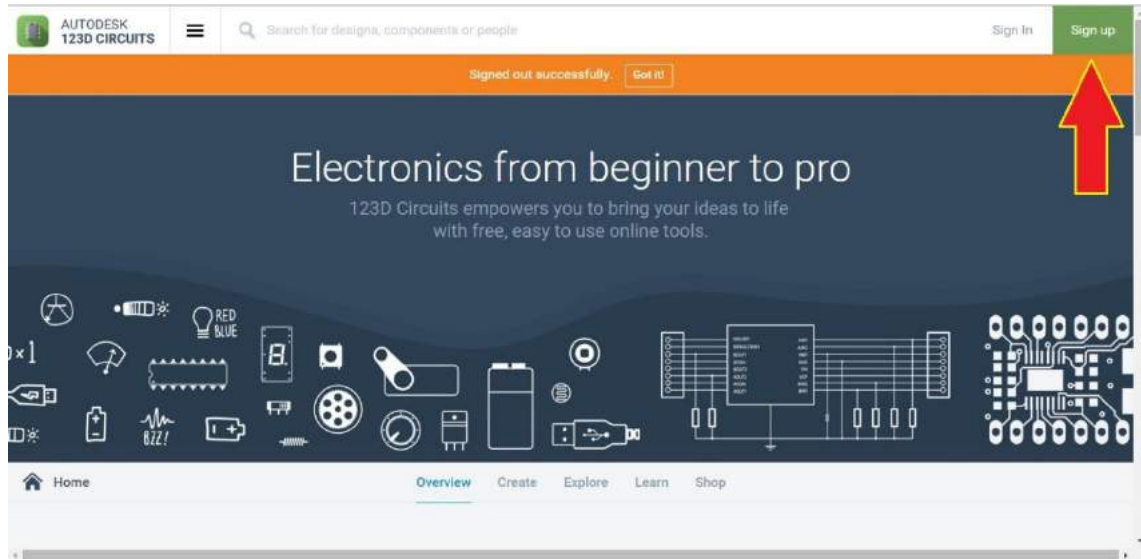
1. USB: Alimentación y comunicación con la PC.
2. GND: Pines de tierra. Punto de referencia que equivale a 0v.
3. PINES DIGITALES: Pines de entrada y salida de señales digitales (leen y escriben 0 o 1).
4. PINES ANALÓGICOS: Pines de entrada de señales analógicas (leen entre 0 voltios a 5 voltios)
5. SALIDA RX, TX: Pines para comunicación externa del Arduino, por ejemplo Bluetooth.
6. MICROCONTROLADOR: El cerebro del Arduino. Es la parte esencial, sin esto no funciona.
7. Vin: Pin de entrada de alimentación del Arduino.

Sé que para muchos por el momento será difícil adquirir un Arduino UNO y su respectivo Kit básico para empezar a practicar, así que usaremos una herramienta de internet para iniciar en nuestros proyectos y pequeñas prácticas desde casa. No se preocupen, para el entrenamiento presencial si contaremos con todo lo necesario para empezar a desarrollar y conquistar el mundo!

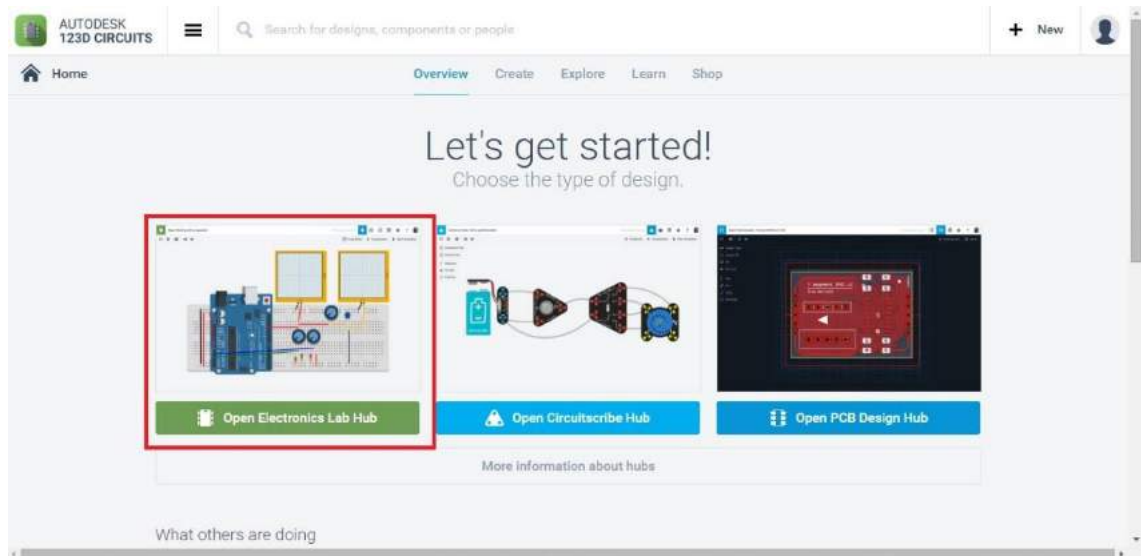


Iniciando en 123D Circuits

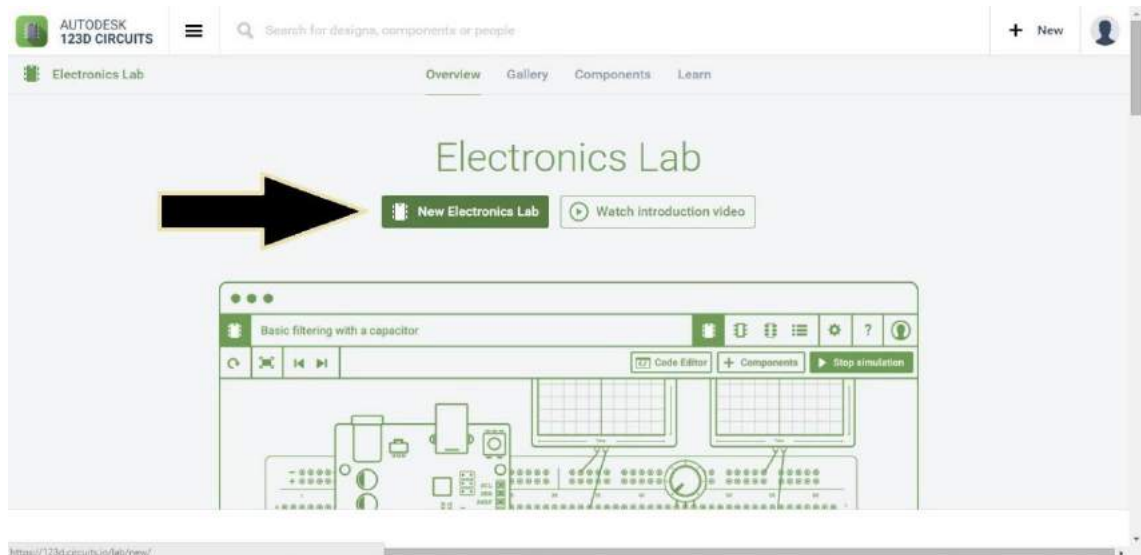
Comencemos! Lo primero que haremos es conocer una interfaz para realizar nuestras primeras pruebas de electrónica: 123D Circuits. Para poder usar el 123D Circuits a plenitud, necesitamos una cuenta, así que a registrarnos. Ingresar a <http://123d.circuits.io/> y crear una cuenta. Para eso clickeamos donde nos señala la flecha. Luego pueden rellenar sus datos, eso les dejo de misión :D



Listo! Estamos logeados! Nos aparecerá una ventana así. Entremos a la opción enmarcada, aquí realizaremos todas las pruebas.



Podemos ver nuestro "pre-entorno". Si indagamos un poco hay demos de otros pequeños trabajos de electrónica de otras personas, que podemos replicar. Click en la selección que les deje.



Ya estamos en nuestro entorno. He encasillado 3 de las áreas que usaremos en 123D Circuits:

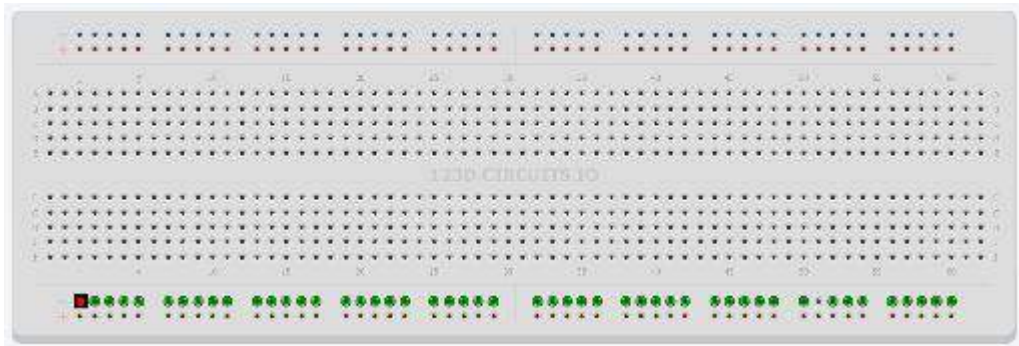
- AZUL:**
 De izquierda a derecha tenemos los botones Rotate, que se gira componentes en el protoboard; Delete, cuya función es eliminar componentes; zoom to fit, que ajusta el zoom para que podamos ver todos los objetos puestos en el área de trabajo; y por último tenemos las opciones Undo y su contraparte, para deshacer cambios recientes.
- AMARILLO:**
 De izquierda a derecha tenemos las vistas disponibles de nuestro proyecto: Vista en el protoboard, en donde podemos visualizar todos los componentes que agregamos (simulación de conexión en físico), Vista en el esquemático que nos muestra planos de conexiones de nuestro proyecto, y Vista en el PCB que nos muestra una vista desde lo mínimo del proyecto.
- CELESTE:**
 De izquierda a derecha tenemos lo siguiente: Code Editor para ingresar nuestro código (Arduino), Components que nos mostrará los componentes disponibles en 123D Circuits y el botón Start Simulation, para darle arranque a nuestra simulación (con o sin código incluido).



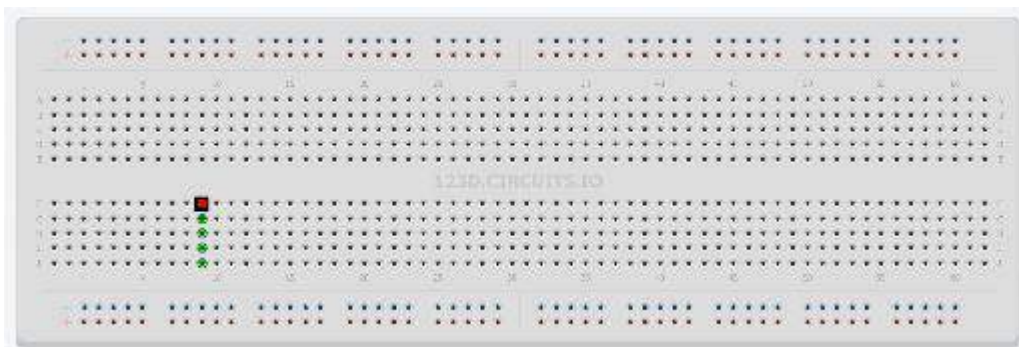
Y lo que les parece curioso a muchos, esa PLACA blanca enorme que tenemos ahí en medio. Su nombre es Protoboard, una placa de pruebas en donde podemos realizar todas nuestras conexiones previas.

¿Cómo funciona el protoboard? Tiene conexiones internas que facilitan el cableado. Estas están dispuestas de la siguiente manera:

- Los extremos tienen conexión horizontal interna. Cualquier cosa que se conecte a alguno de los pines en esa línea, también estará conectada a otra en la misma línea.



- La parte interna tiene conexiones verticales, pero la línea central funciona división, es decir las dos zonas centrales están aisladas entre sí.



Con esto hemos visto lo necesario para conocer nuestro entorno en el que trabajaremos estas semanas y además los conceptos básicos para poder investigar un poco más.

¡Un poquito de Código para terminar!

A lo que la mayoría le teme: al código, a la lógica, a programar. Puedo decir que no es nada de otro mundo, programar es poder ordenar nuestras ideas para realizar una determinada tarea valiéndonos de un software que tomará estos pasos y se los pasará a la computadora, esta las interpretará y nos devolverá resultados según lo que le pedimos. Iremos poco a poco, te dejaré algunas funciones necesarias para un programa de Arduino y la siguiente semana perfeccionaremos nuestro código.

Para la Estructura del código:

```
void setup() {
  /* Recordando: Aquí colocamos nuestras configuraciones.
  Pines de Entrada, Pines de salida, Pre-lectura de datos,
  etc */
}

void loop() {
  /* Y aquí colocamos la parte funcional del código.
  Secuencias, activación o desactivación de pines, envío y
  recepción de datos, etc */
}
```

Variables:

Como en todo lenguaje de programación, en el Arduino también declaramos variables para un mejor orden en nuestro código. La idea de programar no es hacer mucho código, es que con poco código hagamos mucho sin tener que estar dando vueltas, redundando y con mucho estrés por luego intentar entender nuestro propio código.

Aquí te dejare tres tipos de variables del Arduino, a mí parecer los principales y los que más se usan:

```
/* Declarando valores enteros:
Usualmente usados para darle un nombre
determinado a un pin del Arduino */

int led = 13;

void setup() {
  ...
}

void loop () {
  ...
}
```

```
/* Declarando caracteres:
Usualmente usado para almacenar
estados */

char estado = a;

void setup() {
  ...
}

void loop () {
  ...
}
```

```
/* Declarando booleanos
Dos posibles estados: True (Verdad o 1
lógico) o False (Falsedad o 0 lógico) */

Boolean encendido = true;

void setup() {
  ...
}

void loop () {
  ...
}
```

Funciones:

Usaremos dos tipos de funciones: Digitales y Analógicas, ya explicamos de este tema anteriormente, ya deben tener una idea de cómo funcionarán.

Digitales:

```
/* Función pinMode: Configura el modo
del Pin como entrada o salida
(Input o Output) */

int led = 13;

void setup {
  pinMode(led,OUTPUT);
  // El pin 13 configurado como
  salida
}

void loop {
  ...
}
```

```
/* Función digitalRead: Nos permite
la lectura digital de un pin
(lee un '0' o un '1') */

int val_entrada;

void setup {
  ...
}

void loop {
  val_entrada = digitalRead (13);
  // Leera el estado digital ingresante
  en el pin 13
}
```

```
/* Función digitalWrite: Le otorga un
estado de '0' y '1' al pin asignado
(Encendido o Apagado, HIGH o LOW) */

int led = 13;

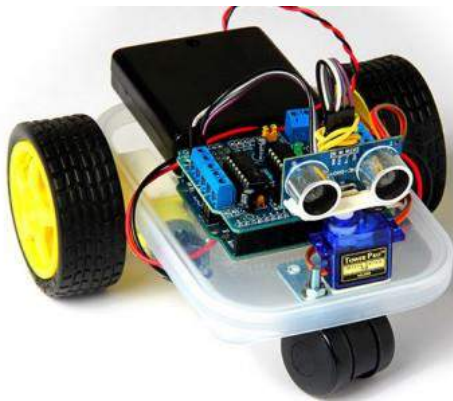
void setup {
  ...
}

void loop {
  digitalWrite(led,HIGH);
  // Emite un pulso ALTO al pin 13
  // HIGH: Encendido, LOW: Apagado
}
```

Analógicas:

```
/* Función analogRead: Lee un valor entre 0  
a 1023 en uno de los pines analógicos */  
  
int a;  
  
void setup() {  
  ...  
}  
  
void loop() {  
  int a = analogRead(A0);  
}
```

```
/* Función analogWrite: Escribe un valor de 0  
a 255 en uno de los pines analógicos.  
Usado usualmente para control de motores */  
  
void setup() {  
  ...  
}  
  
void loop() {  
  analogWrite(9,134);  
  //9 es el Pin, 134 es el valor Analogo escrito  
}
```



¡Sigán adelante! Este será el proyecto que desarrollaremos en el presencial, con nuestras modificaciones.

¿Y porque no lucir bien y diferente con algo de electrónica? ¡Y por supuesto el Arduino nos permite esto!

