

Práctica de laboratorio 1

Introducción a Arduino: Carga de un condensador

1. Objetivos de la práctica.

- Iniciación en el uso de instrumental de laboratorio.
- Iniciación al uso y programación de Arduino.
- Realizar un montaje para estudiar el proceso de carga de un condensador.

1. Introducción teórica.

Arduino es una plataforma de electrónica abierta para la creación de prototipos basada en software y hardware flexibles y fáciles de usar.

Arduino nació con el objeto de proporcionar una plataforma económica y fácil uso, para enseñar electrónica a los alumnos del Instituto de Diseño Interactivo de Ivrea (Italia)¹. Sus creadores y socios fundadores son los italianos Massim Banzi y Gianluca Martino, los estadounidenses Tom Igoe y David Mellis, y el español David Cuartielles, quienes decidieron que su diseño fuera abierto licenciándolo en Creative Commons⁷, protegiendo la marca, pero el hardware, el software y la documentación son libres, se pueden descargar los archivos de diseño del circuito electrónico, los códigos fuentes del software y se pueden modificar, compartir etc.

En la actualidad hay más de un millón de placas Arduino oficiales distribuidas por todo el mundo, y millones de usuarios visitan de forma regular el sitio web² cada trimestre.

Arduino encuentra un amplio uso en la enseñanza en materias relacionadas con la robótica, el control, la adquisición de datos, los diseños interactivos, etc. Así, la sencillez de uso de esta plataforma permite a personas no expertas en electrónica, utilizar en sus creaciones dispositivos electrónicos y controlarlos, de una manera sencilla para ellos.

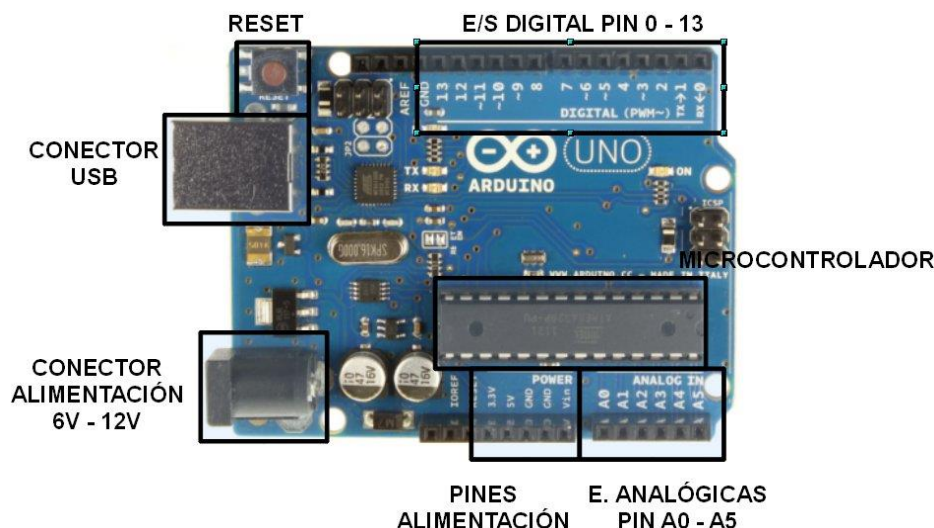
Concretamente, dentro de los diferentes modelos existentes en la actualidad, con sus diferentes características, en el laboratorio de esta asignatura nosotros contaremos y trabajaremos con Arduino UNO.

¹ http://www.eldiario.es/hojaderouter/tecnologia/hardware/arduino-hardware-libre-revolucion-origenes-historia_0_289771082.html

² <https://www.arduino.cc/>

Hardware

Arduino UNO es una plataforma que contiene un microcontrolador ATmega328, y que cuenta con de entradas analógicas, entradas y salidas digitales y salidas PWM (*Pulse Width Modulation*).



De todas las plataformas Arduino, la más representativa y extendida entre los usuarios es Arduino UNO R3, que es la versión actual de las primeras versiones de Arduino, y que suele tomarse como referente y es con la que se suele identificar al sistema Arduino al referirse a él sin más. La sencillez de uso, versatilidad y bajo coste hacen de Arduino UNO R3, una plataforma adecuada tanto como sistema final como sistema de desarrollo para proyectos a implementar con otras plataformas Arduino. Por este motivo es habitual tomar la plataforma Arduino UNO como punto de partida para adentrarse tanto en el mundo de Arduino (conociendo sus características, elementos, etc) como en el mundo de la electrónica.

Arduino UNO es un sistema basado en el microcontrolador de 8 bits ATmega328, de Atmel®, el cual es un chip sencillo y de bajo coste. Por otro lado, Arduino UNO cuenta con 14 pines (0 a 13), cada uno de los cuales se puede configurar como entrada o salida digital. Los pines configurados como salida pueden proporcionar o absorber una corriente de hasta 40 mA, suficiente para excitar multitud de circuitos, sensores, etc. aunque insuficiente para otros, tales como algunos relés, solenoides, motores, etc., para los que es preciso utilizar algún circuito acondicionador. Seis de estos pines se pueden configurar también como salidas PWM, lo que permite, variando el ciclo de trabajo de la señal cuadrada generada en el pin, obtener una tensión cuyo valor medio puede variar entre 0 y 5 V y de esta forma simular una salida analógica sobre una salida digital. La placa incluye un LED conectado al pin 13, lo que permite entre otras cosas utilizarlo como dispositivo de salida en la verificación y depuración de

programas. Los pines 0 y 1 de E/S digital se pueden configurar para utilizarse como un puerto serie.

A su vez, el microcontrolador ATmega328 que usa Arduino UNO, incluye un conversor analógico-digital (A/D) de 6 canales, con una resolución de 10 bits, retornando un valor entero entre 0 y 1023. El uso principal de estos pines es para la lectura de sensores analógicos. Estos pines tienen también toda la funcionalidad de los pines de entrada/salida digitales, como los pines 0 - 13. En consecuencia, si para alguna aplicación se precisan más pines de entrada-salida digital, y no se está usando ningún pin analógico, pueden usarse estos pines para ello. La placa contiene todo lo necesario para utilizar el microcontrolador, cristal del oscilador de 16 MHz, regulador de tensión, etc., siendo suficiente con conectarlo al ordenador a través del cable USB o alimentarlo desde una fuente de alimentación o una batería externa, con un voltaje recomendado de entre 7 a 9 V, para empezar a trabajar con él. La placa cuenta con 32 KB de memoria Flash, 2 KB de memoria SRAM y 1 KB de memoria EEPROM.

Software

En cuanto al software necesario para programar la placa es necesario descargarse e instalar el entorno de desarrollo (IDE)³, así como los diferentes drivers necesarios para su buen funcionamiento.

El entorno es de código abierto y facilita la labor de escribir el código necesario para programar el microcontrolador, así como cargar los códigos fuente en la memoria de la placa. El entorno funciona en Windows, Mac OS X y Linux y está escrito en Java. El lenguaje de programación de Arduino tiene grandes similitudes con el lenguaje C++.

³ <https://www.arduino.cc/en/Main/Software>

```

Blink
/*
  Blink
  Turns on an LED on for one second, then off for one second, repeatedly.

  This example code is in the public domain.
  */

// Pin 13 has an LED connected on most Arduino boards.
// give it a name:
int led = 13;

// the setup routine runs once when you press reset:
void setup() {
  // initialize the digital pin as an output.
  pinMode(led, OUTPUT);
}

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

```

2. Instrumentación y Componentes.

- Placa Arduino UNO.
- Resistencia de 10 K Ω .
- Resistencia de 220 Ω .
- Condensador de 470 μF .

3. Estudio Teórico.

Dado el esquemático de la siguiente figura, y sabiendo que $R1$ tiene un valor de 10K Ω y el condensador $C1$ 470 μF , calcular el tiempo que tardará en cargarse el condensador. Dibujar el proceso de carga y de descarga de dicho condensador.

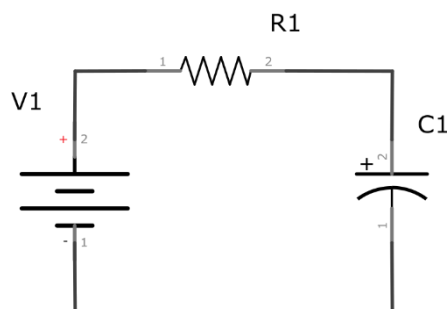


Figura 1.- Circuito para estudio teórico

Nota: La carga y descarga del condensador en circuito RC con fuente de tensión constante (escalón) viene expresada por la siguiente fórmula (véase Apéndice A de la asignatura: Análisis de Circuitos, páginas a10-a12):

$$v_c(t) = (v_c(0) - v_c(\infty))e^{-t/RC} + v_c(\infty)$$

4. Desarrollo de la práctica.

Utilizar un Arduino UNO para realizar un montaje que permita estudiar el proceso de carga y descarga de un condensador. Para ello, realice el siguiente circuito:

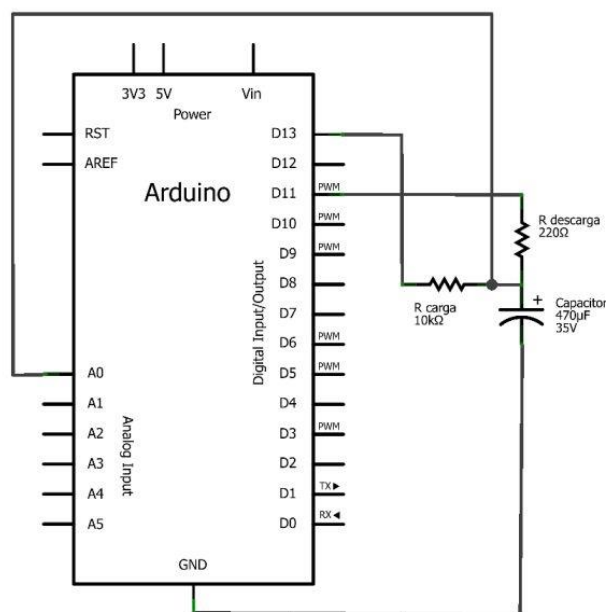


Figura 2.- Esquemático

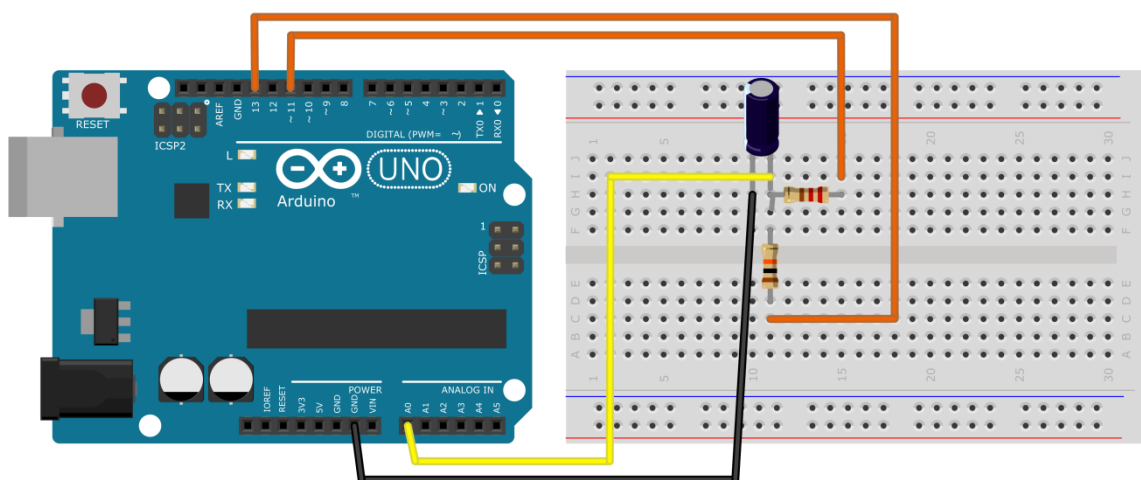


Figura 3.- Montaje

Como se deduce del estudio teórico de la práctica, el tiempo de carga de un condensador en un circuito RC está relacionado con el parámetro τ de tiempo, correspondiente al producto de R y C . Concretamente, se demuestra que un condensador se encuentra totalmente cargado cuando ha transcurrido aproximadamente 5τ .

El objetivo de la práctica se centra en medir y registrar con Arduino la tensión del condensador conforme se realiza la carga del mismo; para posteriormente poder representar la curva obtenida con la ayuda de una hoja de cálculo (p.e.: Microsoft Excel).

Así, mediante la entrada analógica A0 de Arduino mediremos la tensión del condensador. Arduino proporciona este valor discretizando el rango de tensión (de 0 a 5 V) en 1024 valores (de 0 a 1023).

Concretamente, para realizar el proceso de carga del condensador, y su correspondiente registro, serán necesarios los siguientes pasos:

- Partir del condensador descargado (Configurando para ello las salidas 11 y 13 de Arduino a nivel bajo -LOW-).
- Esperar a que el valor de la entrada A0 tenga el valor cero (en ese momento el condensador ya está totalmente descargado).
- Configurar la entrada/salida 11 en este caso como entrada -INPUT- y la 13 en valor alto -HIGH-, para llevar a cabo la carga del condensador.
- Medir el valor de la entrada A0.
- Finalizar el proceso cuando el tiempo llegue a 2τ .
- Importar los valores registrados en la consola del IDE a un fichero Excel y representar en una gráfica X-Y.

El código fuente para Arduino necesario para realizar el anterior proceso de medición de la carga del condensador es el siguiente:

```
#define analogPin 0 // entrada analógica para medir la tensión del condensador
#define chargePin 13 // Pin para carga del condensador
#define dischargePin 11 // Pin para descarga del condensador

unsigned long startTime;
double chargeTime;
unsigned int readSensor;

void setup () {
    pinMode (chargePin, OUTPUT ); // Pone el pin de carga como salida
    digitalWrite (chargePin, LOW );
    Serial.begin (9600); // Inicializa la comunicación serie
}

void loop(){

    /* Descarga del condensador */
    digitalWrite (chargePin, LOW );
    pinMode (dischargePin, OUTPUT );
    digitalWrite (dischargePin, LOW );
    readSensor = analogRead (analogPin);
    while (readSensor > 0) {
        readSensor = analogRead (analogPin);
    }
    pinMode (dischargePin, INPUT ); // Pone el pin de descarga en HI
    Serial.println ( "----CARGA-----" ); // Indica el comienzo de la carga

    /* Carga del condensador */
    digitalWrite (chargePin, HIGH );
    startTime = micros () / 1000000.0;
    readSensor = analogRead (analogPin);

    do {
        chargeTime = (micros () / 1000000.0) - startTime;
        Serial.print (chargeTime);
        Serial.print ( "\t" );
        Serial.println (readSensor);
        readSensor = analogRead (analogPin);
    } while (chargeTime < 9.4); /* Tiempo de carga igual a 2 RC */
    delay (1000);
    exit(0);
}
```

Responda a las siguientes preguntas:

1. Una vez hecha la importación a Excel de los valores registrados en la consola, realice una conversión (en una nueva columna de Excel) para pasar de los valores de lectura codificados (de 0 a 1023) a valores de tensión (de 0 a 5 voltios).
2. Modificar el código fuente de Arduino para comprobar si en 5τ segundos el condensador se ha cargado totalmente. ¿Se ha cargado? ¿Qué tiempo tarda en cargarse totalmente el condensador?
3. El anterior código fuente sólo lleva a cabo un registro del proceso de carga del condensador. Modifique el código de Arduino anterior para que también se registre la descarga del condensador.