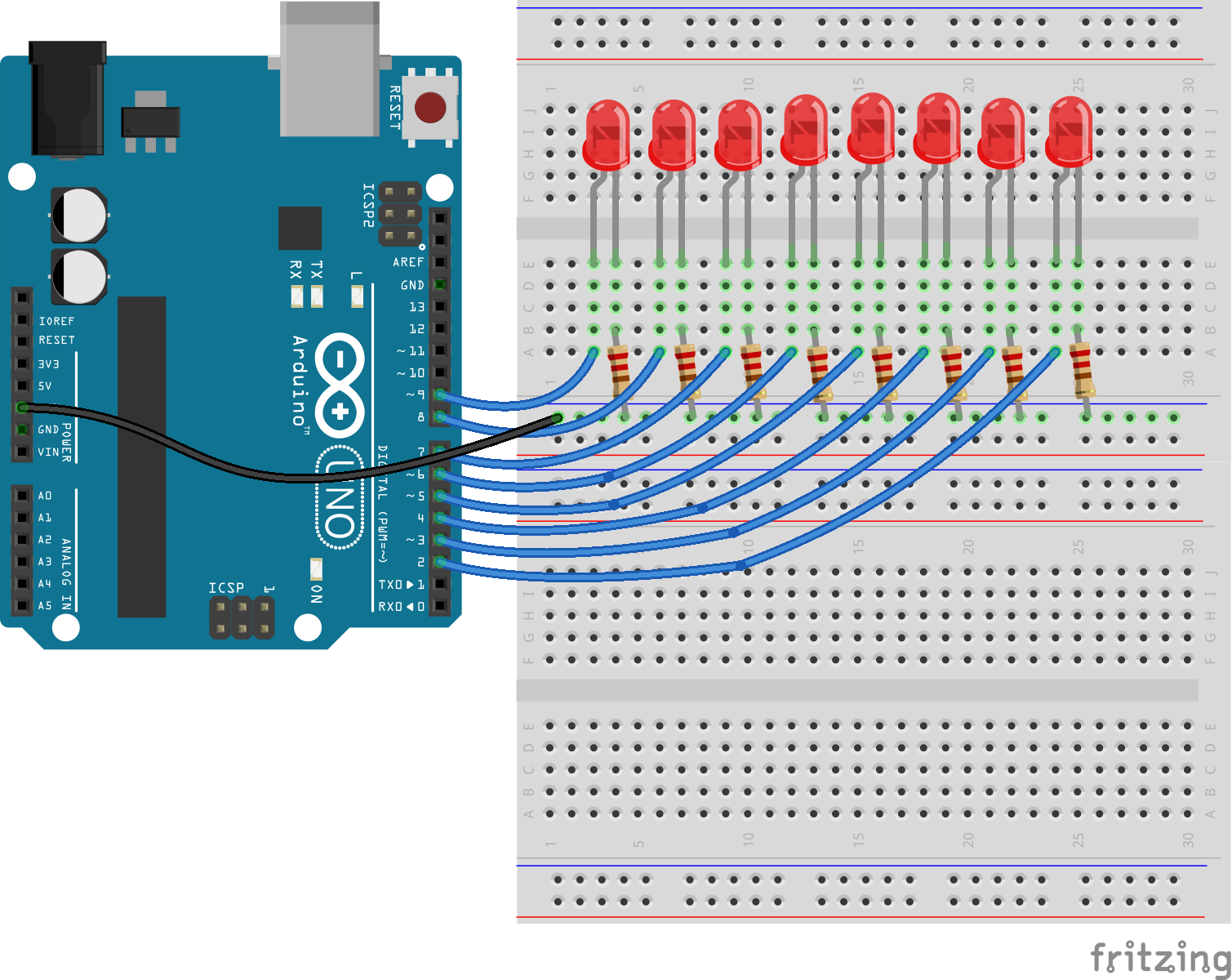
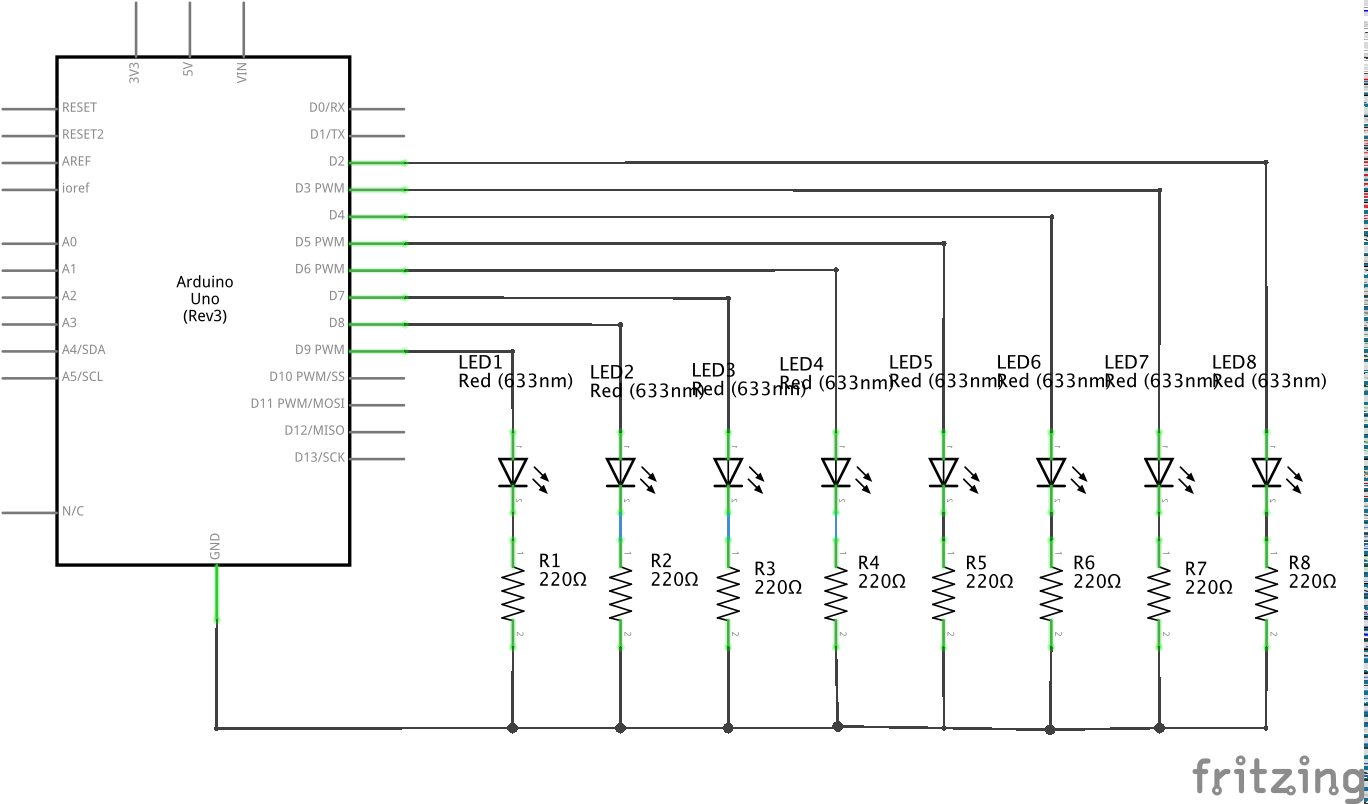
**DECODIFICADOR ASCII – BINARIO**

* **Descripción:** El objetivo de la práctica consiste en decodificar a binario el valor ASCII de un carácter introducido a través de la consola. Para la representación en binario se emplearán 8 LEDs.
* **Objetivos:** Con el presente ejercicio se intenta poner en práctica la conversión a binario y la utilización de la consola serial de Arduino.
* **Ámbito de aplicación:** informática / tecnología
* **Nivel:** medio
* **Conocimientos previos:** El alumnado debe ser capaz de reconocer el sistema de codificación de caracteres alfanuméricos (ASCII) y tener conocimientos de codificación a binario.
* **Diagrama o Esquema:** Se podría realizar un esquema empleando la aplicación *fritzing* (<http://fritzing.org/home/)>





* **Material necesario:** listado de todo el material requerido para la correcta realización de la práctica.
  + 1 placa Arduino.
  + 8 x LED 5mms.
  + 8 x Resistencia de 220Ω
* **Desarrollo de la práctica:** Describir los diferentes pasos a seguir adjuntando imágenes de los más importantes.

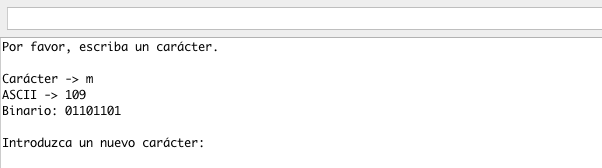
Para la introducción del carácter a través de la consola serial, es necesario utilizar los pines 0 y 1. De este modo, para la representación en binario se emplearán los pines del 2 a 9.

En primer lugar activamos los pines 0-9 como salida, y para ello hacemos uso de un bucle y de la función *pinMode()*.

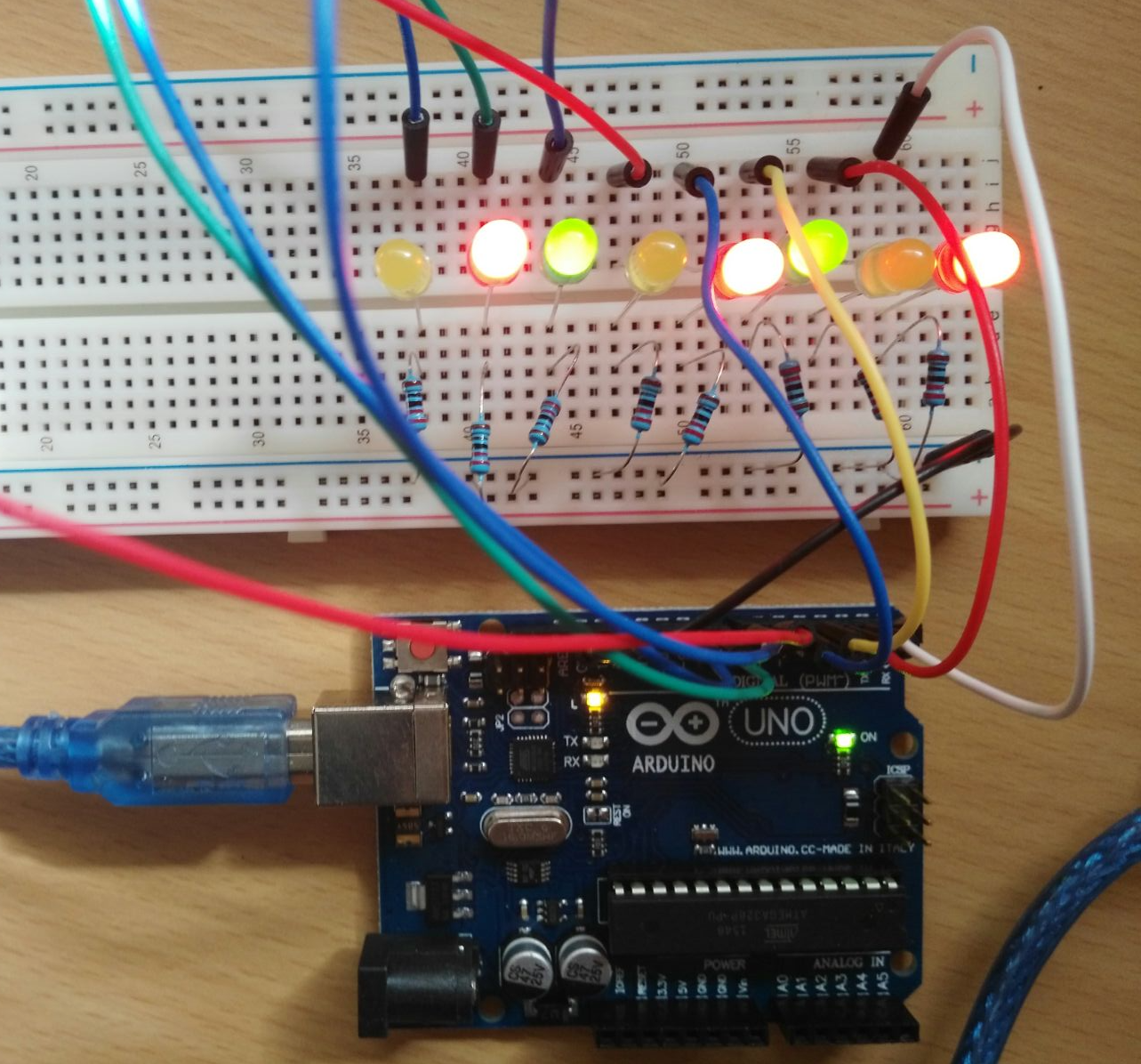
Posteriormente, se activa el canal de comunicación serial a 9600 baudios.

Una vez insertado el carácter por teclado, comprobamos cada led empezando en el menos significativo. Para esto, se utiliza la función bitRead() que devuelve HIGH si el bit está activo y LOW en caso contrario. Finalmente, los valores devueltos por bitRead() son representados en los pines 2 a 9.

A continuación, podemos ver cómo funciona en el caso de introducir el carácter “m”.



El código ASCII para el carácter “m” es 109, así que en binario será representado como 01101101. En la siguiente imagen se puede apreciar como los diferentes LEDs son activados para representar el valor en binario.



* **Código:** A continuación, se puede consultar el código fuente correspondiente a la práctica.

int strNumero; // Variable para almacenar el valor introducido por teclado.

String strBinario; // Variable para almacenar el valor en binario.

void setup(){

// Ya que el puerto serial utiliza los pines 0 y 1, para la representación a binario hacemos uso de los pines del 2 al 9 como salidas

// para conectar los leds correpondientes a cada bit.

for(int b=2; b<=9; b++) {

pinMode(b, OUTPUT); // Activamos los pines 0-9 como salida.

}

Serial.begin(9600); // Activamos el canal de comunicación serial con una velocidad de 9600 baudios.

while (!Serial) {

; // wait for serial port to connect. Needed for native USB port only

}

// Mensaje inicio.

Serial.println("Por favor, escriba un carácter.");

Serial.println();

}

void loop(){

if(Serial.available() > 0) { // ahora quedamos en espera de los digitos desde el puerto serial

strBinario = ""; // Reseteamos variable.

strNumero = Serial.read(); // Lectura del carácter introducido por serial.

// Imprime la información introducida por el usuario.

Serial.print("Carácter -> ");

Serial.write(strNumero);

Serial.println();

Serial.print("ASCII -> ");

Serial.print(strNumero);

// Verificamos que el numero este en el rango válido

if(strNumero > 0 && strNumero <= 255) {

// Hacemos un bucle activando cada led de acuerdo al bit correspondiente

for (int m=0; m<=7; m++) {

// Verificaremos cada led empezando en el menos significativo

// para esto usamos la funcion bitRead

// Si el bit esta ON, la salida sera HIGH, en caso contrario sera LOW

digitalWrite(2+m, bitRead(strNumero, m));

strBinario = (bitRead(strNumero,m)) + strBinario;

}

Serial.println();

Serial.println("Binario: " + strBinario);

}

// Solicita introducir nuevo caracter:

Serial.println();

Serial.println("Introduzca un nuevo carácter:");

Serial.println();

}

}

* **Enlaces de interés:**
  + Web oficial de la librería IRremote:

http://z3t0.github.io/Arduino-IRremote/

* + Web oficial Arduino. Manual función digitalWrite:

https://www.arduino.cc/en/Reference/DigitalWrite

* + Web oficial Arduino. Manual función bitRead:

https://www.arduino.cc/en/Reference/bitRead