

Arduino en el bachillerato tecnológico

Dr. Francisco Pérez García

Jefe del Departamento de Tecnología, Institut Pompeu Fabra, Generalitat de Catalunya

[fperez5@xtec.cat](mailto:fperez5@xtec.cat)

Palabras clave: Internet de las cosas, Internet de los objetos, Arduino, computación física

Resumen

Internet de los objetos, también conocido como internet de las cosas es uno de los ejes de internet del presente y futuro. Arduino es una plataforma de software de código abierto y hardware económico que permite el desarrollo de la imaginación de nuestros estudiantes. Mediante el uso de programación se controlaran diferentes sensores y actuadores, como LEDs, sensores LDR, sensores de diferentes gases o elementos piezoeléctricos. A modo de ejemplo se ha creado diversos prototipos: un aparato con un sistema de reconocimiento de voz que cuando se indica una pieza musical se generan las notas musicales y a la vez se genera el audio correspondiente o un sistema con sensores de gases contaminantes que envía datos a la nube.

Internet of things is one of the main topics in the present and future Internet. Arduino is a cheap open source hardware and software allowing to our students open source imagination. Programming could control different sensors and actuator, as LED, LDRs sensors, gas sensors, and piezo elements. For example it has been created several prototypes: a device with a voice recognition system able to recognize the name of a song creating musical notes and lighten up different LEDs corresponding to the musical notes or a system with air pollutants able to send their data to the cloud.

#### 1. Interés de internet de los objetos en educación

Actualmente hay un gran interés en la alfabetización digital del profesorado haciendo especial énfasis en la interactividad (PEREZ-GARCIA, 2010) y el desarrollo en los alumnos de las competencias informacionales (AREA et al., 2012), muchas veces incluyendo únicamente en competencias relacionadas con internet de los ordenadores, y en el presente artículo presentaré mi experiencia con alumnos de bachillerato intentando ir más allá, hacia el internet de las cosas y la computación física.

La computación física se utiliza con la intención de enseñar como los humanos se comunican con los ordenadores y pueden interactuar con ellos físicamente. En los últimos años estamos pasando del internet de los ordenadores al internet de los objetos, también llamada internet de las cosas.

El concepto de internet de los objetos (*internet of things*) fue usado por primera vez por Kevin Ashton en 1999 aplicado a la identificación de objetos por etiquetas de radiofrecuencia o RFID con la intención de ser identificados e inventariados por ordenadores (ASHTON, 2009) aunque Nikola Tesla en 1926 ya preveía que llevaríamos en el bolsillo toda la información de la Tierra

con todos los objetos interconectados como si fuese un gran cerebro, tal vez sea posible en un futuro próximo gracias a Internet de las cosas.

Internet de los Objetos permitirá que los objetos estén conectados en cualquier momento y en cualquier lugar, con cualquier cosa y cualquier persona idealmente utilizando cualquier ruta, red o servicio. Internet de los objetos es una nueva revolución de Internet. Los objetos se hacen reconocibles y obtienen inteligencia gracias al hecho de que se pueden comunicar información sobre sí mismos y pueden acceder a la información que ha sido recopilada por otras cosas. La iniciativa de internet de los objetos de la Unión Europea (*Internet of things Initiative*, UE) ha planteado para el futuro de la UE un plan de acción para la implementación de internet de las cosas en Europa con una inversión de miles de millones de euros dentro de la Agenda Digital de la UE 2020 y el 7º Programa Marco de Investigación de la UE.

La red actual de ordenadores interconectados evolucionará hacia una red de casi cualquier objeto interconectado desde libros a coches, desde aparatos electrónicos a alimentos. Así, algunos objetos podrían tener dirección IP, otros, sensores para obtener información ambiental. En el campo de alimentación, por ejemplo productos alimenticios que han de guardar la cadena de frío. En el campo de las ciudades inteligentes actuadores con válvulas de aire acondicionado que reaccionan con la presencia de las personas, sensores forestales para evitar la deforestación, coches interconectados para reducir el tráfico y la contaminación. Movilidad inteligente donde se indica los aparcamientos libres, posibilidades de transporte público y privado, pudiendo los despertadores suenan antes o después dependiendo del tráfico. En el campo de la salud, los contenedores de fármacos informaran si han sido tomados o no. En el campo del medio ambiente, las plantas se comunicarán cuando necesitan agua con el aspersor. A modo de ejemplo, actualmente es fácil construir con facilidad un “Botanical”, un sistema llamado Arduino que te informa gracias a un sensor de humedad cuando la planta necesita agua.

## 2. Arduino: la mejor opción para internet de las cosas

Arduino es una plataforma de hardware y software con una serie de características que lo hacen ideal para desarrollar internet de las cosas:

- a) Código abierto: tanto en software como en hardware
- b) Variedad de modelos: Arduino Uno o Leonardo para propósitos generales, Arduino Ethernet para conexiones via RJ45, Arduino Android para teléfonos móviles, Arduino Due o Arduino Mega para mayor número de entradas y salidas analógicas, Arduino Esplora para practicar con sensores ya incorporados, Arduino Lilypad para coser en la ropa, etc
- c) Facilidad y estabilidad de programación del programa Arduino
- d) Cantidad de recursos en internet con ejemplos
- e) Precio económico: rango entre 19.90€ del Arduino Leonardo a 59.99\$ del Arduino Esplora
- f) Disponibles gran cantidad de accesorios compatibles
- g) Disponible software específico para dibujo de circuitos como Fritzing

3. Prácticando con salidas digitales: LEDs parpadeando y código Morse.

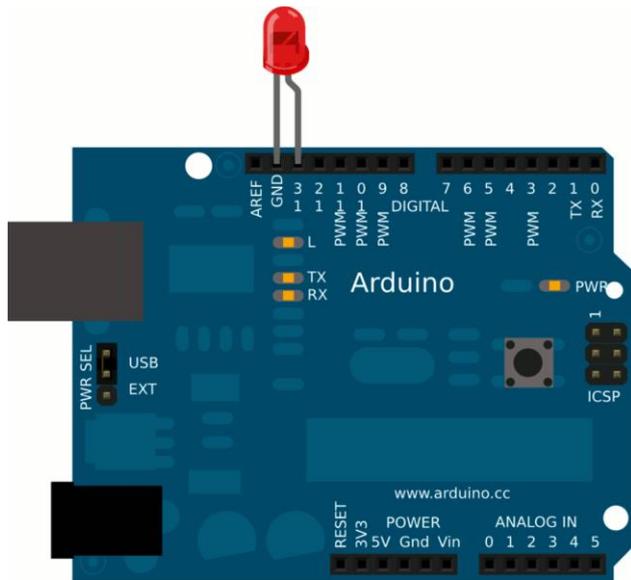


Figura 1. Montaje de Arduino Uno para parpadeo de un LED (código Morse)

Código fuente

```
/*
```

```
 Parpadeo: Enciende un LED un segundo y después lo apaga un segundo, repetidamente*/
```

```
 // Pin 13 es una salida digital que acepta directamente un LED
```

```
 // definimos una variable en el pin número 13 y la llamamos led:
```

```
 int led = 13;
```

```
 // la rutina setup se ejecuta al principio una sola vez o cuando se resetea:
```

```
 void setup() {
```

```
   // iniciar el pin digital como una salida.
```

```
   pinMode(led, OUTPUT);
```

```
 }
```

```
 // la rutina loop se repite sin parar siempre:
```

```
 void loop() {
```

```
   digitalWrite(led, HIGH); // enciende el LED (HIGH es el nivel de voltaje)
```

```
   delay(1000);           // espera un segundo
```

```
   digitalWrite(led, LOW); // apaga el LED haciendo el voltaje LOW
```

```
   delay(1000);          // espera un segundo
```

```
 }
```

Se pueden modificar los tiempos indicados en milisegundos para hacer señales largas o cortas de código morse o tratar de emplear una librería llamada “telegraph.h” que simplifica la tarea de enviar un mensaje.

Otras posibilidades son el cubo de LEDs

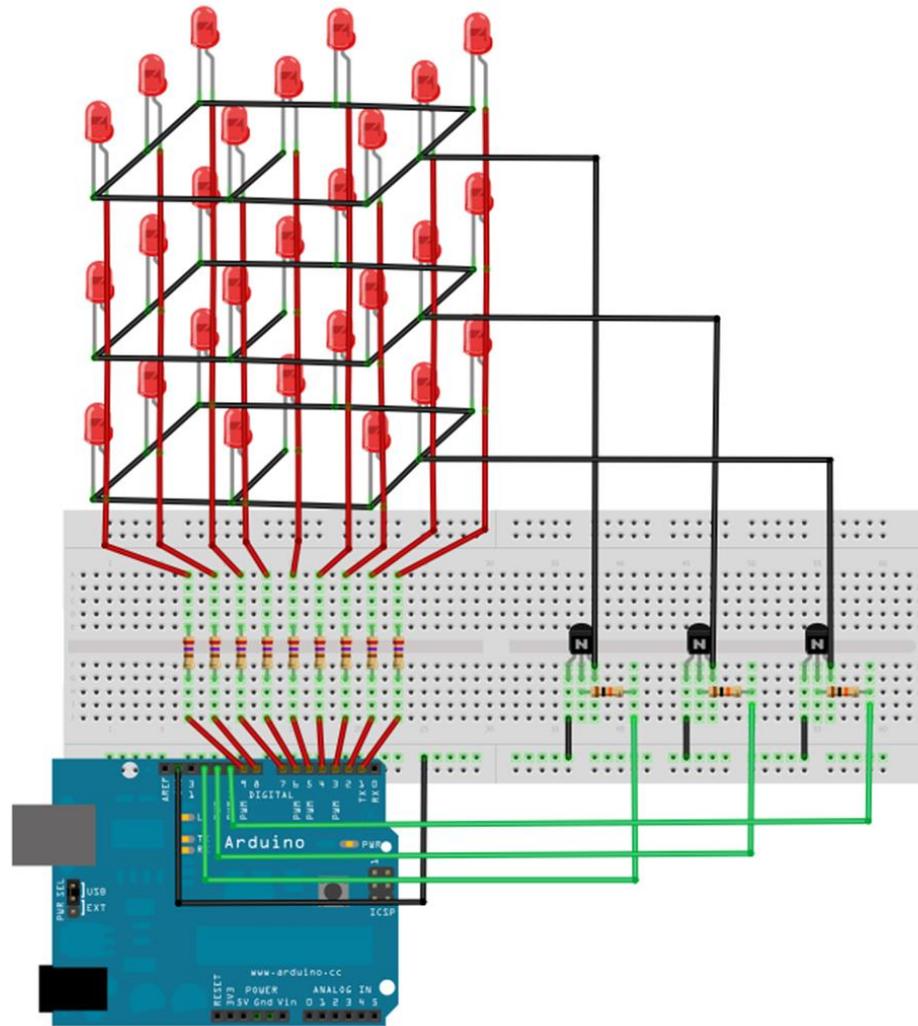


Figura 2. Cubo de LEDs. Mediante la programación se pudo controlar la secuencia de LEDs (ver videos con diferentes secuencias de leds en <http://blocs.xtec.cat/mecanica> y código fuente en la presentación de la página principal de <http://blocs.xtec.cat/tech>)

4. Practicando con entradas digitales: Interruptores y LED  
En esta práctica se trata de controlar un LED con un interruptor

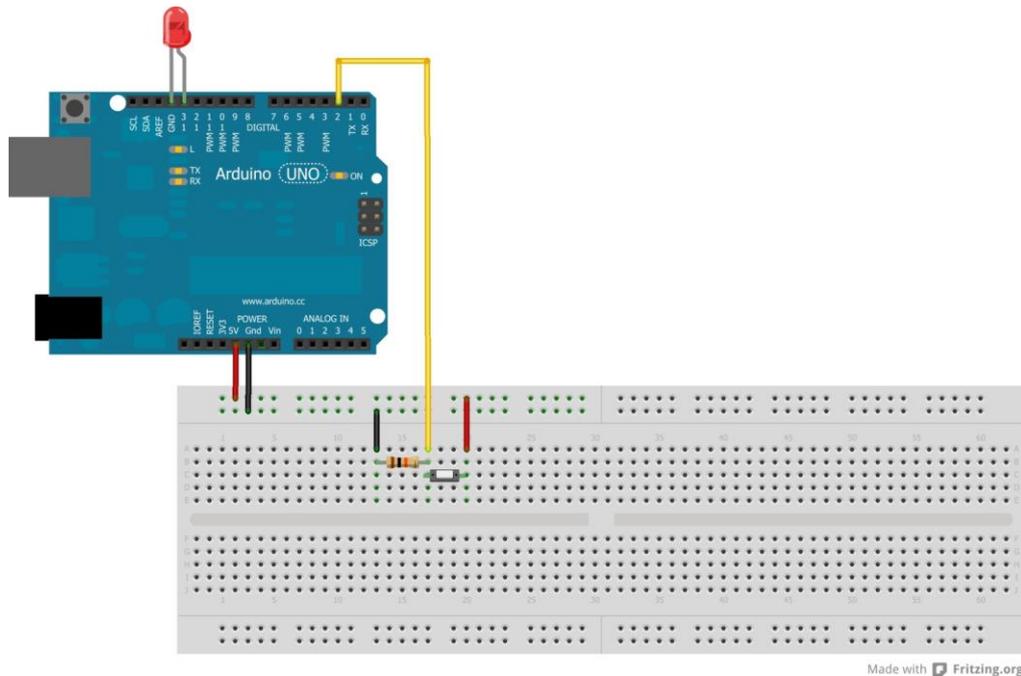


Figura 3. Interruptor como entrada digital que controla un LED.

```

/* Interruptor
Enciende y apaga un LED conectado al pin digital 13 cuando se presiona al botón
conectado al pin 2 */
// constantes que no cambian, se usan para definir los pins:
const int pulsadorPin = 2; // el número del pin del interruptor
const int ledPin = 13; // el número del pin del LED
// variables que cambian:
int pulsadorEstado = 0; // variable que lee el estado del interruptor
void setup() {
  // iniciar el LED como salida:
  pinMode(ledPin, OUTPUT);
  // iniciar el pulsador como entrada:
  pinMode(pulsadorPin, INPUT);
}
void loop(){
  // leer el estado del valor del pulsador:
  pulsadorEstado = digitalRead(pulsadorPin);
  // comprobar si el pulsador está presionado.
  if (pulsadorEstado == HIGH) {
    // enciende el LED si el pulsador está presionado:
    digitalWrite(ledPin, HIGH);
  }
  else {
    // en caso contrario apaga el LED:
    digitalWrite(ledPin, LOW);
  }
}

```

5. Practicando con entradas y salidas analógicas: Luz de un color del LED en función de la iluminación de un sensor LDR (video en <http://bloqs.xtec.cat/mecanica>)

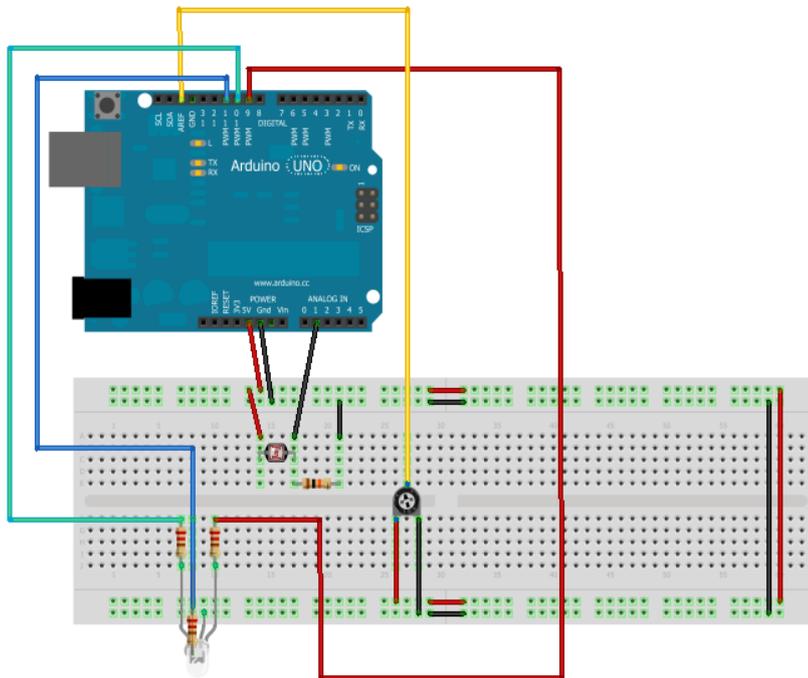


Figura 4. Mediante este circuito conseguimos que en función de la luz detectada por un sensor LDR cambie el color resultante en un LED-RGB

Este programa lo que hace es cambiar de color un led rgb mediante , según de la intensidad de luz solar que haya en el ambiente (los colores son programados en el programa).

*//Sensor LDR activa colores diferentes en LED RGB*

```
int valorLDR = 1;  
int ledRojo = 9;  
int ledVerde=10;  
int ledAzul=11;  
int pinLDR = 1;
```

*//Para el LED RGB se necesitan 3 salidas analógicas PWM diferentes que darán una señal de la intensidad de los 3 colores principales, el azul, el rojo y el verde.*

```
void setup()  
{
```

```
pinMode(ledRojo, OUTPUT);  
pinMode(ledVerde, OUTPUT);  
pinMode(ledAzul, OUTPUT);  
analogReference(EXTERNAL);
```

*//Asignacion de pines de los leds del circuito anterior, por los 3 colores, y dejamos otra vez el analogReference(EXTERNAL)*

*}*

*void loop()*

*{*

*valorLDR = analogRead(pinLDR);*

*if(valorLDR >= 1023)*

*{*

*digitalWrite(ledRojo, 128);*

*digitalWrite(ledVerde, 0);*

*digitalWrite(ledAzul, 0);*

*//Los valores de entrada analógica son de 10 bits, valores de 0 a 1023 leídos por el sensor LDR. En función de estos valores daremos un valor de salida analógica, que son de 8 bits, valores de 0 a 255 en cada uno de los tres pines que dan combinados cualquier color.*

*}*

*else if((valorLDR >= 959) & (valorLDR < 1023))*

*{*

*digitalWrite(ledRojo, 255);*

*digitalWrite(ledVerde, 0);*

*digitalWrite(ledAzul, 0);*

*}*

*else if((valorLDR >= 895) & (valorLDR < 959))*

*{*

*digitalWrite(ledRojo, 255);*

*digitalWrite(ledVerde, 128);*

*digitalWrite(ledAzul, 0);*

*}*

*else if((valorLDR >= 831) & (valorLDR < 895))*

*{*

*digitalWrite(ledRojo, 255);*

*digitalWrite(ledVerde, 255);*

*digitalWrite(ledAzul, 0);*

*}*

*else if((valorLDR >= 767) & (valorLDR < 831))*

*{*

*digitalWrite(ledRojo, 255);*

*digitalWrite(ledVerde, 255);*

*digitalWrite(ledAzul, 128);*

*}*

*else if((valorLDR >= 703) & (valorLDR < 767))*

*{*

*digitalWrite(ledRojo, 128);*

*digitalWrite(ledVerde, 255);*

*digitalWrite(ledAzul, 255);*

*}*

```

else if((valorLDR >= 639) & (valorLDR < 703))
{
digitalWrite(ledRojo, 128);
digitalWrite(ledVerde, 128);
digitalWrite(ledAzul, 255);
}
else if((valorLDR >= 575) & (valorLDR < 639))
{
digitalWrite(ledRojo, 0);
digitalWrite(ledVerde, 128);
digitalWrite(ledAzul, 255);
}
else if((valorLDR >= 511) & (valorLDR < 575))
{
digitalWrite(ledRojo, 0);
digitalWrite(ledVerde, 0);
digitalWrite(ledAzul, 255);
}
else if((valorLDR >= 447) & (valorLDR < 511))
{
digitalWrite(ledRojo, 0);
digitalWrite(ledVerde, 0);
digitalWrite(ledAzul, 128);
}
else if((valorLDR >= 383) & (valorLDR < 447))
{
digitalWrite(ledRojo, 0);
digitalWrite(ledVerde, 128);
digitalWrite(ledAzul, 0);
}
else if((valorLDR >= 319) & (valorLDR < 383))
{
digitalWrite(ledRojo, 0);
digitalWrite(ledVerde, 255);
digitalWrite(ledAzul, 0);
}
else if((valorLDR >= 255) & (valorLDR < 319))
{
digitalWrite(ledRojo, 128);
digitalWrite(ledVerde, 255);
digitalWrite(ledAzul, 0);
}
else if((valorLDR >= 191) & (valorLDR < 255))
{
digitalWrite(ledRojo, 0);
digitalWrite(ledVerde, 255);
digitalWrite(ledAzul, 128);
}
else if((valorLDR >= 127) & (valorLDR < 191))
{

```

```

digitalWrite(ledRojo, 128);
digitalWrite(ledVerde, 255);
digitalWrite(ledAzul, 128);
}
else if((valorLDR >= 63) & (valorLDR < 127))
{
digitalWrite(ledRojo, 128);
digitalWrite(ledVerde, 128);
digitalWrite(ledAzul, 128);
}
else if((valorLDR >=0) & (valorLDR < 63))
{
digitalWrite(ledRojo, 55);
digitalWrite(ledVerde, 55);
digitalWrite(ledAzul, 55);
}
else
{
digitalWrite(ledRojo, 0);
digitalWrite(ledVerde, 0);
digitalWrite(ledAzul, 0);
}

}
void color(int rojo, int verde, int azul)

{
analogWrite(ledRojo, 255-rojo);
analogWrite(ledVerde, 255-verde);
analogWrite(ledAzul, 255-azul);

//Aquí al final ponemos la escritura del PWM de cada color, para que se puedan
completar las ordenes y colores que le damos al Arduino.
}

```

## 6. Sensores de polución y presentación de datos en la nube

A modo de ejemplo se introduce el código correspondiente al sensor de dióxido de carbono MG811 (códigos similares para sensores de otros gases contaminantes de la serie MQ de Hanwei Electronics). Videos en <http://blocs.xtec.cat/tech>

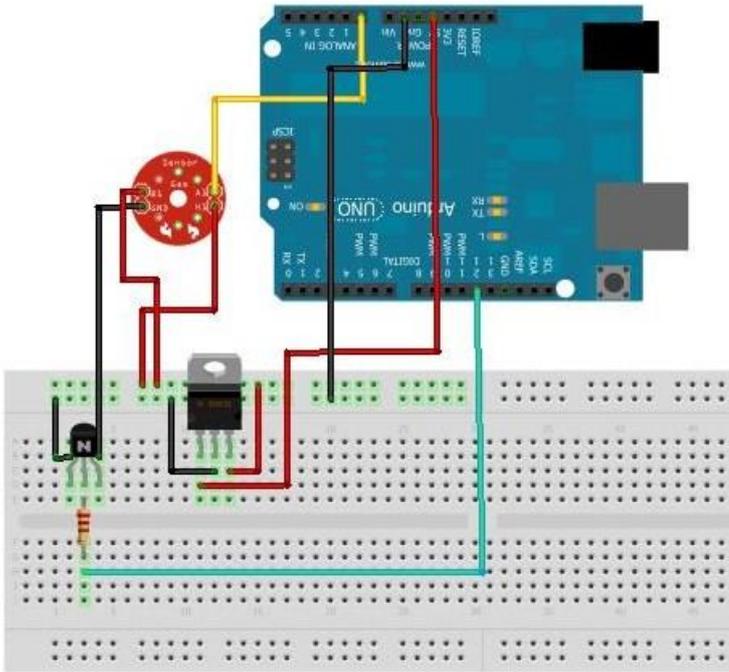


Figura 5. Estructura básica del circuito con un sensor tipo MQ (Hanwei Electronics)

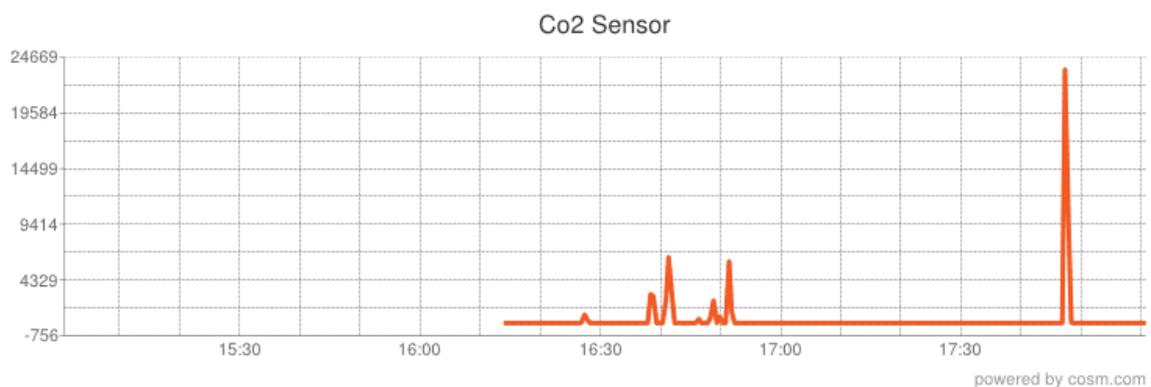


Figura 6. Resultados online de los niveles de dióxido de carbono de nuestro circuito, los picos pequeños son debidos a la respiración sobre el sensor y el pico mayor a un mechero. Los niveles basales son de 400ppm.

//Código para sensor dióxido de carbono conectado a [www.cosm.com](http://www.cosm.com)

```
#include <Ethernet.h>
```

```
#include <string.h>
```

```
#include <SPI.h>
```

```
#define SHARE_FEED_ID    101673
```

```
#define REMOTE_FEED_ID    256
```

```

#define REMOTE_FEED_DATASTREAMS 4

#define UPDATE_INTERVAL 5000

#define RESET_INTERVAL 5000

#define DC_GAIN (8.5)

#define READ_SAMPLE_INTERVAL (50)

#define READ_SAMPLE_TIMES (5)

#define ZERO_POINT_VOLTAGE (0.220)

#define REACTION_VOLTGAE (0.020)

#define PACHUBE_API_KEY
"XwpsubGiAzgs0G8zIz4Rq4vvULKSAKxPZitxaUM0YmY4OD0g"

byte mac[] = { 0x90, 0xA2, 0xDA, 0x0D, 0x00, 0xFB };

byte ip[] = { 192, 168, 0, 244 };

byte gateway[] = { 192, 168, 0, 194 };

byte subnet[] = { 255, 255, 255, 0 };

byte remoteServer[] = { 216,52,233,122 };

float remoteSensor[REMOTE_FEED_DATASTREAMS];

int MG_PIN = 3;

int BOOL_PIN = 4;

int percentage;

float CO2Curve[3] = {2.602,ZERO_POINT_VOLTAGE,(REACTION_VOLTGAE/(2.602-3))};

void setup()

{

  Serial.begin(9600);

  pinMode(BOOL_PIN, INPUT);

  digitalWrite(BOOL_PIN, HIGH);

  Serial.print("MG-811 Demonstration\n");

  setupEthernet();

}

```

```

void loop()
{
    float volts;

    volts = MGRead(MG_PIN);

    Serial.print( "SEN-00007:" );

    Serial.print(volts);

    Serial.print( "V      " );

    percentage = MGGetPercentage(volts,CO2Curve);

    Serial.print("CO2:");

    Serial.print(percentage);

    Serial.print( "ppm" );

    Serial.print("\n");

    if (digitalRead(BOOL_PIN) ){

        Serial.print( "=====BOOL is HIGH=====" );

    } else {

        Serial.print( "=====BOOL is LOW=====" );

    }

    Serial.print("\n");

    pachube_in_out();

    delay (5000);

}

float MGRead(int mg_pin)

{

    int i;

    float v=0;

    for (i=0;i<READ_SAMPLE_TIMES;i++) {

        v += analogRead(mg_pin);

```

```

    delay(READ_SAMPLE_INTERVAL);
}
v = (v/READ_SAMPLE_TIMES) *5/1024 ;
return v;
}
int MGGetPercentage(float volts, float *pcurve)
{
if ((volts/DC_GAIN )>=ZERO_POINT_VOLTAGE) {
    return 400;
} else {
    return pow(10, ((volts/DC_GAIN)-pcurve[1])/pcurve[2]+pcurve[0]); } }

```

7. Creación de un prototipo con un sistema de reconocimiento de voz y un elemento piezoeléctrico capaz de producir notas musicales y al mismo tiempo encender led correspondientes a las diferentes notas musicales. Video disponible en <http://bloqs.xtec.cat/tech>

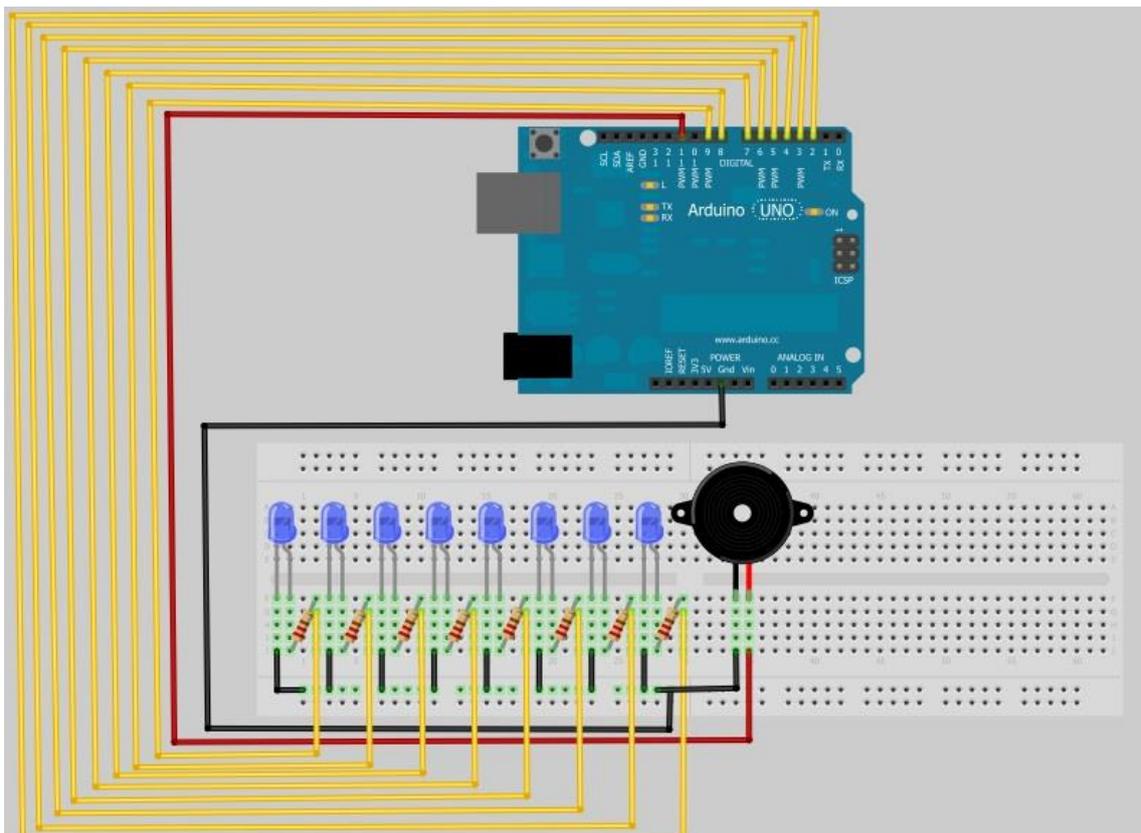


Figura 7. Circuito una vez reconocido el audio correspondiente a una canción mediante un micrófono, reproduce las notas musicales un elemento piezoeléctrico mientras a la vez se encienden los LEDs correspondientes a diferentes notas musicales

//Código del proyecto de Audio

```
#if defined(ARDUINO) && ARDUINO >= 100
```

```
#include "Arduino.h"
```

```
#include "SoftwareSerial.h"
```

```
SoftwareSerial port(12,13);
```

```
#else
```

```
#include "WProgram.h"
```

```
#include "NewSoftSerial.h"
```

```
NewSoftSerial port(12,13);
```

```
#endif
```

```
#include "EasyVR.h"
```

```
EasyVR easyvr(port);
```

```
int speakerPin = 11;
```

```
int length = 100; /* number of notes */
```

```
char notes_NOI_DE_LA_MARE[] = "mfssISDSlls mfssISDSls mfssfmrdrmd  
mfssfmrdrd"; /* orden de las notas representadas por sus iniciales, un espacio no hay musica  
*/
```

```
int beats_NOI_DE_LA_MARE[] = { 2, 1, 1, 2, 1, 1, 2, 1, 1, 3, 1, 1, 2, 1, 1, 2, 1, 1, 2, 1,  
1, 3, 1, 2, 1, 1, 2, 1, 1, 2, 1, 1, 3, 1, 1, 2, 1, 1, 2, 1, 1, 2, 1, 1, 3 };
```

```
char notes_HAPPY_BIRTHDAY[] = "sslsDS sslsRD ssMSlls RRDsls"; /* orden de las  
notas representadas por sus iniciales, un espacio no hay musica */
```

```
int beats_HAPPY_BIRTHDAY[] = { 2, 1, 2, 2, 2, 3, 1, 2, 1, 2, 2, 2, 3, 1, 2, 1, 2, 2, 2, 3,  
1, 2, 1, 2, 2, 2, 3 };
```

```
char notes_HIMNE_ALEGRIA[] = "SDRRDSlsslSSI SDRRDSlsslSlS lSslSDSslSDSslSll  
SDRRDSlsslSlS"; /* orden de las notas representadas por sus iniciales, un espacio no hay musica  
*/
```

```
int beats_HIMNE_ALEGRIA[] = { 2, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 2, 2, 1, 2, 1, 1, 1, 1, 1, 1,
1, 1, 1, 1, 2, 2, 1, 2, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 2, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 2, 2 };
```

```
int tempo = 200;
```

```
void playTone (int tone, int duration) {
```

```
for (long i = 0; i < duration * 1000L; i += tone * 2) {
```

```
digitalWrite(speakerPin, HIGH);
```

```
delayMicroseconds(tone);
```

```
digitalWrite(speakerPin, LOW);
```

```
delayMicroseconds(tone);
```

```
}
```

```
}
```

```
void playNote (char note, int duration) {
```

```
char names[] = {'d','r','m','f','s','l','S','D', 'R', 'M'};
```

```
int tones[] = {1915, 1700, 1519, 1432, 1275, 1136, 1014, 956, 900, 875 };
```

```
char names2[] = {'R','M'};
```

```
int tones2[] = {900,875};
```

```
for (int i = 0; i < 8; i++) {
```

```
if (names[i] == note) {
```

```
playTone(tones[i], duration);
```

```
}
```

```
}
```

```
}
```

```
//Grupos y comandos
```

```
enum Groups
```

```
{
```

```
GROUP_0 = 0,
```

```
GROUP_1 = 1,
```

```
};
```

```

enum Group0
{
    G0_ARDUINO = 0,
};

enum Group1
{
    G1_NOI_DE_LA_MARE = 0,
    G1_HAPPY_BIRTHDAY = 1,
    G1_HIMNE_ALEGRIA = 2,
};

EasyVRBridge bridge;

int8_t group, idx;

void setup()
{
    pinMode(speakerPin, OUTPUT);

    pinMode(0, OUTPUT);

    pinMode(1, OUTPUT);

    pinMode(2, OUTPUT);

    pinMode(3, OUTPUT);

    pinMode(4, OUTPUT);

    pinMode(5, OUTPUT);

    pinMode(6, OUTPUT);

    pinMode(7, OUTPUT);

    pinMode(8, OUTPUT);

    pinMode(9, OUTPUT);

    if (bridge.check())
    {

```

```

cli();

bridge.loop(0, 1, 12, 13);

}

// run normally

Serial.begin(9600);

port.begin(9600);

if (!easyvr.detect())

{

    Serial.println("EasyVR not detected!");

    for (;;);

}

easyvr.setPinOutput(EasyVR::IO1, LOW);

Serial.println(" EasyVR detected!");

easyvr.setTimeout(5);

easyvr.setLanguage(3);

group = EasyVR::TRIGGER; /*empezar un grupo para personalizar*/

}

void action();

void loop()

{

    Serial.print("Say a command in Group");

    Serial.println(group);

    easyvr.recognizeCommand(group);

    do

    {

        /* esperar al comando hablado */

    }

}

```

```

while (!easyvr.hasFinished());

idx = easyvr.getCommand();

if (idx >= 0)
{
    /* mostrar mensaje */

    uint8_t train = 0;

    char name[32];

    Serial.print("Command: ");

    Serial.print(idx);

    if (easyvr.dumpCommand(group, idx, name, train))
    {
        Serial.print(" = ");

        Serial.println(name);
    }

    else

        Serial.println();

    easyvr.playSound(0, EasyVR::VOL_FULL);

    /* realiza una acción */

    action();
}

else /* errores or fuera de tiempo límite */
{
    if (easyvr.isTimeout())

        Serial.println("Timed out, try again...");

    int16_t err = easyvr.getError();

    if (err >= 0)
    {

```

```

    Serial.print("Error ");
    Serial.println(err, HEX);
}
group = GROUP_0;
}
}
void action()
{
    switch (group)
    {
    case GROUP_0:
        switch (idx)
        {
        case GO_ARDUINO:
            /* action code HERE */
            group = GROUP_1;
            break;
        }
        break;
    case GROUP_1:
        switch (idx)
        {
        case G1_NOI_DE_LA_MARE:
            group = GROUP_0;
            for (int i = 0; i < length; i++) {
            if (notes_NOI_DE_LA_MARE[i] == 'd') {
                digitalWrite(2, HIGH);

```

```
delay(100);  
digitalWrite(2, LOW);  
delay(1);  
}  
if (notes_NOI_DE_LA_MARE[i] == 'r') {  
    digitalWrite(3, HIGH);  
    delay(100);  
    digitalWrite(3, LOW);  
    delay(1);  
}  
if (notes_NOI_DE_LA_MARE[i] == 'm') {  
    digitalWrite(4, HIGH);  
    delay(100);  
    digitalWrite(4, LOW);  
    delay(1);  
}  
if (notes_NOI_DE_LA_MARE[i] == 'f') {  
    digitalWrite(5, HIGH);  
    delay(100);  
    digitalWrite(5, LOW);  
    delay(1);  
}  
if (notes_NOI_DE_LA_MARE[i] == 's') {  
    digitalWrite(6, HIGH);  
    delay(100);  
    digitalWrite(6, LOW);  
    delay(1);  
}
```

```

}

if (notes_NOI_DE_LA_MARE[i] == 'l') {
    digitalWrite(7, HIGH);
    delay(100);
    digitalWrite(7, LOW);
    delay(1);
}

if (notes_NOI_DE_LA_MARE[i] == 'S') {
    digitalWrite(8, HIGH);
    delay(100);
    digitalWrite(8, LOW);
    delay(1);
}

if (notes_NOI_DE_LA_MARE[i] == 'D') {
    digitalWrite(9, HIGH);
    delay(100);
    digitalWrite(9, LOW);
    delay(1);
}

if (notes_NOI_DE_LA_MARE[i] == ' ') {
    delay(beats_NOI_DE_LA_MARE[i] * tempo); /* rest */
} else {
    playNote(notes_NOI_DE_LA_MARE[i], beats_NOI_DE_LA_MARE[i] * (tempo * 2));
}

/* pause between notes */
delay(tempo / 2);
}

```



```

}

if (notes_HAPPY_BIRTHDAY[i] == 's') {
    digitalWrite(4, HIGH);
    delay(100);
    digitalWrite(4, LOW);
    delay(1);
}

if (notes_HAPPY_BIRTHDAY[i] == 'l') {
    digitalWrite(5, HIGH);
    delay(100);
    digitalWrite(5, LOW);
    delay(1);
}

if (notes_HAPPY_BIRTHDAY[i] == 'S') {
    digitalWrite(6, HIGH);
    delay(100);
    digitalWrite(6, LOW);
    delay(1);
}

if (notes_HAPPY_BIRTHDAY[i] == 'D') {
    digitalWrite(7, HIGH);
    delay(100);
    digitalWrite(7, LOW);
    delay(1);
}

if (notes_HAPPY_BIRTHDAY[i] == ' ') {
    delay(beats_HAPPY_BIRTHDAY[i] * tempo); /* rest */
}

```

```

} else {

    playNote(notes_HAPPY_BIRTHDAY[i], beats_HAPPY_BIRTHDAY[i] * tempo);

}

    /* pause between notes */

    delay(tempo / 2);

}

    break;

    case G1_HIMNE_ALEGRIA:

        group = GROUP_0;

        for (int i = 0; i < length; i++) {

if (notes_HIMNE_ALEGRIA[i] == 'd') {

    digitalWrite(0, HIGH);

    delay(100);

    digitalWrite(0, LOW);

    delay(1);

}

if (notes_HIMNE_ALEGRIA[i] == 'r') {

    digitalWrite(1, HIGH);

    delay(100);

    digitalWrite(1, LOW);

    delay(1);

}

if (notes_HIMNE_ALEGRIA[i] == 'm') {

    digitalWrite(2, HIGH);

    delay(100);

    digitalWrite(2, LOW);

    delay(1);

}

```

```
}  
  
if (notes_HIMNE_ALEGRIA[i] == 'f') {  
    digitalWrite(3, HIGH);  
    delay(100);  
    digitalWrite(3, LOW);  
    delay(1);  
}  
  
if (notes_HIMNE_ALEGRIA[i] == 's') {  
    digitalWrite(4, HIGH);  
    delay(100);  
    digitalWrite(4, LOW);  
    delay(1);  
}  
  
if (notes_HIMNE_ALEGRIA[i] == 'l') {  
    digitalWrite(5, HIGH);  
    delay(100);  
    digitalWrite(5, LOW);  
    delay(1);  
}  
  
if (notes_HIMNE_ALEGRIA[i] == 'S') {  
    digitalWrite(6, HIGH);  
    delay(100);  
    digitalWrite(6, LOW);  
    delay(1);  
}  
  
if (notes_HIMNE_ALEGRIA[i] == 'D') {  
    digitalWrite(7, HIGH);
```

```

    delay(100);

    digitalWrite(7, LOW);

    delay(1);
}

if (notes_HIMNE_ALEGRIA[i] == ' ') {

    delay(beats_HIMNE_ALEGRIA[i] * tempo); /* rest */

} else {

    playNote(notes_HIMNE_ALEGRIA[i], beats_HIMNE_ALEGRIA[i] * tempo);

}

/* pausa entre notas */

delay(tempo / 2);

}

    break;

}

    break;

}

}

```

## 8. Experiencias con los alumnos

Los alumnos que han realizado todos los experimentos pertenecen a asignaturas del Bachillerato tecnológico, en grupos de 16 alumnos durante los cursos 2011-2012 y 2012-2013 durante 4 horas semanales todo el curso.

El primer trimestre consistió en programación en C, el segundo trimestre practicas sencillas con Arduino, el tercer trimestre, proyectos más complejos con Arduino.

Los alumnos grabaron en video con sus móviles los experimentos, dibujaron los circuitos y explicaron los códigos en sus blogs o páginas web.

La comprensión de un lenguaje de programación es una capacidad clave que debería estar incluida en el currículum del bachillerato, ya que es una habilidad fundamental en las ingenierías, que además ayuda a desarrollar el pensamiento abstracto, el pensamiento crítico y la creatividad.

Los alumnos reconocen las posibilidades de cualquier tipo de entradas y salidas digitales solo limitadas por la imaginación de los alumnos ya que el valor económico de sensores y actuadores no es elevado. El precio de todos los componentes electrónicos para trabajar con los alumnos ha sido inferior a 1000 euros obtenidos del proyecto PILE indicado en los agradecimientos. A pesar de ello con un presupuesto mucho menor, unos 300 euros, se pueden hacer la mayor parte de prácticas.

De cara al curso 2013-2014 en función de los resultados obtenidos se pretende potenciar aún más la creatividad desde el principio, evitando la parte inicial de programación en C, ya que pueden aprender directamente el lenguaje practicando con Arduino desde el primer día.

#### 9. Agradecimientos

A la Generalitat de Catalunya por el proyecto PILE *Contemporary Sciences and Research* al Instituto Pompeu Fabra de Martorell, de enseñanza en inglés de materias no lingüísticas, en nuestro caso del Bachillerato tecnológico y científico.

#### 10. Bibliografía

AREA, Manuel; GUTIÉRREZ, Alfonso; VIDAL, Fernando (2012). Alfabetización digital y competencias informacionales. pp. 225. Madrid: Fundación Telefónica / Ed. Ariel. ISBN 978-84-08-00823-1. En Revista Didáctica, Innovación y Multimedia, núm. 23 <http://www.pangea.org/dim/revista23.htm>

ASHTON, Kevin. That "Internet of Things" Thing RFID Journal 22 de julio de 2009 disponible en <http://www.rfidjournal.com/article/view/4986>

PÉREZ GARCÍA, Francisco (2010). Alfabetización digital del profesorado: herramientas educativas Interactivas. En Revista Didáctica, Innovación y Multimedia, núm. 16 <http://www.pangea.org/dim/revista16>

Internet of Things Initiative, Unión Europea. <http://www.iot-i.eu>

Página web oficial de Arduino con ejemplos [www.arduino.cc](http://www.arduino.cc)

Algunas páginas del autor con información sobre Arduino: <http://blocs.xtec.cat/mecanica>, <http://blocs.xtec.cat/pile>, <http://blocs.xtec.cat/tech>