

Formación en Tecnología, Programación y Robótica

Itinerario Educación
Secundaria
Comunidad Autónoma
de Madrid
Actividad práctica



Actividad con Arduino

1_ Nombre, apellidos y correo electrónico de EducaMadrid del participante.

Julián Santos Alcón

julian.santos@educa.madrid.org

2_ Título de la unidad didáctica

Encendido y apagado de ledes con un mando a distancia de infrarrojos.

3_ Curso

TPR - 3ESO

TPR Proyectos tecnológicos - 4ESO

En función del nivel de los grupos, puede ser realizado en los dos niveles.

4_ Descripción del proyecto

El proyecto consiste en desarrollar un sistema de encendido/apagado de 10 ledes utilizando un mando a distancia.

Los alumnos deben configurar un receptor de infrarrojos para recibir la señal se un mando a distancia. Se deben poder diferenciar 12 tipos de señales diferentes que corresponderán a:

- Botones 1 - 9 del mando a distancia: permitirán encender/apagar los ledes numerados del 1 al 9. Sólo un led de entre los nueve primeros puede estar encendido.
- Botón ON/OFF: permitirá apagar/encender el sistema.
- Botos CH+ CH- hará que se encienda el led anterior/siguiente al encendido.

El led de encendido/apagado tendrá un color diferente y se situará en la última posición.

Los pasos del proyecto serán:

- Montaje de un receptor de infrarrojos y desarrollo del sketch para el envío a través del monitor serie de las frecuencias de los botones necesarios del mando a distancia.
- Asignación en el código de funciones asociadas a cada uno de las frecuencias que deben ser tenidas en cuenta. Se recomienda el uso de la estructura switch().
- Montaje de los ledes y prueba de funcionamiento.
- Una vez el proyecto cumple los requisitos inicialmente planteados, se pueden añadir nuevas funcionalidades (botones que inicien secuencias de led, parpadeo de los ledes con pulsación continuada...) o modificar el código con técnicas más avanzadas (por ejemplo, Port Manipulation utilizando PORTB y PORTD para escribir varios pines de forma simultánea) o el uso de arrays.
- Presentación de la solución realizada a los compañeros.

5_ Competencias que se trabajan

- Digital
- Aprender a aprender
- Matemática y básicas en ciencia y tecnología
- Cívicas y sociales
- Sentido de iniciativa y espíritu emprendedor
- Competencias sociales y cívicas

6_ Objetivos

- Desarrollar programas en un lenguaje de programación textual
- Analizar las características básicas de funcionamiento de diferentes componentes electrónicos activos
- Describir las características de los sensores de infrarrojos, identificando sus principios de funcionamiento físico
- Utiliza con precisión el entorno de programación de un sistema electrónico.
- Desarrollar, en colaboración con sus compañeros de equipo, un proyecto de sistema robótico

7_ Contenidos

- Formulación de un proyecto tecnológico. Identificación del problema. Análisis de su naturaleza.
- Innovación y creatividad para la búsqueda de soluciones tecnológicas.
- Diseño, montaje y medida de los circuitos electrónicos de un proyecto tecnológico.
- Programación de los circuitos electrónicos de un proyecto tecnológico.
- Documentación de un prototipo desarrollado a través de un proyecto tecnológico.

8_ Criterios de evaluación

- Montaje adecuado del receptor IR y calibración para el mando a distancia utilizado en el proyecto (20%)
- Implementación correcta del programa, que soluciona el problema planteado en la descripción del proyecto (30%)
- Código estructurado, se hace uso de funciones y de estructuras de código más allá de las simples (15%)
- El montaje de los componentes en la placa es ordenado, se respetan los colores asignados para los cables, especialmente GND y Vcc, así como cables con análoga funcionalidad (salidas digitales) y es fácil de interpretar (15%)
- Correcto trabajo en equipo en el taller, distribución de tareas y colaboración de todos los miembros en las diferentes fases del proyecto (20%)
- Los alumnos pueden añadir elementos adicionales que mejoren su funcionamiento o impliquen un conocimiento más avanzado de robótica y control (hasta un 20% adicional)

9 Código del programa Arduino

SKETCH 1 - CODIGO PARA LA IDENTIFICACION DE FRECUENCIAS

```
/*
 * PROYECTO FINAL - VT2 ROBOTICA AVANZADA
 * DETECCION DE FRECUENCIAS
 *Codigo del receptor de infrarrojos tomado de:
 * http://blog.whatgeek.com.pt/arduino/keyes-ir-remote-sensor/
 * https://learn.sparkfun.com/tutorials/
 * ir-communication/receiving-ir-example
 * y modificado apropiadamente
 *
 * Información sobre los métodos de la clase IRrecv en:
 * https://www.pjrc.com/teensy/td_libs_IRremote.html
 * JSA 2018 @ ScholarTIC - creandocodigo_
 * Licencia Creative Commons (by - nc - sa)
 */

// Biblioteca para el uso de IR Remoto
#include <IRremote.h>

// Pin de señal del receptor IR
int receptor = 2;
// Crea un objeto tipo receptor
// Asociado al pin definido arriba
IRrecv irrecv(receptor);
// Variable para frecuencia recibida
decode_results resultado;

void setup() {
  // Inicializa puerto serie para
  // comunicación con Processing
  Serial.begin(9600);
  // Inicializa el receptor
  irrecv.enableIRIn();
  // Hara parpadear el led integrado al
  // recibir señal el receptor IR
  irrecv.blink13(true);
}

void loop() {
  // Si se reciben datos en el receptor
  if (irrecv.decode(&resultado))
  {
    // Para cada señal recibida mostramos su frecuencia
    Serial.println(resultado.value,HEX);
    // Receptor IR continua a la espera de señal IR
    irrecv.resume();
  }
}
```

SKETCH 2 - CODIGO PARA LA IMPLEMTACION CONOCIDAS LAS FRECUENCIAS

```
/*
 * PROYECTO FINAL - VT2 ROBOTICA AVANZADA
 *
 * Codigo del receptor de infrarrojos tomado de:
 * http://blog.whatgeek.com.pt/arduino/keys-ir-remote-sensor/
 * https://learn.sparkfun.com/tutorials/
 * ir-communication/receiving-ir-example
 * y modificado apropiadamente
 *
 * Información sobre los métodos de la clase IRrecv en:
 * https://www.pjrc.com/teensy/td_libs_IRremote.html
 * JSA 2018 @ ScolaTIC - creandocodigo_
 * Licencia Creative Commons (by - nc - sa)
 */

// Biblioteca para el uso de IR Remoto
#include <IRremote.h>

// Definición de las frecuencias del mando a distancia.
// Dependen del mando utilizado.
// Deben ser medidas y modificadas al cambiar de mando.
const long uno = 0xE0E020DF;
const long dos = 0xE0E0A05F;
const long tres = 0xE0E0609F;
const long cuatro = 0xE0E010EF;
const long cinco = 0xE0E0906F;
const long seis = 0xE0E050AF;
const long siete = 0xE0E030CF;
const long ocho = 0xE0E0B04F;
const long nueve = 0xE0E0708F;
const long onoff = 0xE0E040BF;
const long arriba = 0xE0E048B7;
const long abajo = 0xE0E008F7;

// Asignacion de num. de pin
int led_onoff = 12;
int led_1 = 11;
int led_2 = 10;
int led_3 = 9;
int led_4 = 8;
int led_5 = 7;
int led_6 = 6;
int led_7 = 5;
int led_8 = 4;
int led_9 = 3;
// Pin de señal del receptor IR
int receptor = 2;

// Variable para encendido/apagado
// Inicialmente apagado
bool encendido = false;

// Variable que indica ultimo led encendido
// Para utilizar arriba y abajo
int ultimo = 0;

// Crea un objeto tipo receptor
```

```
// Asociado al pin definido arriba
IRrecv irrecv(receptor);

// Variable para frecuencia recibida
decode_results resultado;

// Funcion que apaga todos los ledes
// y enciende unicamente el pasado argumento
// El led on/off funciona de forma independiente
void unico_led(int led)
{
  for (int i = 3; i < 12; i = i + 1)
  {
    digitalWrite(i,LOW);
  }
  digitalWrite(led,HIGH);
}

// Apaga todos los ledes relacionados con numeros
void apagar_todo()
{
  for (int i = 3; i < 12; i = i + 1)
  {
    digitalWrite(i,LOW);
  }
}

void setup() {
  // Inicializ. pines GPIO 3 - 12 salida
  // Inicialmente apagados (LOW)
  for (int i = 3; i < 13; i = i + 1)
  {
    pinMode(i, OUTPUT);
    digitalWrite(i,LOW);
  }
  digitalWrite (12,HIGH);
  // Inicializa puerto serie para
  // comunicación con Processing
  Serial.begin(9600);
  // Inicializa el receptor
  irrecv.enableIRIn();
  // Hara parpadear el led integrado al
  // recibir señal el receptor IR
  irrecv.blink13(true);
}

void loop() {
  // Si se reciben datos en el receptor
  if (irrecv.decode(&resultado))
  {
    Serial.println(resultado.value,HEX);
    // Asignamos a cada tecla una funcion segun la descripcion
    // OJO, cada mando funciona con valores diferentes, por lo
    // no funcionara con cualquier mando a distancia
    // Redefinir frecuencias segun control remoto

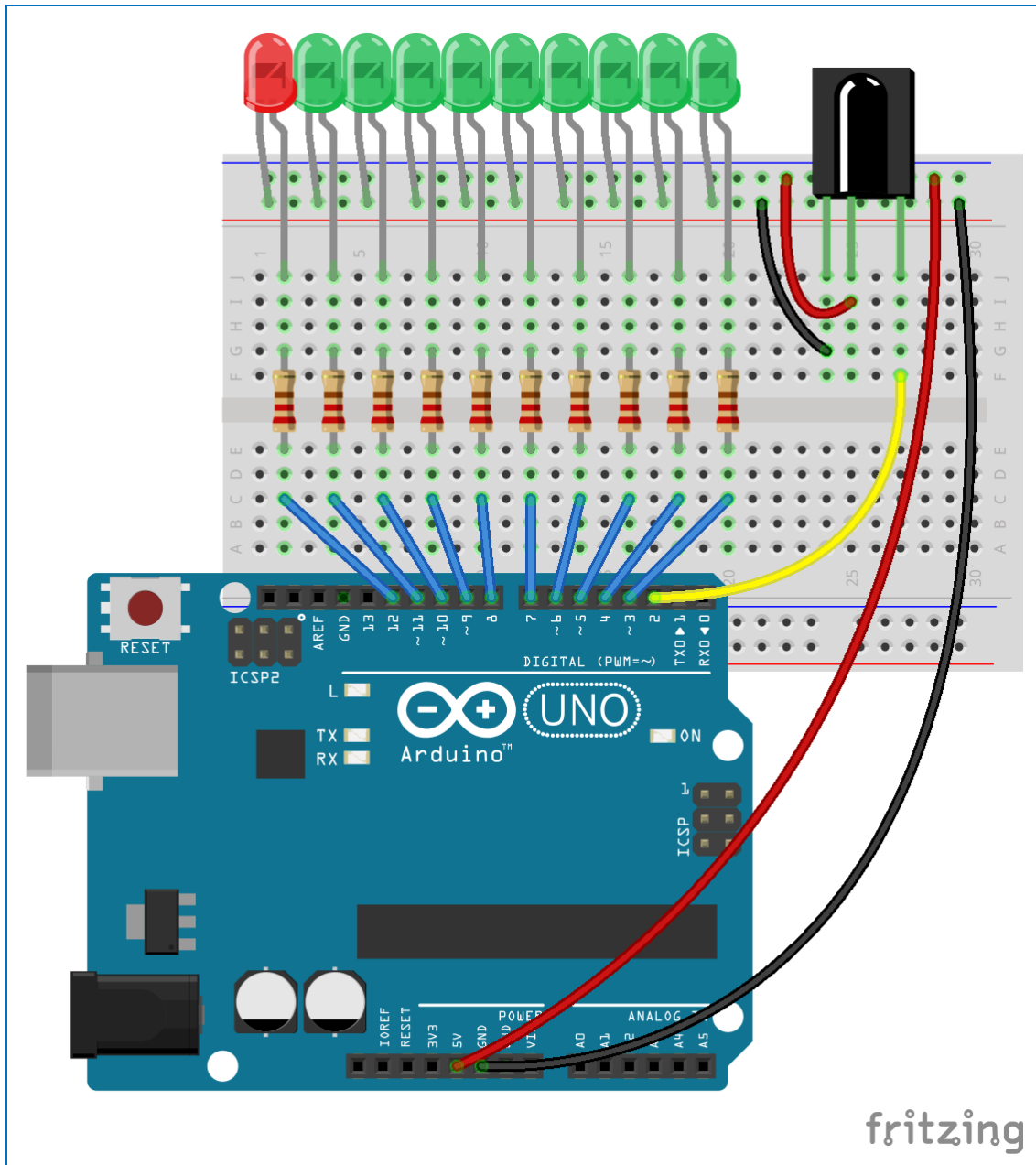
    if (resultado.value == onoff)
    {
      // Es necesario encender primero el dispositivo
    }
  }
}
```

```
if (encendido == true)
{
  encendido = false;
  digitalWrite(led_onoff, HIGH);
  apagar_todo();
  ultimo = 0;
  Serial.println("Acabas de apagar");
  // Para evitar lecturas consecutivas
  delay(100);
}
else
{
  encendido = true;
  digitalWrite(led_onoff, LOW);
  Serial.println("Acabas de encender");
  unico_led(led_1);
  ultimo = led_1;
  // Para evitar lecturas consecutivas
  delay(100);
}
}

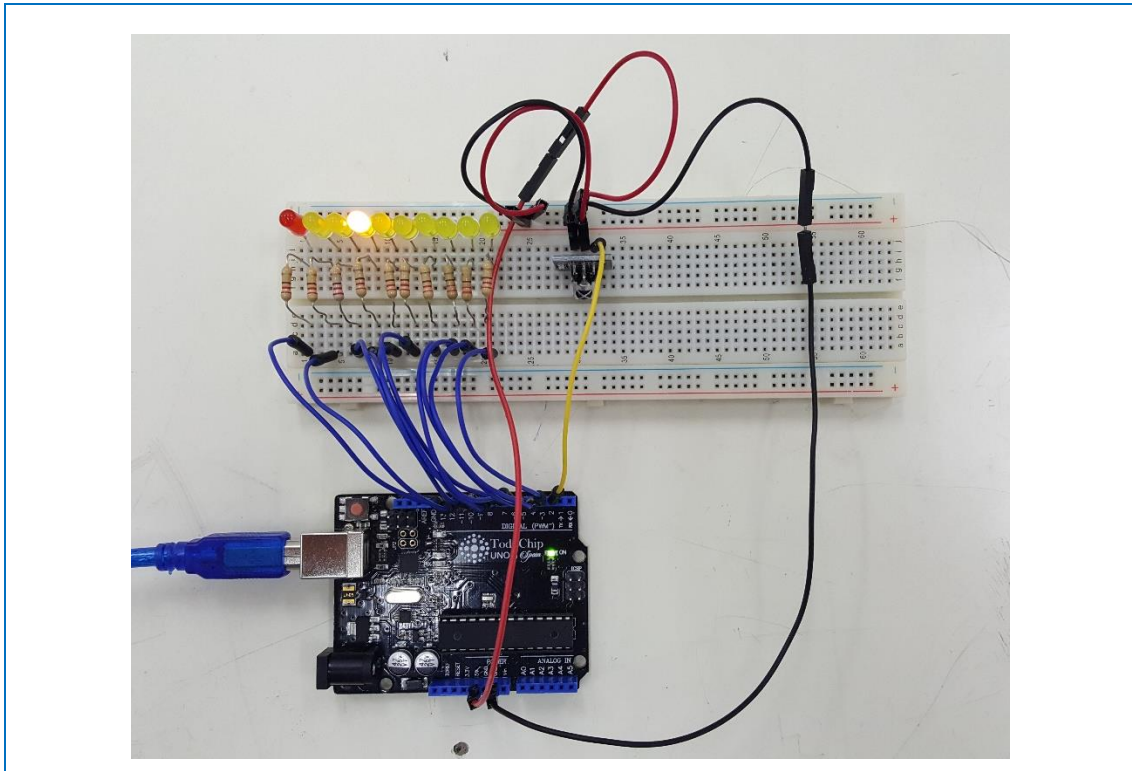
if (encendido == true)
{
  switch (resultado.value){
    case (uno):
      Serial.println("1");
      // Apago el resto de ledes
      // y enciendo led_1
      unico_led(led_1);
      ultimo = led_1;
      break;
    case (dos):
      Serial.println("2");
      unico_led(led_2);
      ultimo = led_2;
      break;
    case (tres):
      Serial.println("3");
      unico_led(led_3);
      ultimo = led_3;
      break;
    case (cuatro):
      Serial.println("4");
      unico_led(led_4);
      ultimo = led_4;
      break;
    case (cinco):
      Serial.println("5");
      unico_led(led_5);
      ultimo = led_5;
      break;
    case (seis):
      Serial.println("6");
      unico_led(led_6);
      ultimo = led_6;
      break;
    case (siete):
      Serial.println("7");
```

```
        unico_led(led_7);
        ultimo = led_7;
        break;
    case (ocho):
        Serial.println("8");
        unico_led(led_8);
        ultimo = led_8;
        break;
    case (nueve):
        Serial.println("9");
        unico_led(led_9);
        ultimo = led_9;
        break;
    case (arriba):
        Serial.print("Cambiando a led ");
        ultimo = ultimo - 1;
        // Si hemos pasado del último
        // Volvemos al primero
        if (ultimo == (led_9-1)) ultimo = led_1;
        Serial.println (ultimo);
        unico_led(ultimo);
        break;
    case (abajo):
        Serial.print("Cambiando a led ");
        ultimo = ultimo + 1;
        // Si hemos pasado del primero
        // Volvemos al último
        if (ultimo == (led_1+1)) ultimo = led_9;
        Serial.println (ultimo);
        unico_led(ultimo);
        break;
    default:
        break;
}
}
// Receptor IR continua a la espera de señal IR
irrecv.resume();
}
```


10_ Esquema de conexionado



11_ Fotografía del montaje



X

Marque con una cruz si acepta que esta unidad didáctica quede recogida bajo una licencia Creative Commons (by - nc - sa) (Reconocimiento, No Comercial, Compartir Igual) para ser compartida por todos los miembros de la comunidad de docentes:

<http://es.creativecommons.org/blog/licencias/>

A background network diagram consisting of a complex web of light blue lines and circular nodes, resembling a circuit board or a data network, set against a dark teal background.

Telefonica

EDUCACIÓN
DIGITAL

SCOLARTIC_
Creando Código