

PRÁCTICA NÚMERO 6:

COMUNICACIÓN ENTRE DOS ARDUINOS CON RF 433 MHz MEDIANTE PULSADOR



Nombre del instituto: I.E.S Politécnico

Nombre del ciclo: Sistemas electrotécnicos y automatizados

ÍNDICE

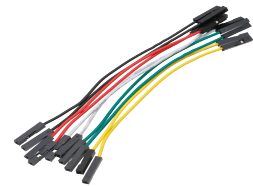
- 1. Descripción de la práctica**
- 2. Materiales**
- 3. Conexión (fritzing)**
- 4. Código de programación**
- 5. Conclusiones y aplicaciones**

1. Descripción de la práctica

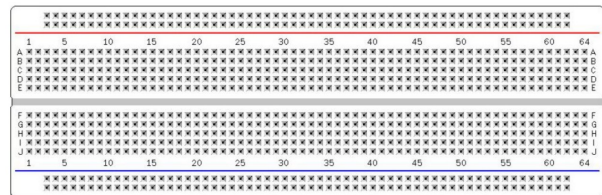
El funcionamiento de dicha práctica consiste en comunicar dos arduinos (uno hace de receptor y otro hace de emisor) con dos módulos de radiofrecuencia (receptor y emisor) a 433 MHz. En este caso el arduino emisor va a emitir una señal para que el receptor genere un parpadeo en el pin 13. En este caso vamos a añadirle un pulsador a nuestro arduino emisor con el objetivo de mandar la señal de pulso mientras mantengamos pulsado el interruptor y una vez soltemos el pulsador deje de emitirse la señal al arduino receptor.

2. Materiales

-Conectores jumper: es un cable con un conector en cada punta (o a veces sin ellos), que se usa normalmente para interconectar entre sí los componentes en una placa de pruebas. Se utilizan de forma general para transferir señales eléctricas de cualquier parte de la placa de prototipos a los pines de entrada/salida de un microcontrolador.



-Placa protoboard: es una especie de tablero con orificios, en la cual se pueden insertar componentes electrónicos y cables para armar circuitos. Como su nombre lo indica, esta tableta sirve para experimentar con circuitos electrónicos, con lo que se asegura el buen funcionamiento del mismo.



-Módulo RF 433MHz: Los módulos de radio frecuencia RF 433MHz son transmisores/receptores inalámbricos que podemos emplear como forma de comunicación entre procesadores como Arduino.

Este tipo de módulos emisor FS1000A) y el receptor (XY-MK-5V) se han hecho populares como medio de comunicación, principalmente, por su bajo coste.

La frecuencia de operación es de 433MHz, aunque también existen módulos similares a 315MHz. Ambas frecuencias pertenecen a bandas libres, por lo que su uso es gratuito.

El alcance depende del voltaje con el que alimentemos el módulo y la antena que usemos. A 5V y con la antena del módulo, el alcance difícilmente excederá de los 2 metros. Alimentando a 12V y con una antena de cobre de 16.5cm el rango en exteriores puede alcanzar 300 metros.

La longitud de la antena óptima depende de la frecuencia del módulo. Antenas habituales para RF 433MHz son un cable de cobre de 16.5cm y la helicoidal 4.5mm de diámetro y 22mm o 34mm de longitud.

La comunicación es simplex (canal único y unidireccional) y tienen baja velocidad de transmisión (típicamente 2400bps). Se realiza, básicamente, por modulación ASK (amplitude shift keying). No

disponen de filtro ni ID por hardware, por lo que si queremos una comunicación robusta tendremos que implementarlo por software

Este tipo de comunicación RF 433MHz es muy frecuente en proyectos caseros de electrónica y robótica, debido a su bajo precio y medio-largo alcance en condiciones adecuadas, aunque tenemos que tener en cuenta que la robustez y calidad de la comunicación puede ser limitada.

EMISOR

D9 — DATA
 5V — Vcc
 GND — GND



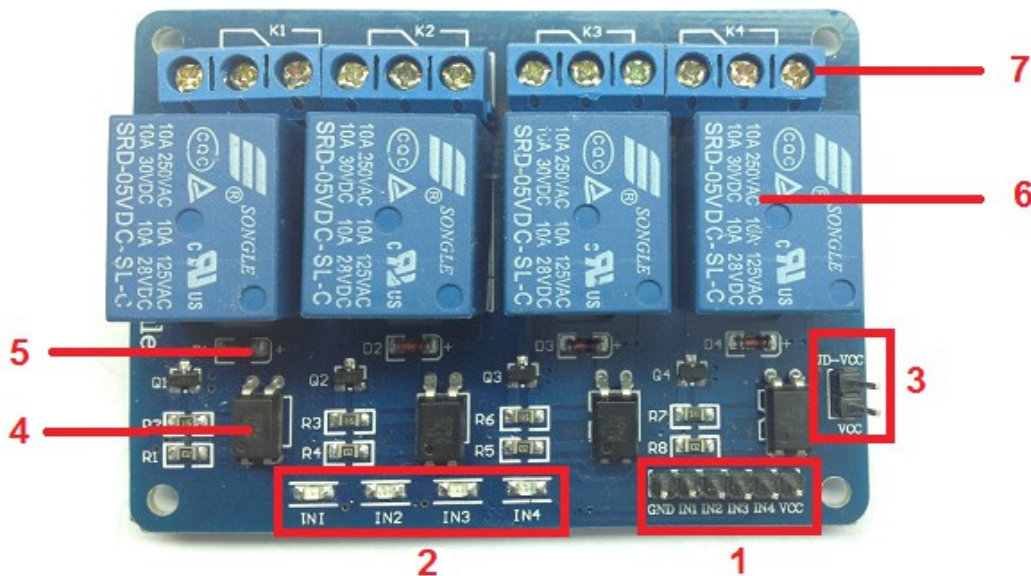
RECEPTOR

5V — Vcc
 DATA — DATA
 D9 — DATA
 GND — GND

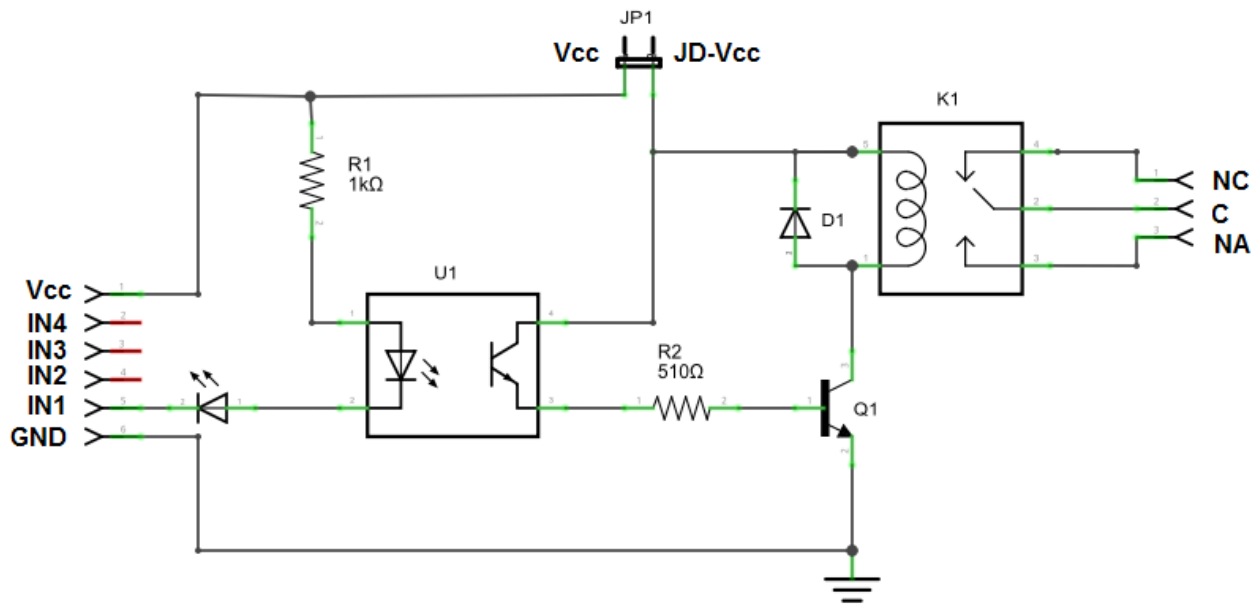


Módulo de relés: Se trata de un módulo de 4 relés (o relays) que funcionan a 5 Voltios, capaces de manejar cargas de hasta 10 Amperes en 250 Voltios, convenientemente aislados mediante optoacopladores de las entradas, las que cuentan con leds individuales que sirven como indicadores de estado.

Los distintos componentes del módulo pueden verse en la siguiente imagen:

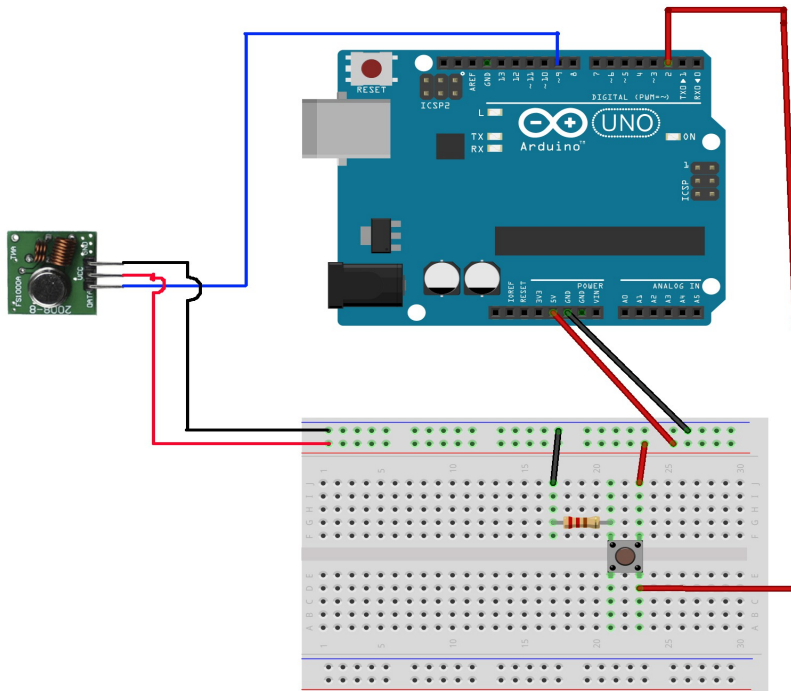


Como se puede apreciar, la placa tiene un conector de entradas (IN1 a IN4) y alimentación (GND es masa o negativo y Vcc es el positivo) [1], cuatro leds que indican el estado de las entradas [2], un jumper selector para la alimentación de los relés [3], cuatro optoacopladores del tipo FL817C [4], cuatro diodos de protección [5], cuatro relés marca SONGLE con bobinas de 5V y contactos capaces de controlar hasta 10 Amperes en una tensión de 250V [6] y cuatro borneras, con tres contactos cada una (Común, Normal abierto y Normal cerrado), para las salidas de los relés [7].



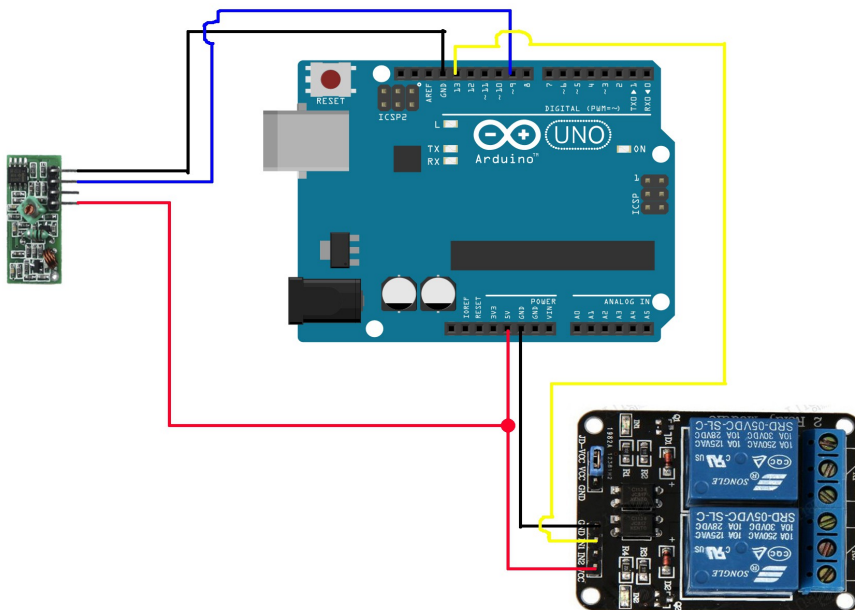
3. Conexionado (fritzing)

EMISOR:



fritzing

RECEPTOR:



fritzing

4. Código de programación

Antes de meter la programación deberemos de descargarnos la librería de los módulos de RF y añadirla de forma automática o manual al programa IDE de arduino.

(<http://www.airspayce.com/mikem/arduino/VirtualWire/index.html>)

Código para arduino emisor:

```
/*
EMISOR

Envia los mensajes "Apagar" y "Encender" en intervalos de un segundo

Escrito por Nano en beneficio de los seres humanos
www.robologs.net
*/

#include <VirtualWire.h>
const int inputPin = 2;

int value = 0;
void setup()
{
  //Iniciamos el Serial y la comunicacion por radio
  pinMode(inputPin, INPUT);
  Serial.begin(9600);
  Serial.write("Sistema encendido\n");
  vw_setup(2000);
}
void loop()
{
  value = digitalRead(inputPin); //lectura digital de pin
  //mandar mensaje a puerto serie en función del valor leído
  if (value == LOW) {
    send("Encender");
  }
  if (value == HIGH) {
    send("Apagar");
  }
  delay(100);
}

//Funcion para enviar el mensaje
void send (char *message)
{
  vw_send((uint8_t *)message, strlen(message)); //Envia el mensaje
  vw_wait_tx(); //Espera hasta que se haya acabado de transmitir todo

  Serial.println(message); //Muestra el mensaje por Serial
}
```

Código para arduino receptor:

```
/*
RECEPTOR

Apaga el LED 13 si recibe el mensaje "Apagar"
Enciende el LED 13 si recibe el mensaje "Encender"

Escrito por Nano en beneficio de los seres humanos
www.robologs.net
*/

#include <VirtualWire.h>

//Creamos un mensaje
//La constante VW_MAX_MESSAGE_LEN viene definida en la libreria
byte message[VW_MAX_MESSAGE_LEN];
byte messageLength = VW_MAX_MESSAGE_LEN;

void setup()
{
  pinMode(13, OUTPUT); //Configuramos el pin 13

  Serial.begin(9600); //Iniciamos el Serial
  Serial.println("Iniciando...");

  vw_setup(2000);
  vw_rx_start();
}
void loop()
{
  if (vw_get_message(message, &messageLength))
  {
    if(comparar("Encender") == 0){
      digitalWrite(13, HIGH);
      Serial.write("LED Encendido\n");
    }
    else if(comparar("Apagar") == 0)
    {
      digitalWrite(13,LOW);
      Serial.write("LED Apagado\n");
    }
  }
}

char comparar(char* cadena) {
  //Esta funcion compara el string cadena con el mensaje recibido.
  //Si son iguales, devuelve 1. Si no, devuelve 0.

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



```
if(message[i] != cadena[i])
{
    return 1;
}
}

return 0;
}
```

5. Conclusiones y aplicaciones

Simplemente es una modificación de la anterior práctica en la que esta vez no se manda la señal con solo conectar el arduino, sino que una vez que tengamos conectado el arduino podemos mandar o no la señal a nuestro antojo mediante el uso de un pulsador. Como ventaja es que podemos activar y desactivar nuestro receptor sin tener que conectar o desconectar el arduino transmisor como ocurría en el caso anterior.

Como aplicación personalmente lo utilizaría para el encendido de una luminaria a través del módulo de relé o la activación y desactivación de un motor de una persiana o de una puerta de cochera.