

PROYECTO A4: MANDO Y RECEPTOR DE INFRARROJOS



Llevamos tres días con Roby intentando hacer un programa para la prueba del laberinto... Y no hay forma. Roby está más atascado que una libélula en una telaraña.

Y un alumno mío me ha preguntado: ¡Oye, maestro! ¿Y si pudiera manejarlo con un mando? ¡Lo controlo desde lejos y nadie se da cuenta!

No es muy ético, pero bueno. Como en las olimpiadas. Mientras no te pillen, ¡no te quitan la medalla!



por [Aurelio Gallardo](#)

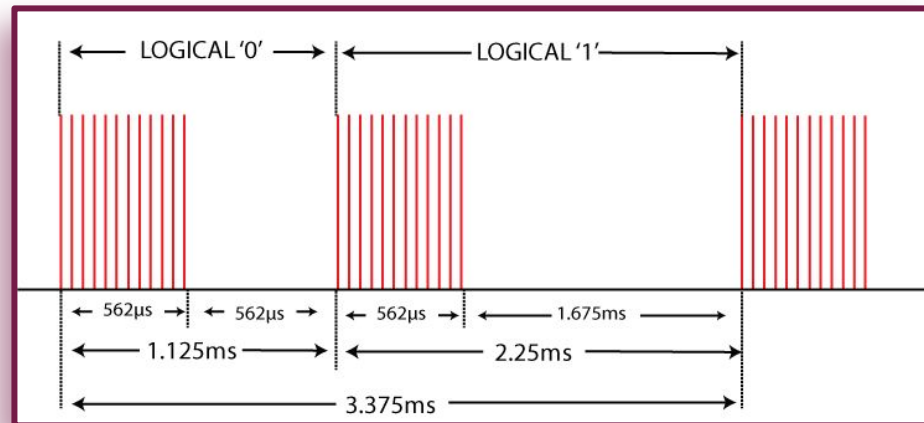
Mandos a distancia ¿cómo funcionan?



Los mandos a distancia son muy comunes en nuestra vida diaria. Nos permiten controlar TV, aires acondicionados... Utilizan distintos protocolos de envío de datos: NEC, SONY, RC5, RC6... Veremos como ejemplo el protocolo NEC: un led infrarrojo, de unos 940nm, envía una onda modulada de 38KHz. El resto de protocolos varía esta modulación entre los 36 y los 50KHz.

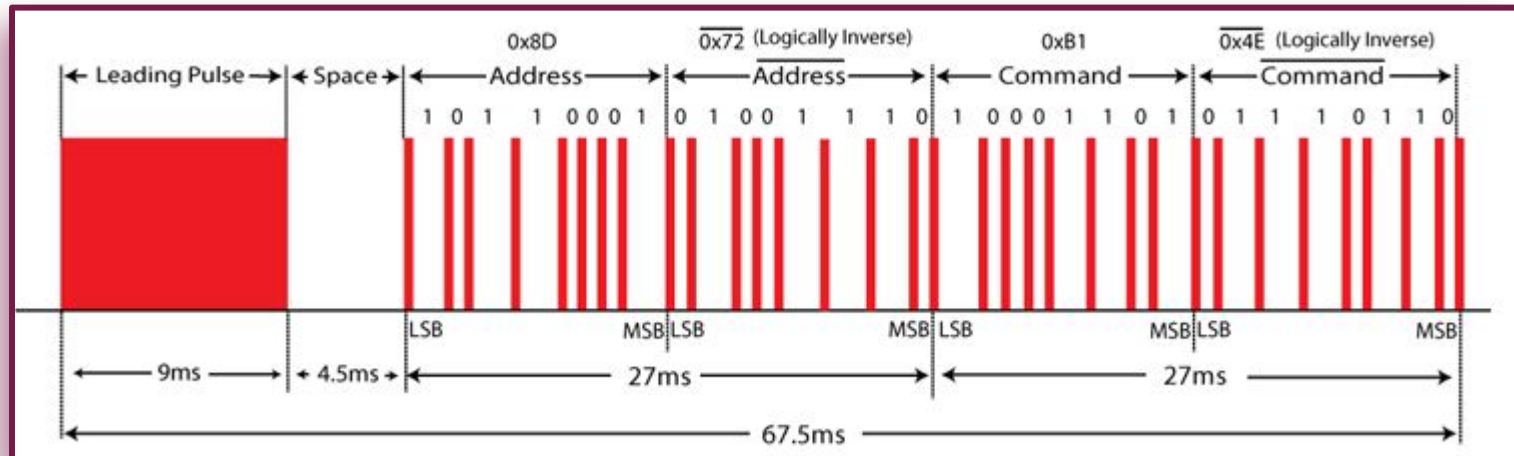
La modulación se consigue por PDM (Pulse Distance Modulation). Un 0 lógico y un 1 lógico se caracterizan por sus períodos de actividad y de “silencio”. Según la siguiente gráfica:

¿Funciona mi mando?
Apunta el led hacia la cámara de un móvil y haz como si hicieras una fotografía. Pula un botón del mando y verás el parpadeo a través de la pantalla del móvil



La información...

Ya sabemos mandar ceros y unos. Pero ¿qué información envía un mando? Pues envía 8 bits indicando el código del mando y 8 bits indicando el código de la instrucción. Así pueden usarse 256 mandos distintos con 256 códigos diferentes. Además, cada código se envía por duplicado (normal y negado) tras un período de 9ms de señal y 4.5 ms de pausa.

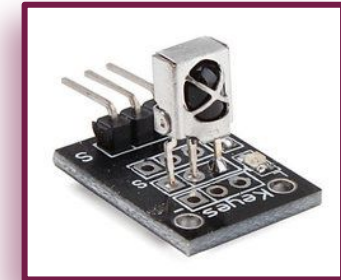


El alcance un mando infrarrojo suele ser de unos 3m y esa distancia disminuye si la visual del receptor no es la normal al mismo.

KY-022 (VS1838)

El receptor KY-022 utiliza el sensor de infrarrojos 1838, que es un receptor de tres terminales incluyendo un demodulador en la banda de 36-38kHz, un filtro PCM (Pulse Code Modulation) y preamplificación rechazo de luz ambiental.

El esquema de conexionado a Arduino es el que sigue:



Aliexpress: 0.26€
por cada uno + g.
env.

Podemos variar
el pin de
entrada de
datos a
cualquier otro

Asegurarse de que las conexiones hacen bien contacto



El programa de ejemplo

En primer lugar, cargamos la librería correspondiente de Arduino. Podemos encontrarla en el [Gestor de librerías...](#)

IRremote by shirriff Versión 2.2.3 **INSTALLED**
Send and receive infrared signals with multiple pr
[More info](#)

Enlace a [GITHUB](#) de la librería.
Enlace a los distintos [objetos](#),
[métodos y funciones](#).

```
#include <IRremote.h> //Biblioteca control infrarrojo
// IRremote by shirriff

int RECV_PIN = 11; // Pin de recepción de

IRrecv receptor(RECV_PIN); // inicializo objeto de recepción

decode_results resultados; //objeto resultados

// *****
// SETUP
// *****
void setup()
{
  Serial.begin(9600); // inicializa monitor serie
  Serial.println("[I N I C I O]");
  receptor.enableIRIn();
  // Inicializa el receptor para escuchar datos
  receptor.blink13(true); //permite que parpadee
  // pin 13 al recibir datos.
}
```

Explicación de órdenes principales
https://www.pjrc.com/teensy/td_libs_IRremote.html

El programa de ejemplo

```
// *****  
// LOOP  
// *****  
void loop() {  
  if (receptor.decode(&resultados)) {  
    // Si consigue decodificar un resultado..  
    // https://www.pjrc.com/teensy/td\_libs\_IRremote.html  
  
    if (resultados.decode_type == NEC) {  
      Serial.print("NEC: ");  
    } else if (resultados.decode_type == SONY) {  
      Serial.print("SONY: ");  
    } else if (resultados.decode_type == RC5) {  
      Serial.print("RC5: ");  
    } else if (resultados.decode_type == RC6) {  
      Serial.print("RC6: ");  
    } else if (resultados.decode_type == UNKNOWN) {  
      Serial.print("UNKNOWN: ");  
    } // imprime el tipo de codificación recibida  
  
    Serial.println(resultados.value, HEX);  
    Serial.println("bits: " + (String) resultados.bits);  
    // Más información con las órdenes  
    // results.rawbuf (un array de los pulsos IR)  
    // results.rawlen (número de items almacenados en el array)  
  
    receptor.resume(); // Preparado para recibir el siguiente valor  
  }  
}
```

Ejecutado el programa, lanza el Monitor Serie para ver por pantalla los códigos recibidos.

Si no estás bien orientado, se puede recibir un dato espurio, normalmente tipo UNKNOWN



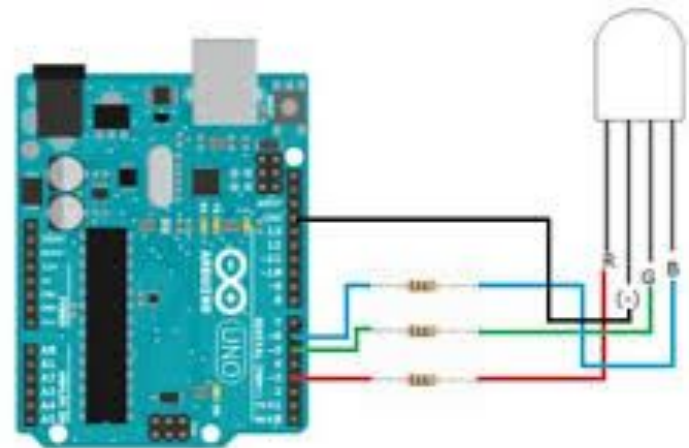
Ejercicio A: Estudio del daltonismo



Para realizar esta práctica necesitarás los siguientes...

Materiales: arduino uno, ordenador con IDE de Arduino, placa protoboard, cables dupont macho-macho y hembra-macho, cable USB de conexión, receptor VS1838 (KY022), mando a distancia (¿cualquiera?), led tricolor o RGB cátodo común (negativo, pata más larga, a tierra (Si no dispones de él, podemos simularlo con tres leds separados: rojo, verde y azul)).

- El trabajo consiste en un estudio “casero” del daltonismo.
- Usando el led tricolor, y nuestro mando, pensaremos en un programa en el que, pulsando los distintos botones del mismo, vayamos cambiando la distinta cantidad de rojo, verde y azul.
- Un paciente deberá contarnos qué color de luz es capaz de contemplar.
- El rojo, a la izquierda del común, debería llevar una resistencia de 220 ohm. El azul y el verde, de 100 ohm.



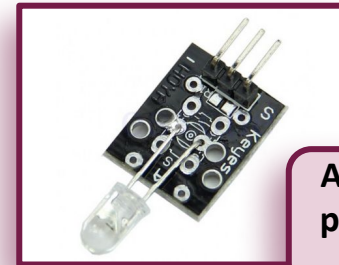
Enviando datos...

Si quieres enviar datos, debes usar el dispositivo KY-005. Este diodo led infrarrojo es capaz de enviar los datos modulados necesarios.

Según la web https://www.pjrc.com/teensy/td_libs_IRremote.html el programa de transmisión es sencillo. Eso sí, en **ARDUINO UNO debe enviarse por el pin 3** (estudiar cada caso en qué pin, según la placa). Teóricamente, así, se puede “copiar” un mando.

En ATtiny85 es el pin 1 (ver <https://github.com/z3t0/Arduino-IRremote>) en el README.

En algunos protocolos, como el de SONY, los códigos hay que enviarlos más de una vez.



Aliexpress: 0.35€
por cada uno + g.
env.

```
#include <IRremote.h>

IRsend irsend;
int pinButton = 11;
int valor_button;

void setup()
{
  Serial.begin(9600);
  pinMode(pinButton, INPUT);
}

void loop()
{
  valor_button = digitalRead(pinButton);
  if(valor_button == HIGH)
  {
    irsend.sendNEC(0x20DF10EF, 32);
    Serial.println("Enviado");
    delay(100);
  }
}
```