# Ejercicio 10: Infrarrojo (mando a distancia)

<http://www.dx.com/p/hengjiaan-ir-infrared-receiver-module-kit-w-remote-controller-461997>

La tecnología de emisión de datos por infrarrojos es característica de la mayoría de los mandos a distancia que tenemos en el hogar para controlar los dispositivos multimedia. Su funcionamiento consiste en emitir rayos de luz intermitentes que configuran una señal que es recibida por un aparato con forma de bombilla. Si el mando tiene una bombillita o cristal por delante, funciona por infrarrojos.

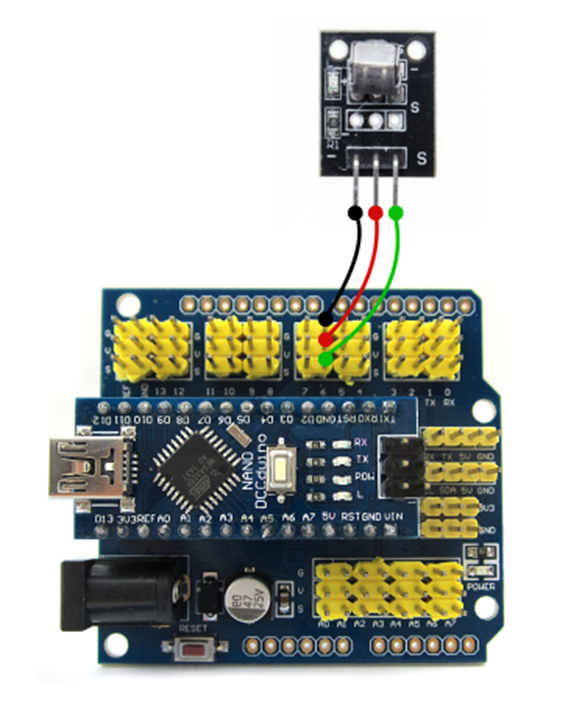
Las ventajas más destacables son la sencillez de utilización, el bajo coste, y la disponibilidad de esta tecnología en multitud de aparatos.

Las desventajas principales son que no puede haber obstáculos en la línea entre emisor y receptor, tampoco atraviesa paredes, y la principal, que cada aparato tiene su propio mando y acabamos llenando la mesita de la sala de mandos.

En este ejercicio vamos a recibir señales de un mando a distancia para poder ejecutar las acciones que queramos, por ejemplo encenderemos y apagaremos una luz, o aumentaremos y disminuiremos la intensidad de esa luz.

Al ser una tecnología muy madura, ya se conoce el formato de envío que utilizan los diferentes fabricantes, con lo cual la librería que vamos a usar contiene funciones sencillas para trabajar con cada tipo de mando. No obstante, a modo de ejemplo vamos a mostrar todos los datos que vienen desde un mando cualquiera ya que si no conocemos su modelo, también podemos copiar sus órdenes e integrar todas en un mando único, que controle los aparatos comerciales y los “propios”.

El esquema de conexiones necesario:



El código relevante para la aplicación:

// instalar previamente la librería IRRemote

#include <Irremote.h>

// asignar el pin donde llegan los datos y configurar para infrarrojo

IRrecv irrecv(PIN\_LECTURA);

irrecv.enableIRIn();

// copiar los datos que llegan en la variable mensaje

irrecv.decode(&mensaje)

El código completo:

/\*\*

Ejemplo de mando a distancia por infrarrojos

Borrar la librería IRRemoteRobot si existe porque colisiona con la que vamos a instalar

(cerrar arduino, ir a la carpeta librerías y borrarla)

Instalar la librería IRRemote

Ayuda: http://www.pcbheaven.com/userpages/RC\_Protocol\_and\_Modulation/

PINSET:

- GND a tierra

- VCC a 5 Voltios

- Signal a pin 6

\*/

#include <IRremote.h>

int PIN\_LECTURA = 6;

IRrecv irrecv(PIN\_LECTURA);

decode\_results mensaje;

void setup() {

Serial.begin(9600);

irrecv.enableIRIn();

}

void loop() {

// copiar los datos que llegan en la variable mensaje

if (irrecv.decode(&mensaje)) {

// mostrar todos los datos posibles;

dump(&mensaje);

irrecv.resume();

delay (100);

// ejecutar en función del botón del mando

if (mensaje.value == 0xFF02FD) {

Serial.println("¡¡¡¡¡¡¡¡¡¡¡¡ BOTON OK !!!!!!!!!!");

}

}

}

void dump(decode\_results \*results) {

// Dumps out the decode\_results structure.

// Call this after IRrecv::decode()

int count = results->rawlen;

if (results->decode\_type == UNKNOWN) {

Serial.print("Unknown encoding: ");

}

else if (results->decode\_type == NEC) {

Serial.print("Decoded NEC: ");

}

else if (results->decode\_type == SONY) {

Serial.print("Decoded SONY: ");

}

else if (results->decode\_type == RC5) {

Serial.print("Decoded RC5: ");

}

else if (results->decode\_type == RC6) {

Serial.print("Decoded RC6: ");

}

else if (results->decode\_type == PANASONIC) {

Serial.print("Decoded PANASONIC - Address: ");

Serial.print(results->address, HEX);

Serial.print(" Value: ");

}

else if (results->decode\_type == LG) {

Serial.print("Decoded LG: ");

}

else if (results->decode\_type == JVC) {

Serial.print("Decoded JVC: ");

}

else if (results->decode\_type == AIWA\_RC\_T501) {

Serial.print("Decoded AIWA RC T501: ");

}

else if (results->decode\_type == WHYNTER) {

Serial.print("Decoded Whynter: ");

}

Serial.print(results->value, HEX);

Serial.print(" (");

Serial.print(results->bits, DEC);

Serial.println(" bits)");

Serial.print("Raw (");

Serial.print(count, DEC);

Serial.print("): ");

for (int i = 1; i < count; i++) {

if (i & 1) {

Serial.print(results->rawbuf[i]\*USECPERTICK, DEC);

}

else {

Serial.write('-');

Serial.print((unsigned long) results->rawbuf[i]\*USECPERTICK, DEC);

}

Serial.print(" ");

}

Serial.println();

}