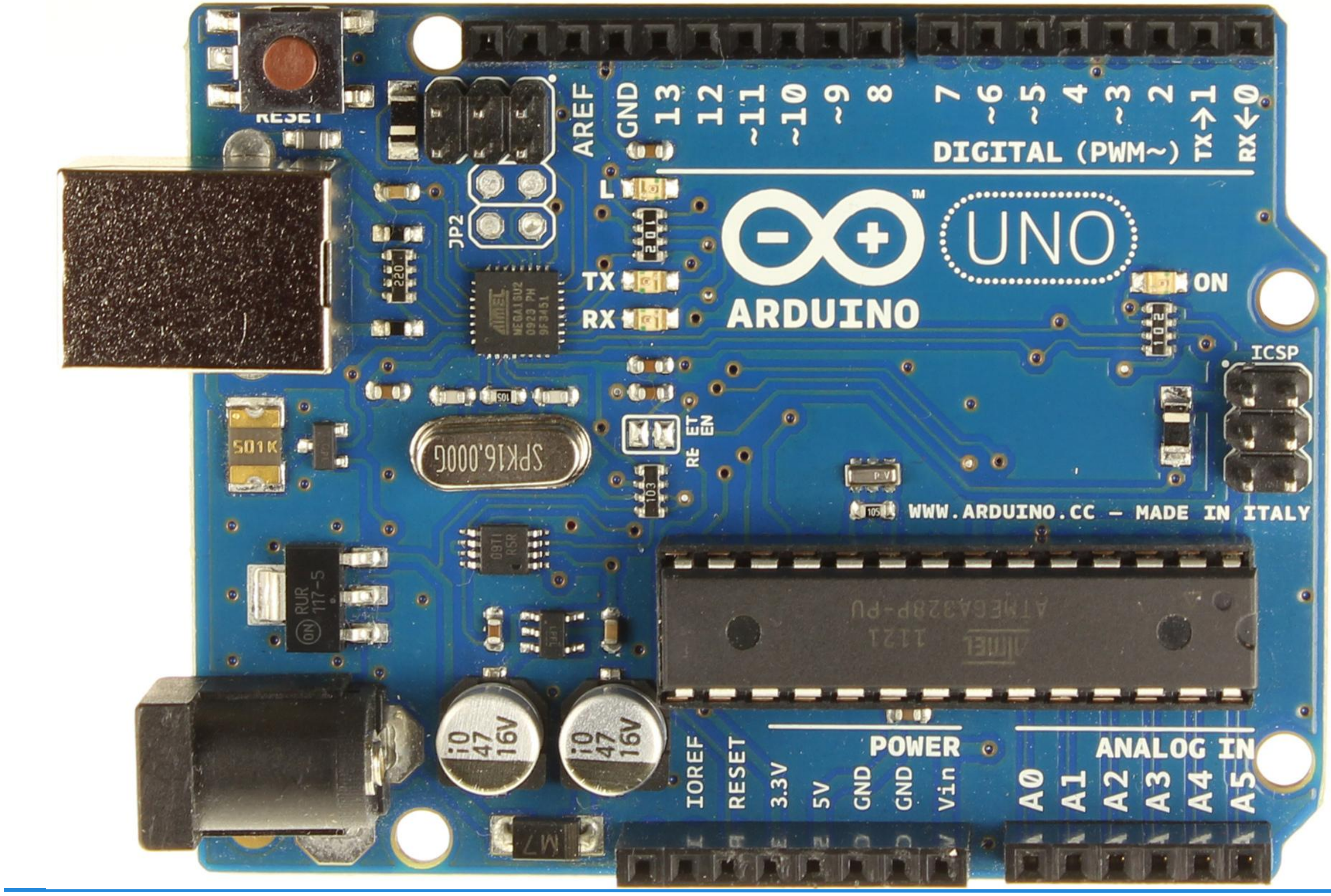
Curso avanzado sobre Arduino

ElCacharreo

ElCacharreo.com

Arduino avanzado: Presente



Arduino avanzado: Presente

José Antonio Vacas Martínez

blog

javacasm@elcacharreo.com

twitter

linkedin

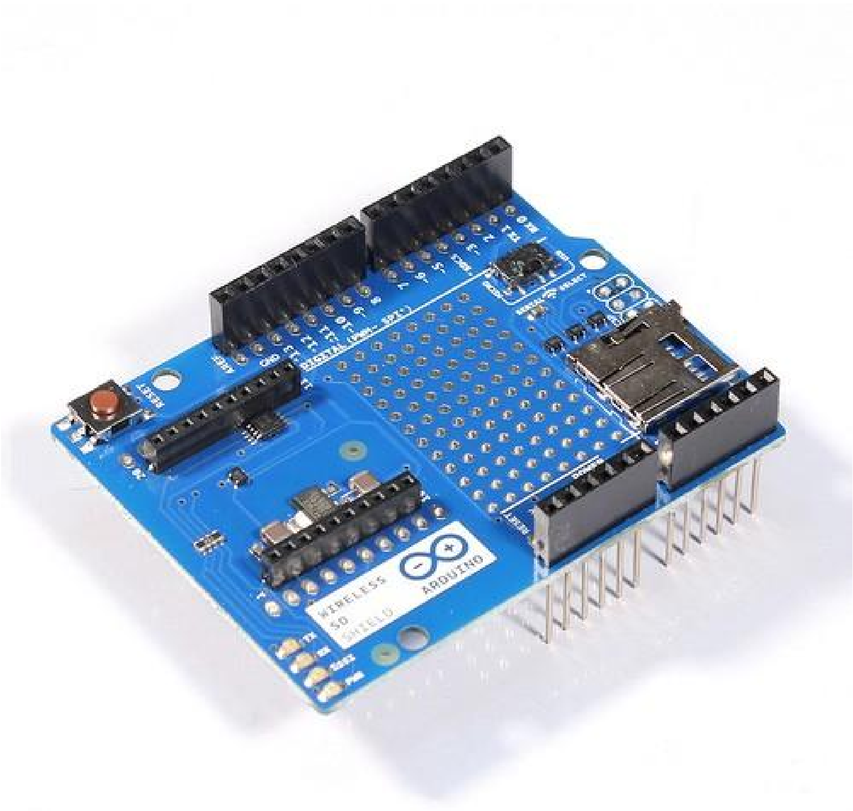
Arduino avanzado: Zigbee

2 Modos:

- Como transceiver: envía y recibe lo que arduino le dice via TX y RX

- StandAlone: se puede programar para recoger datos y enviar con cierta freciuencia

Arduino avanzado: Zigbee

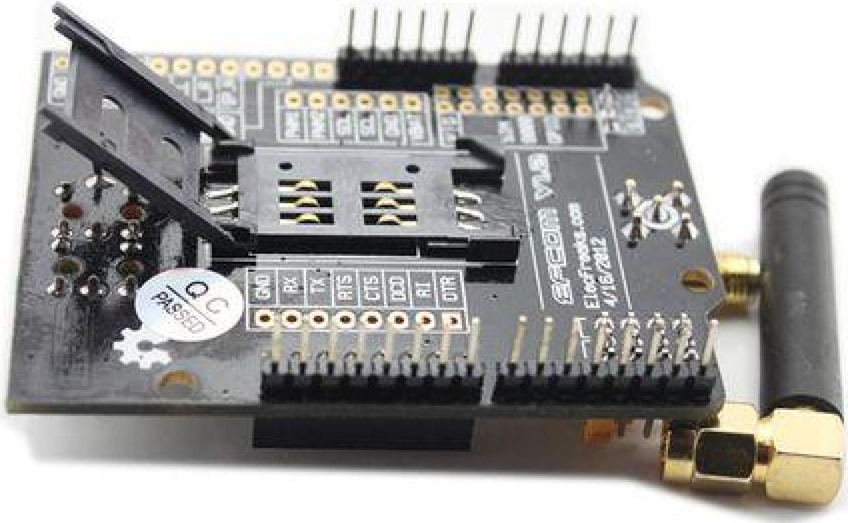


Arduino avanzado: Zigbee

```
void setup()
{
 Serial.begin(9600);
 pinMode(9,OUTPUT);
}

void loop()
{
 Serial.print('H');
 digitalWrite(9,HIGH);
 delay(1000);
 Serial.print('L');
   digitalWrite(9,LOW);
 delay(1000);
}
```

```
const int ledPin = 13; // the pin that the LED is attached to
int incomingByte;      // a variable to read incoming serial data into
void setup() {
 Serial.begin(9600);
 pinMode(ledPin, OUTPUT);}
void loop() {
 // see if there's incoming serial data:
 if (Serial.available() > 0) {
   // read the oldest byte in the serial buffer:
   incomingByte = Serial.read();
   // if it's a capital H (ASCII 72), turn on the LED:
   if (incomingByte == 'H') {
     digitalWrite(ledPin, HIGH);
   }
   // if it's an L (ASCII 76) turn off the LED:
   if (incomingByte == 'L') {
     digitalWrite(ledPin, LOW);   } }}
```

Arduino avanzado: GPRS



Arduino avanzado: GPRS

```
//Serial Relay - Arduino will patch a
//serial link between the computer and the GPRS Shield
//at 19200 bps 8-N-1
//Computer is connected to Hardware UART
//GPRS Shield is connected to the Software UART

#include <SoftwareSerial.h>

SoftwareSerial mySerial(2, 3);

void setup()
{
  mySerial.begin(19200);          // the GPRS baud rate
  Serial.begin(19200);            // the GPRS baud rate
}

void loop()
{
  if (mySerial.available())
    Serial.write(mySerial.read());
  if (Serial.available())
    mySerial.write(Serial.read());

}
```

Comandos AT

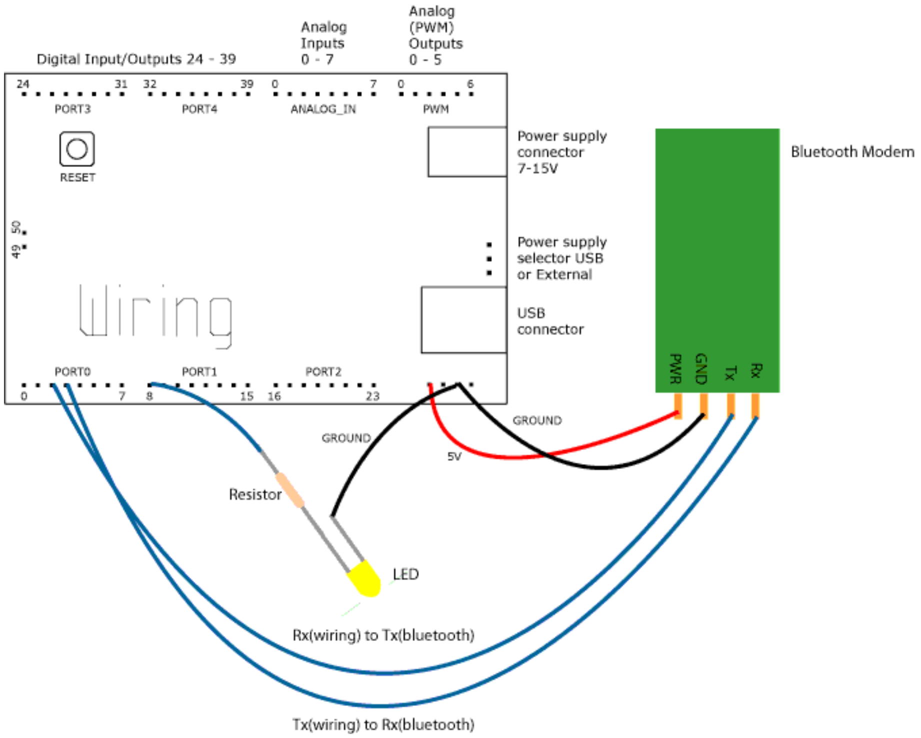
AT+IPR=19200

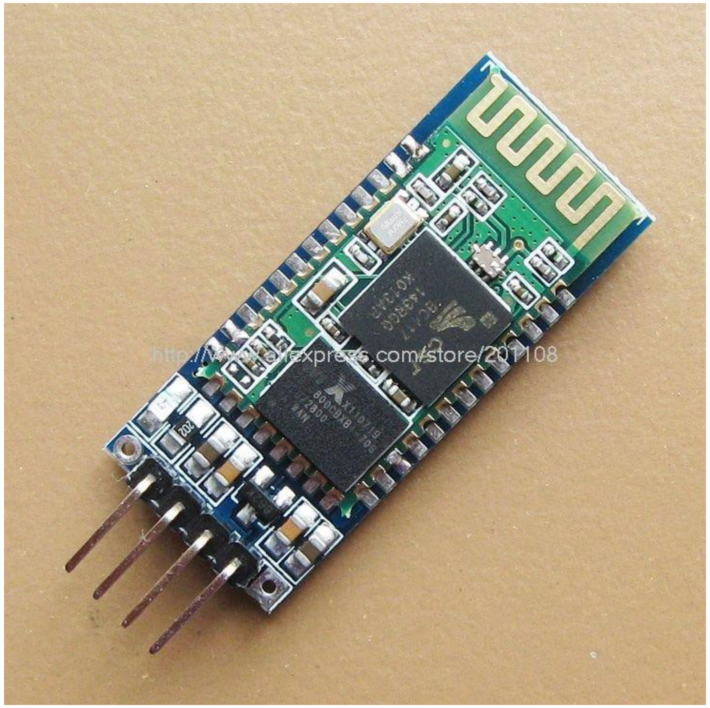
AT+CMGF=1

AT+CMGS="+918446043032"

http://www.elecfreaks.com/wiki/index.php?title=EFCom_GPRS/GSM_Shield

Arduino avanzado: Bluetooth

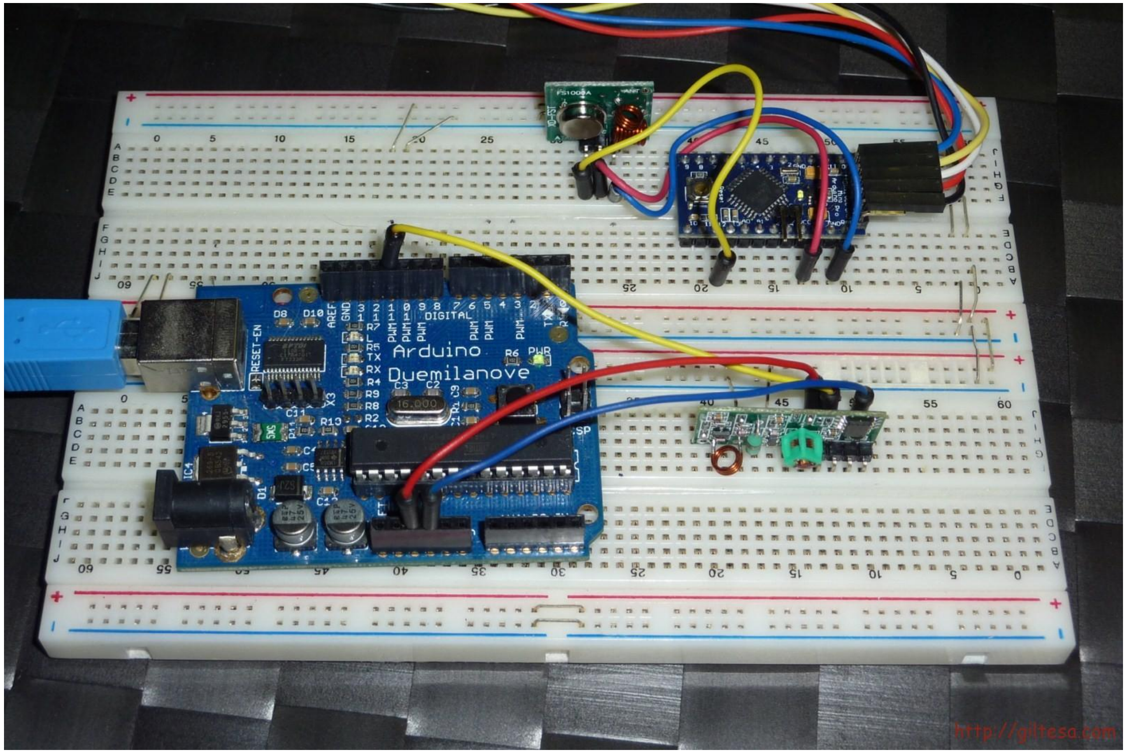

Digital Input/Outputs 24 - 39
Analog Inputs 0 - 7
Analog (PWM) Outputs 0 - 5
24
31
32
39
0
7
0
6
PORT3
PORT4
ANALOG_IN
PWM
RESET
Power supply connector 7-15V
Power supply selector USB or External
USB connector
49
50
Wiring
PORT0
PORT1
PORT2
0
7
8
15
16
23
Bluetooth Modem
PWR
GND
Tx
Rx
GROUND
GROUND
5V
Resistor
LED
Rx(wiring) to Tx(bluetooth)
Tx(wiring) to Rx(bluetooth)

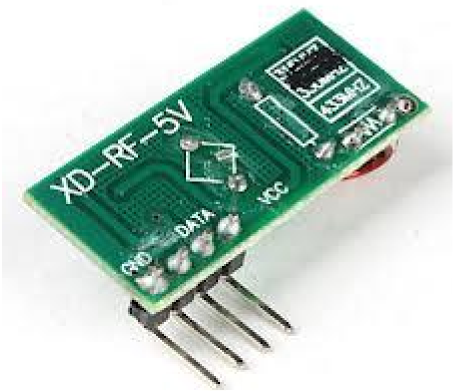


Aplicaciones Android:
BlueChat y Bluetooth Manager

RF 433MHz

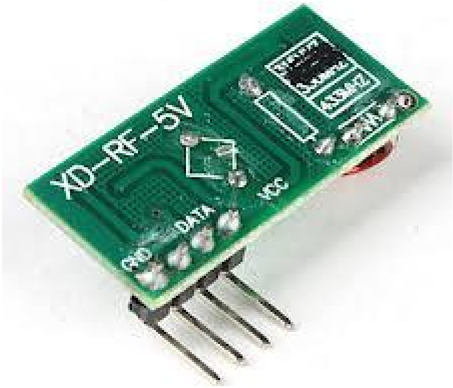
Estaciones meteorológicas, interruptores remotos por radio
http://www.instructables.com/id/Wireless-power-outlets-for-home-automation-using-A/





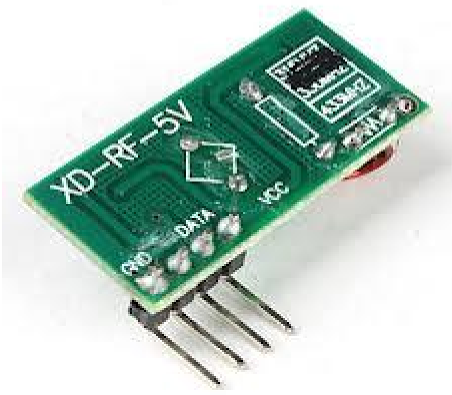
RF 433MHz :Emisor

```
#include <VirtualWire.h>

const int led_pin = 11;
const int transmit_pin = 12;
const int receive_pin = 2;
const int transmit_en_pin = 3;

void setup(){
  // Initialise the IO and ISR
  vw_set_tx_pin(transmit_pin);
  vw_set_rx_pin(receive_pin);
  vw_set_ptt_pin(transmit_en_pin);
  vw_set_ptt_inverted(true); // Required for DR3100
  vw_setup(2000);       // Bits per sec}

byte count = 1;

void loop(){
  char msg[7] = {'h','e','l','l','o',' ','#'};

  msg[6] = count;
  digitalWrite(led_pin, HIGH); // Flash a light to show transmitting
  vw_send((uint8_t *)msg, 7);
  vw_wait_tx(); // Wait until the whole message is gone
  digitalWrite(led_pin, LOW);
  delay(1000);
  count = count + 1;}
```



RF 433MHz :Receptor

```
#include <VirtualWire.h>

const int led_pin = 6;
const int transmit_pin = 12;
const int receive_pin = 11;
const int transmit_en_pin = 3;
void setup(){
  delay(1000);
  Serial.begin(9600);// Debugging only
  Serial.println("setup");
  vw_set_tx_pin(transmit_pin);    // Initialise the IO and ISR
  vw_set_rx_pin(receive_pin);
  vw_set_ptt_pin(transmit_en_pin);
  vw_set_ptt_inverted(true); // Required for DR3100
  vw_setup(2000);     // Bits per sec
  vw_rx_start();      // Start the receiver PLL running}
void loop(){
  uint8_t buf[VW_MAX_MESSAGE_LEN];
  uint8_t buflen = VW_MAX_MESSAGE_LEN;
  if (vw_get_message(buf, &buflen)) // Non-blocking    {
     int i;
     digitalWrite(led_pin, HIGH); // Flash a light received good message
      Serial.print("Got: ");

      for (i = 0; i < buflen; i++)       {
        Serial.print(buf[i], HEX);
        Serial.print(' ');      }
      Serial.println();
    digitalWrite(led_pin, LOW);    }}
```



RF 433MHz :¿expectativas?

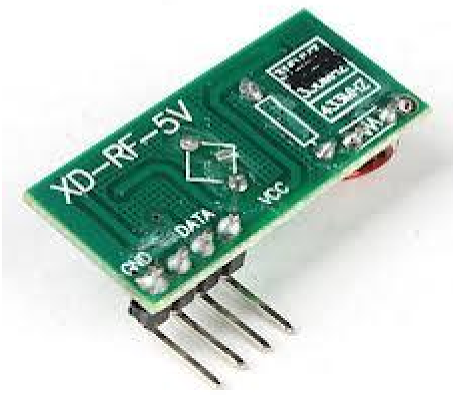
Realistic Performance Expectations

Many of the cheapest RF modules are sold with very unrealistic claims of data rate and maximum communication distance, and sometimes with very little (or even incorrect) documentation. VirtualWire will help these modules perform as well as they can, but the old saying applies: "you get what you pay for".

For example, the 315 MHz modules shown above were documented with only this image. These modules worked very reliably when sitting only close to each other on a table. When separated by about 20 feet with ordinary office furniture, and a 13 cm wire attached to each (in the middle of the 10 to 15 cm suggested), they were able to communicate, but approximately 20% of messages were corrupted.

Perhaps using better antennas could help, but each board has a loading coil that appears to be designed for relatively short antennas, and no other documentation seems to exist regarding best antennas.

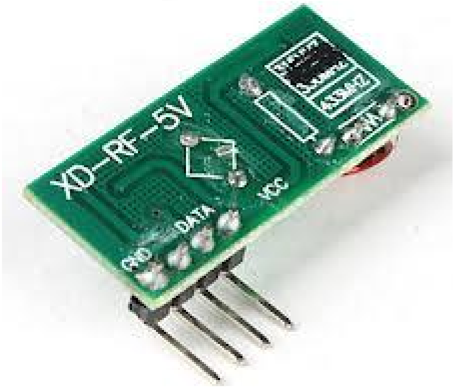
These modules can work well for low performance, non-critical applications. For more demanding applications, more sophisticated (and more expensive) RF modules should be considered.



RF 433

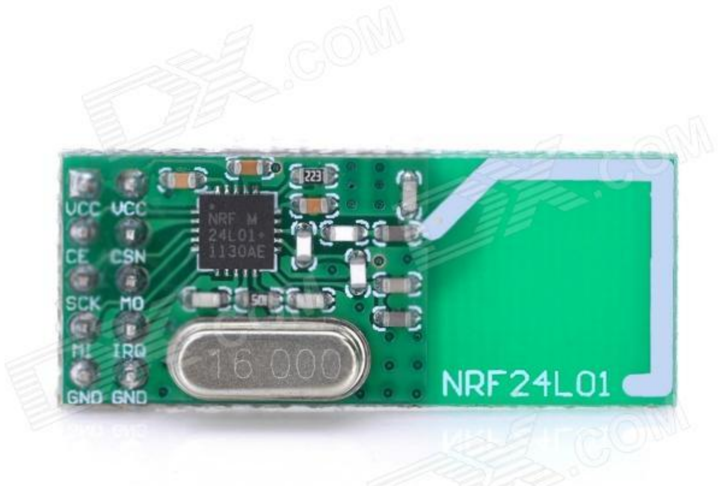
Estaciones meteorológicas, interruptores remotos por radio
http://www.instructables.com/id/Wireless-power-outlets-for-home-automation-using-A/

http://arduinohome.com/category/proyectos/
http://arduinohome.com/2012/04/04/probando-el-modulo-de-radio-frecuencia-433mhz-para-domotica/
http://jeelabs.org/2011/01/28/meet-the-ook-433-plug/
http://jeelabs.org/?s=ookDecoder+
http://jeelabs.org/2010/04/13/an-ook-scope/



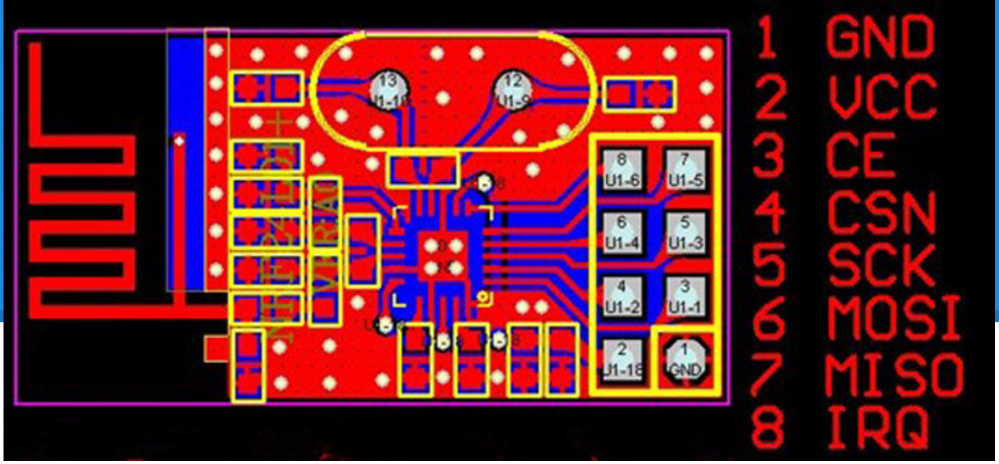
Nordic NRF2401

Producto: http://dx.com/p/nrf24l01-2-4ghz-wireless-transceiver-module-126467
Información http://club.dx.com/forums/forums.dx/threadid.1175714
Código ejemplo de cliente y servidor: https://sites.google.com/site/ucieecs129a/team-4-files/snippets/arduinosendreceivecode
Playground: http://playground.arduino.cc/InterfacingWithHardware/Nrf24L01



2.4GHz
250KBits/s, 1MBits/s o 2MBits/s
1.8V a 3.6V
DataSheet http://www.nordicsemi.com/eng/nordic/download_resource/8765/2/27999719
Detalles http://www.nordicsemi.com/eng/Products/2.4GHz-RF/nRF24L01P

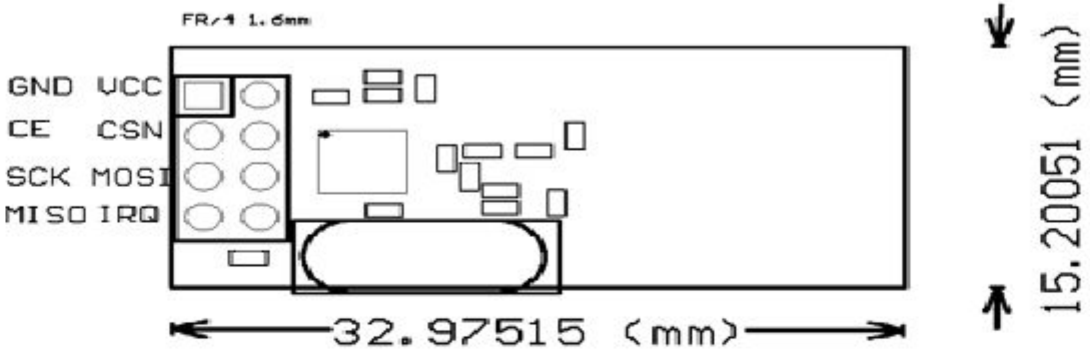
Nordic NRF2401



Página oficial de la librería Mirf https://github.com/aaronds/arduino-nrf24l01

Existen otras 2 versiones:
Una apta para attiny https://github.com/stanleyseow/arduino-nrf24l01
Una apta para raspberry https://github.com/stanleyseow/RF24

Documentación en http://playground.arduino.cc/InterfacingWithHardware/Nrf24L01



Conclusiones

Gracias por vuestra atención