

CODIGO PARA EL SCANNER

```
/*
 *
 * Este programa es software libre; puedes redistribuirlo y / o
 * Modificarlo bajo los términos de la Licencia Pública General de GNU.
 * Versión 2 tal como fue publicada por la Free Software Foundation.
 *
 * Copie y pegue el código en su ide de arduino y a experimentar amigos!!!
 *
 */

/**
 * Escaner de canal
 *
 * * Ejemplo para detectar interferencias en los distintos canales disponibles.
 * * Esta es una buena herramienta de diagnóstico para comprobar si está
 * * eligiendo un
 * * Buen canal para su aplicación.
 * *
 * * Inspirado en cpixip.
 * * Ver http://arduino.cc/forum/index.php/topic,54795.0.html
 */

#include <SPI.h>
#include "nRF24L01.h"
#include "RF24.h"
#include "printf.h"

//
// Hardware configuration
//

// Set up nRF24L01 radio on SPI bus plus pins 9 & 10

RF24 radio(9,10);

//
// Channel info
//

const uint8_t num_channels = 128;
uint8_t values[num_channels];

//
// Setup
//

void setup(void)
{
  //
  // Print preamble
  //

  Serial.begin(57600);
  printf_begin();
  printf("\n\rRF24/examples/scanner/\n\r");

  //
  // Setup and configure rf radio
  //
}
```

```

radio.begin();
radio.setAutoAck(false);

// Get into standby mode
radio.startListening();
radio.stopListening();

// Print out header, high then low digit
int i = 0;
while ( i < num_channels )
{
    printf("%x",i>>4);
    ++i;
}
printf("\n\r");
i = 0;
while ( i < num_channels )
{
    printf("%x",i&0xf);
    ++i;
}
printf("\n\r");
}

//
// Loop
//

const int num_reps = 100;

void loop(void)
{
    // Clear measurement values
    memset(values,0,sizeof(values));

    // Scan all channels num_reps times
    int rep_counter = num_reps;
    while (rep_counter--)
    {
        int i = num_channels;
        while (i--)
        {
            // Select this channel
            radio.setChannel(i);

            // Listen for a little
            radio.startListening();
            delayMicroseconds(128);
            radio.stopListening();

            // Did we get a carrier?
            if ( radio.testCarrier() )
                ++values[i];
        }
    }

    // Print out channel measurements, clamped to a single hex digit
    int i = 0;
    while ( i < num_channels )
    {
        printf("%x",min(0xf,values[i]&0xf));
        ++i;
    }
    printf("\n\r");
}

```

```
}
```

```
// vim:ai:cin:sts=2 sw=2 ft=cpp
```