

---

## Guía Módulo de Comunicación Radio

En esta guía se realiza la configuración y programación del módulo de radio APC220 para la transmisión de datos obtenidos desde el sensor de temperatura y presión BMP280.

Enlace a lista de reproducción:

Material necesario

- 1 Arduino UNO
- Dos módulos radio APC220
- Convertidor USB to TTL
- Cables macho-hembra, macho-macho
- Sensor BMP280
- Protoboard o placa de prototipado
- Pila de 9V, batería portátil para móviles o cargador de móvil con conectar USB hembra

### 1. Introducción

En cualquier transmisión de información hay dos elementos:

- **Transmisor:** Es el elemento encargado de codificar y enviar la señal que deseamos transmitir, en nuestro caso el CanSat es el elemento transmisor
- **Receptor:** Es el elemento encargado de recibir y decodificar la señal transmitida, nuestra estación base.

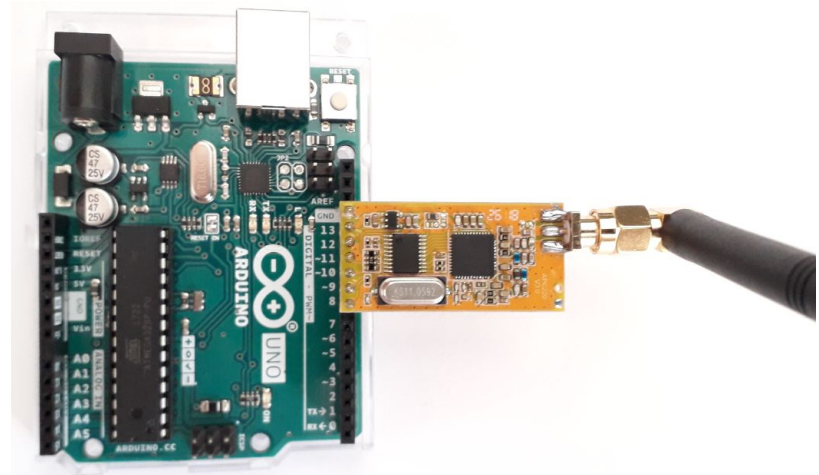
El módulo de radio APC220 permite ajustar sus parámetros, y ambos dispositivos pueden funcionar tanto de transmisor como de receptor, permitiendo una comunicación bidireccional. Sin embargo, para esta práctica sólo utilizaremos la opción de recibir datos.

## 2. Configuración

Para realizar la configuración de frecuencia de funcionamiento y otros parámetros necesarios para la comunicación, hay dos formas: a través de un programa ejecutable (RF-Magic<sup>1</sup>) y a través de una programación en Arduino. En este caso, vamos a utilizar la programación en Arduino, ya que se evitan problemas de compatibilidad y drivers. Es necesario realizar la configuración de los dos módulos por separado, es decir, subir el programa de configuración en los dos.

**Paso 1:** Conecta uno de los módulos APC220 a Arduino para configuración, siguiendo el esquema de conexión.

Arduino UNO	APC220
GND	GND
13	VCC
12	EN
11	RXD
10	TXD
9	AUX
8	SET



Esquema de conexión para configuración APC220. También se puede realizar la conexión mediante cables.

**Paso 2:** Conectar el Arduino al puerto USB de nuestro ordenador y cargar el programa de configuración. Este código nos permite leer la configuración actual y cambiar los parámetros. Copiar y pegar el programa de configuración en el IDE de Arduino. Después cargarlo y abrir el puerto serie. Programa configuración APC220

```
/*
Programa configuración APC220
Fuente: https://github.com/inopya/APC220\_Transceiver

Esquema de conexión módulo APC220 con Arduino

ARDUINO    APC220
GND  ---->  GND
13   ---->  VCC
12   ---->  EN
11   ---->  RXD
10   ---->  TXD
9    ---->  AUX
8    ---->  SET

/*

Cuando se lee la configuracion del modulo se obtiene un linea similar a esta: PARAM
415370 2 9 3 0

"PARAM AAAAAA B C D E"
```

<sup>1</sup> Para consultar cómo se realiza la configuración a través del programa RF-Magic, puedes visitar la wiki oficial del fabricante: [https://wiki.dfrobot.com/APC220\\_Radio\\_Data\\_Module\\_SKU\\_TEL0005\\_](https://wiki.dfrobot.com/APC220_Radio_Data_Module_SKU_TEL0005_)

AAAAAA, es la frecuencia de trabajo del modulo expresada en KHz, Puede oscilar entre 418MHz y 455MHz

- en el ejemplo 415.370 MHz

B, es la velocidad de transmision de radio frecuencia puede tomar los siguientes valores

- 1 (2400bps), 2 (4800bps), 3 (9600bps), 4 (19200bps)

C, es la potencia de emision, puede tomar valores entre 0 y 9, siendo 9 la mayor potencia

D, velocidad de transferencia entre el modulo y arduino o PC , toma valores entre 0 y 6: 0 (1200bps), 1 (2400bps), 2 (4800bps), 3 (9600bps), 4 (19200bps), 5 (38400bps), 6 (57600bps)

- en el ejemplo 9600bps

E, es el control de paridad de la informacion emitida por RF

- 0 (sin control de paridad), 1 (paridad par), 2 (paridad impar)

- sin control de paridad

Para grabar informacion se ha de enviar una linea similar...

WR 434000 3 9 3 0

Esta configuracion seria: Frecuencia de emision 434MHz, velocidad RF 9600, maxima potencia, Puerto serie 9600 y sin control de paridad

\*/

```
#define __VERSION__ "\Configuracion del modulo RF433 APC220 v1.0\n"
```

```
//Librerias
```

```
#include <SoftwareSerial.h>
```

```
//Declaracion de constantes
```

```
#define SET      8
```

```
#define AUX      9
```

```
#define TXD     11
```

```
#define RXD     10
```

```
#define EN      12
```

```
#define VCC     13
```

```
#define GND     GND
```

```
//definimos el puerto serie Software para comunicar con el modulo RF
SoftwareSerial APCport(RXD, TXD);
```

```
void setup()
```

```
{
```

```
  Serial.begin(9600);
```

```
  Serial.println(F(__VERSION__));
```

```
//iniciar el puerto serie software para comunicar con el APC220
```

```
  APCport.begin(9600);
```

```
  pinMode(SET, OUTPUT);
```

```
  pinMode(AUX, INPUT);
```

```
  pinMode(EN, OUTPUT);
```

```
  pinMode(VCC, OUTPUT);
```

```
  digitalWrite(SET, HIGH);
```

```
  digitalWrite(VCC, HIGH);
```

```
  digitalWrite(EN, HIGH);
```

```
  delay(1000);
```

```
  write_config();
```

```
  delay(1000);
```

```

    read_config();
    delay(5000);
}

void loop()
{
    //no hacemos nada en esta seccion
}

// ESCRIBIR CONFIGURACION

void write_config()
{
    Serial.println(F("ESTABLECIENDO NUEVA CONFIGURACION...\n"));
    digitalWrite(SET, LOW);          // poner en modo configuracion
    delay(50);

    //Parametros de configuración
    APCport.print("WR 415370 3 9 3 0");
    APCport.write(0x0D);
    APCport.write(0x0A);
    delay(100);
    digitalWrite(SET, HIGH);
}

// LEER CONFIGURACION

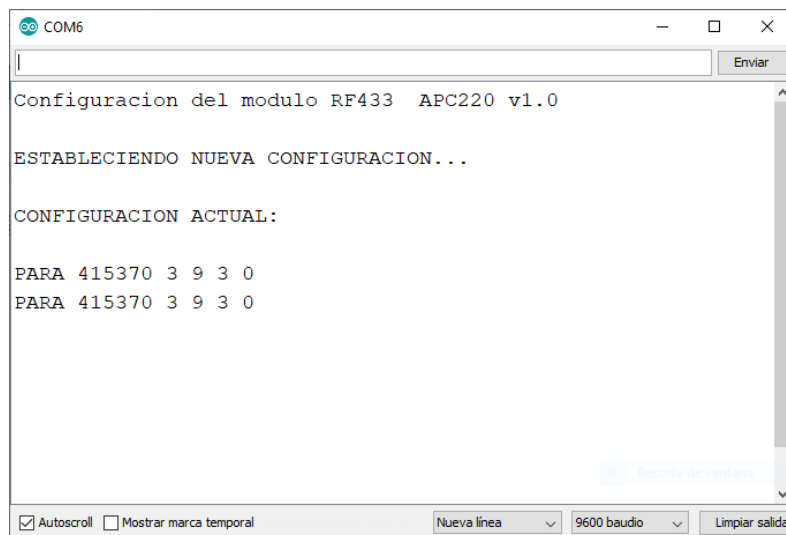
void read_config()
{
    Serial.println(F("CONFIGURACION ACTUAL:\n"));
    digitalWrite(SET, LOW);          // poner en modo configuracion
    delay(50);                      // pausa para estabilizar
    APCport.print("RD");             // peticion de datos
    APCport.write(0x0D);             // fin de linea
    APCport.write(0x0A);
    delay(100);                      // pausa para estabilizar

    while (APCport.available()) {
        Serial.write(APCport.read());
    }
    digitalWrite(SET, HIGH);         // volver al modo normal
}

//Fin del programa

```

La salida en este caso es:



---

Significado de la respuesta:

- El primer número **415370** es la frecuencia de trabajo del módulo expresada en KHz. Puede oscilar entre 418 MHz y 455 MHz
- El siguiente número **3**, es la velocidad de transmisión de radio frecuencia, puede tomar los siguientes valores: 1 (2400 bps), 2 (4800 bps), 3 (9600 bps), 4 (19200 bps)
- El tercer número **9**, es la potencia de emisión, puede tomar valores entre 0 y 9, siendo 9 la mayor potencia
- El cuarto número **3**, es la velocidad de transferencia entre el módulo y arduino o PC, toma valores entre 0 y 6: 0 (1200 bps), 1 (2400 bps), 2 (4800 bps), 3 (9600 bps), 4 (19200 bps), 5 (38400 bps), 6 (57600 bps)
- El último número **0**, es el control de paridad de la información emitida por RF: 0 (sin control de paridad), 1 (paridad par), 2 (paridad impar).

Cambiando estos parámetros en la línea `APCport.print("WR 415370 3 9 3 0")`, podemos cambiar la configuración de nuestro módulo de radio.

```
101 //Parametros de configuración
102 APCport.print("WR 415370 3 9 3 0");
103 APCport.write(0x0D);
104 APCport.write(0x0A);
105 delay(100);
106 digitalWrite(SET, HIGH);
107 }
```

**Paso 3:** Desconectar el módulo configurado y conectar el segundo módulo, realizando exactamente el mismo proceso.

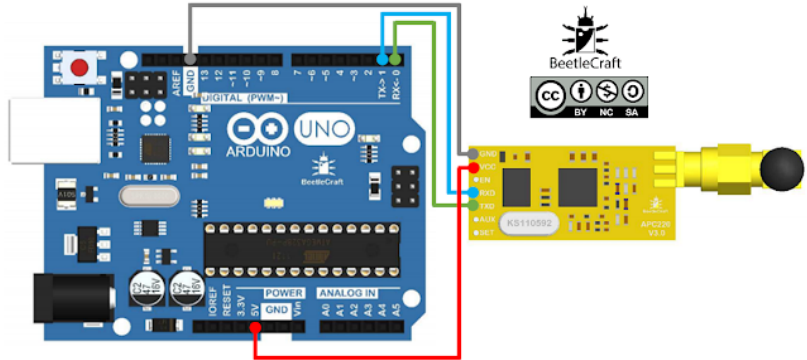
Los dos módulos, transmisor y receptor, deben tener la misma configuración para funcionar adecuadamente.

### 3. Test de Funcionamiento

Una vez los dos módulos están configurados a la **misma frecuencia y mismos parámetros**, uno se conecta al Arduino y el otro a nuestro ordenador a través del convertor USB.

**Paso 1:** Conexión módulo transmisor (Arduino):

Arduino UNO	APC220 Emisor
GND	GND
5V	VCC
TX	RXD
RX	TXD

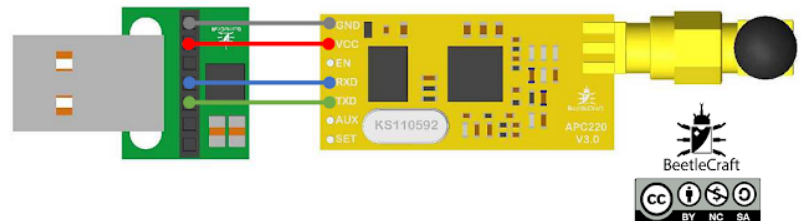


Esquema de conexión módulo transmisor en Arduino.

fuelle: <https://beetlecraft.blogspot.com/2015/10/tutorial-apc220.html>

**Paso 2:** Conexión módulo **receptor** (convertor USB):

Convertor USB	APC220 Receptor
GND	GND
VCC	VCC
TX	RXD
RX	TXD



Esquema de conexión módulo receptor con convertor USB.

Fuente: <https://beetlecraft.blogspot.com/2015/10/tutorial-apc220.html>

**Paso 3:** Programa de prueba.

**Nota:** Al cargar el programa en Arduino, es necesario desconectar los pines TX y RX del APC220, ya que son los mismos que se utilizan para cargar los programas. Una vez cargado el programa, se vuelve a conectar.

Cargamos el programa de test en el sistema transmisor (Arduino + APC220 transmisor). El programa de test sólo consiste en imprimir un *Hola Mundo* por el puerto serie (copiar y pegar el programa de prueba):

```
/*  
PROGRAMA DE PRUEBA: APC220  
CONEXION:  
    RXD: Arduino Pin 1  
    TXD: Arduino Pin 0  
    GND: Arduino GND  
    VCC: Arduino 5V
```

```

Autor: Renato H.
http://beetlecraft.blogspot.pe/
El siguiente programa es de uso publico, cualquier modificacion o mal uso del mismo
que pudiera ocasionar el mal funcionamiento de la plataforma de uso de la misma no es
responsabilidad del autor
*/

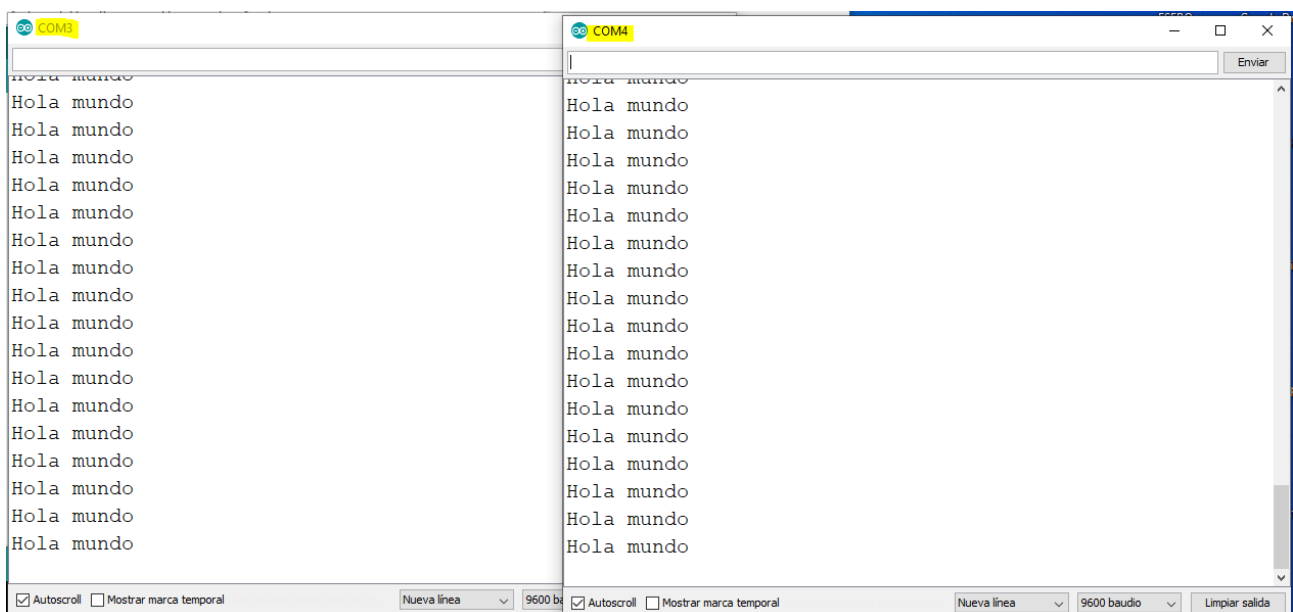
void setup(){
  Serial.begin(9600); // Velocidad de comunicacion
                      // La velocidad del puerto serial debe ser
                      // la misma que la de configuracion del modulo
}

void loop(){
  Serial.println("Hola mundo"); // Mensaje "Hola mundo"
  delay(1000);                  // Retraso de envio cada 1 segundo
}

```

#### Paso 4: Conectar módulo receptor.

Conectamos el sistema receptor (Conversor USB + APC220 receptor) a otro puerto USB de nuestro ordenador. Abrimos de nuevo el programa Arduino, desde el icono del escritorio y seleccionamos el puerto COM que aparece nuevo, **diferente al que tenemos en la placa Arduino**. Al abrir el puerto serie de nuestro transmisor y receptor, podemos ver que imprime lo mismo, es decir, nuestro receptor nos muestra el puerto serie de nuestro transmisor.



COM 4 Arduino con módulo emisor, COM 3 Módulo receptor

**Paso 5:** Conectar Arduino a una batería portátil, un cargador de móvil o una pila de 9V. Para la pila de 9V esta es la tabla de conexión realizando la conexión:

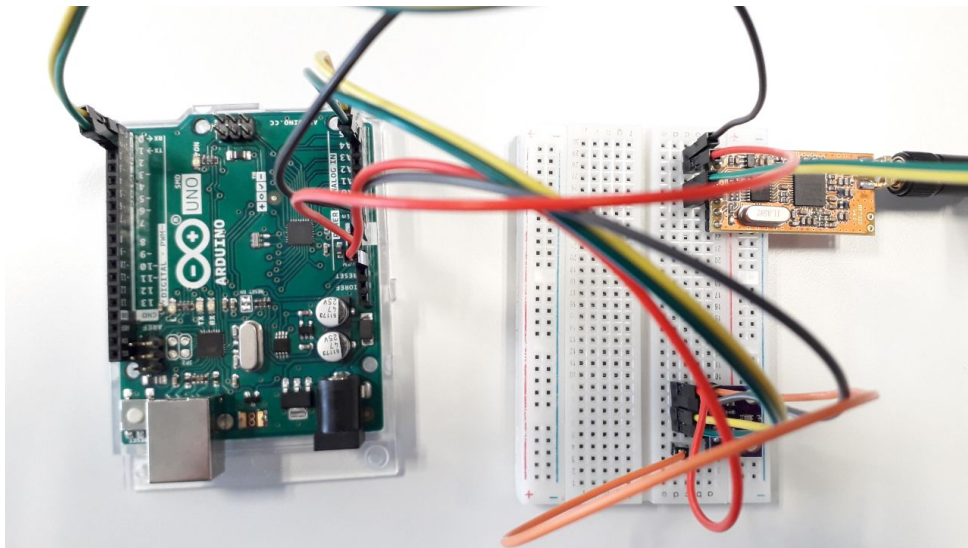
Arduino	Pila (9 V)
Vin	+ (cable rojo)
GND	- (cable negro)

---

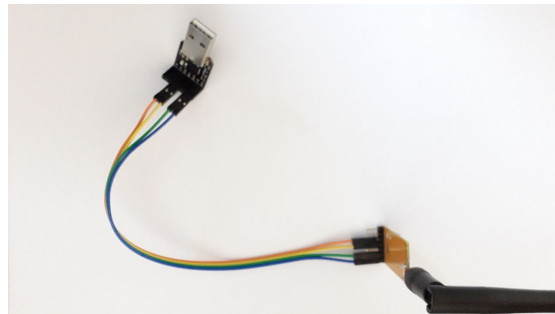
## Ejercicio 1: Datos de presión atmosférica y temperatura

Añadir al circuito el sensor BMP280, conectándolo a Arduino según lo visto en la anterior sesión y cargar la programación desarrollada en el ejercicio 3, en la que tenemos los datos ordenados por comas, incluyendo una variable paquete y el nombre de nuestro equipo.

Abrir el puerto serie del emisor y el receptor para comprobar que los datos se reciben correctamente. Conectar el sistema emisor (Arduino y módulo de radio) a una batería externa y comprobar la recepción de datos.



Circuito transmisor: BMP280 y módulo transmisor de datos



Sistema receptor

## Ejercicio Ampliación

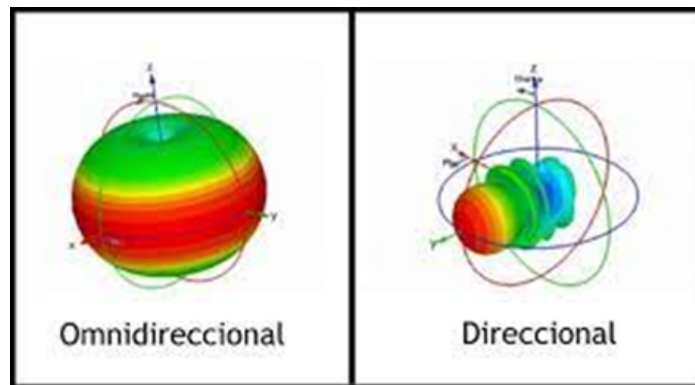
¿Es posible evitar desconectar los pines RX y TX al cargar un programa con el módulo de radio conectado? Busca información sobre la librería 'SoftwareSerial.h'.



---

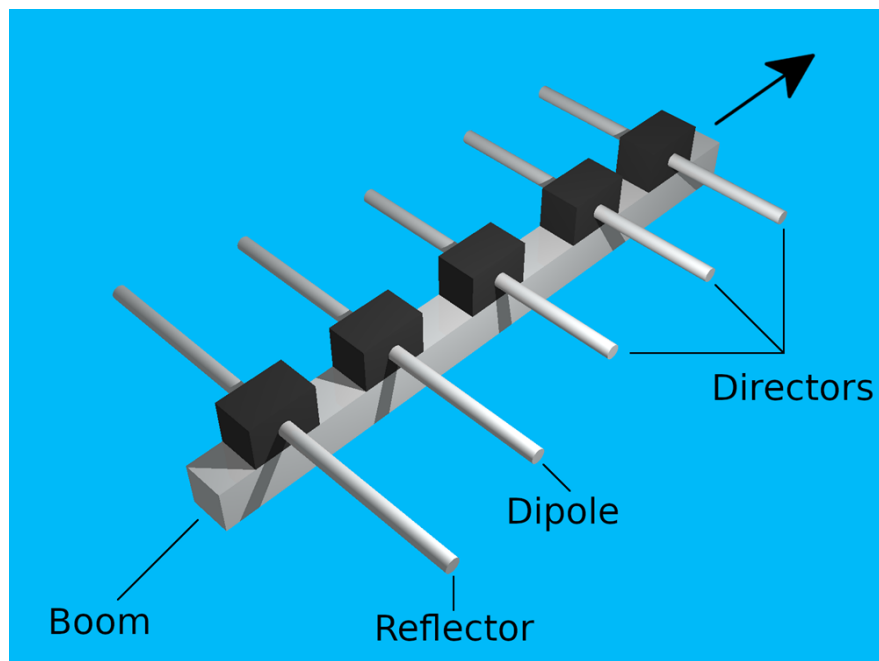
## Introducción Antena Yagi para mejorar la comunicación de datos

Las emisiones radio por defecto son omnidireccionales, por esta razón la distancia de comunicaciones puede verse reducida innecesariamente, ya que no necesitamos recibir más que desde la dirección del satélite.



Esquema omnidireccional y direccional.

Con una antena yagi conseguiremos una optimización de la potencia de escucha y transmisión de las ondas de radio. Información sobre Antenas yagi: [https://es.wikipedia.org/wiki/Antena\\_Yagi](https://es.wikipedia.org/wiki/Antena_Yagi)



Esquema Antena Yagi

### Herramientas para el para el cálculo de dimensiones de la Antena Yagi

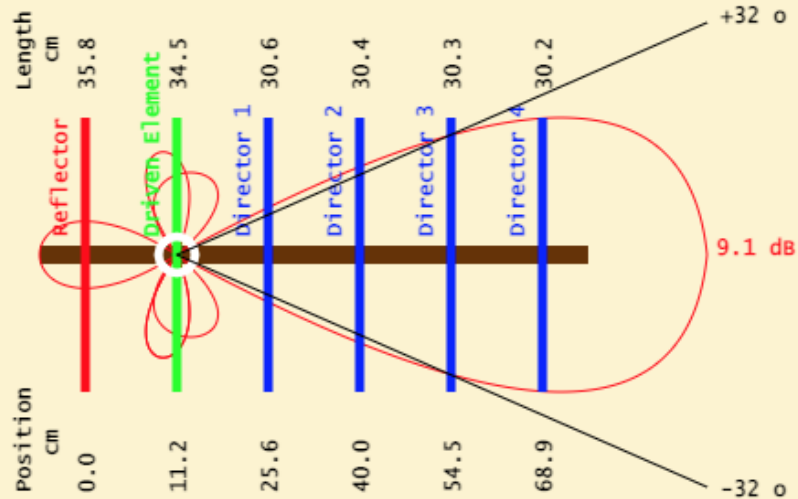
En estas páginas web se pueden realizar los cálculos de las dimensiones de nuestra Antena Yagi, introduciendo la frecuencia y el número de elementos que queremos tener.

- **Opción 1:** Yagi Antena Calculator: <https://sites.google.com/view/kn9b/yagi>

Elements **6**

Units **M, cm, mm** v

values immediately when CALCULATE is clicked. You may need to allow active content to run on your web browser. It makes some reason, the content or the functionality of the web page is blocked.



#### YAGI Antenna Solution

Reflector Element; Length = 35.8 cm, Position on Boom = 0.0 cm  
DP/Driven Element; Length = 34.5 cm, Position on Boom = 11.2 cm  
Director Number 1; Length = 30.6 cm, Position on Boom = 25.6 cm  
Director Number 2; Length = 30.4 cm, Position on Boom = 40.0 cm  
Director Number 3; Length = 30.3 cm, Position on Boom = 54.5 cm  
Director Number 4; Length = 30.2 cm, Position on Boom = 68.9 cm

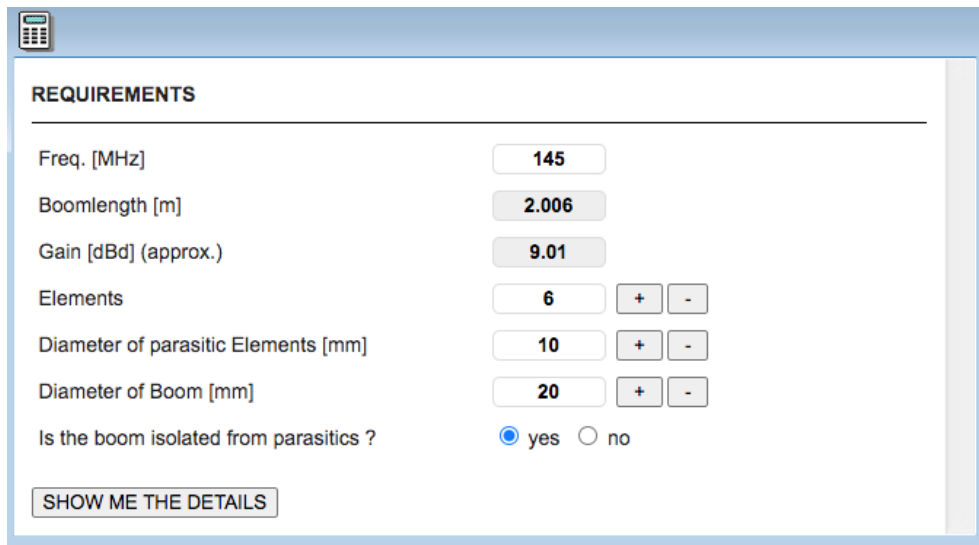
#### Key Parameters

Gain = 9.1 dB  
3 dB Beamwidth = 63 °  
One Wavelength = 72.2 cm  
Freq = 415.370 MHz  
# Elements = 6  
Length Units = cm

Antena Yagi Calculator

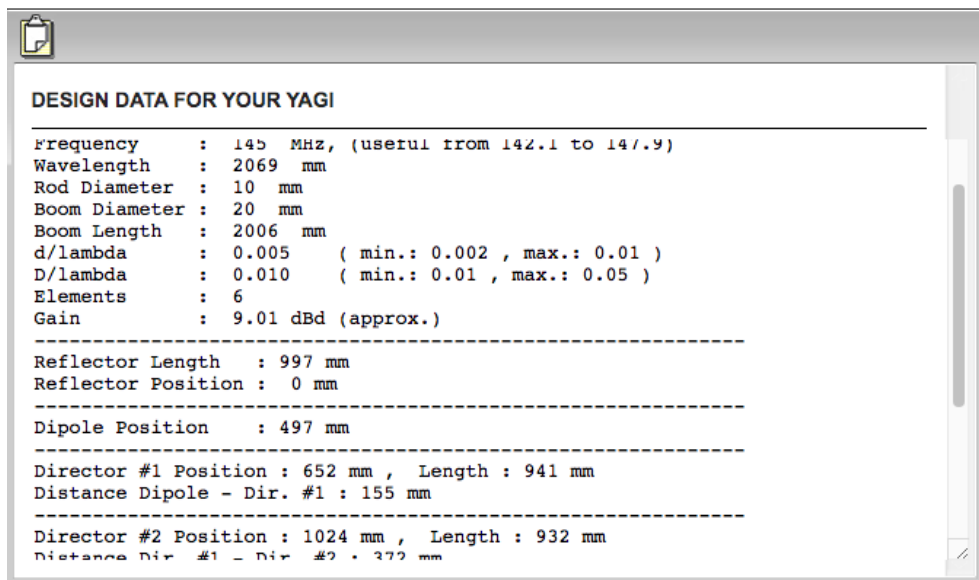
- **Opción 2:** Online calculator:

[https://www.changpuak.ch/electronics/yagi\\_uda\\_antenna\\_DL6WU.php](https://www.changpuak.ch/electronics/yagi_uda_antenna_DL6WU.php)



**REQUIREMENTS**

Freq. [MHz]	<input type="text" value="145"/>
Boomlength [m]	<input type="text" value="2.006"/>
Gain [dBd] (approx.)	<input type="text" value="9.01"/>
Elements	<input type="text" value="6"/> <input type="button" value="+"/> <input type="button" value="-"/>
Diameter of parasitic Elements [mm]	<input type="text" value="10"/> <input type="button" value="+"/> <input type="button" value="-"/>
Diameter of Boom [mm]	<input type="text" value="20"/> <input type="button" value="+"/> <input type="button" value="-"/>
Is the boom isolated from parasitics ?	<input checked="" type="radio"/> yes <input type="radio"/> no



**DESIGN DATA FOR YOUR YAGI**

```

Frequency      : 145 MHz, (usable from 142.1 to 147.9)
Wavelength    : 2069 mm
Rod Diameter   : 10 mm
Boom Diameter  : 20 mm
Boom Length    : 2006 mm
d/lambda       : 0.005 ( min.: 0.002 , max.: 0.01 )
D/lambda       : 0.010 ( min.: 0.01 , max.: 0.05 )
Elements       : 6
Gain           : 9.01 dBd (approx.)
-----
Reflector Length : 997 mm
Reflector Position : 0 mm
-----
Dipole Position  : 497 mm
-----
Director #1 Position : 652 mm , Length : 941 mm
Distance Dipole - Dir. #1 : 155 mm
-----
Director #2 Position : 1024 mm , Length : 932 mm
Distance Dir. #1 - Dir. #2 : 372 mm

```

---

## Enlaces de interés

Radio Communication

[https://esamultimedia.esa.int/docs/edu/T11\\_Radio\\_Communication.pdf](https://esamultimedia.esa.int/docs/edu/T11_Radio_Communication.pdf)

Datasheet APC220

[https://image.dfrobot.com/image/data/TEL0005/APC220\\_Datasheet.pdf](https://image.dfrobot.com/image/data/TEL0005/APC220_Datasheet.pdf)

Configuración a través del software APC220 RF-Magic:

<https://beetlecraft.blogspot.com/2015/10/tutorial-apc220.html>

Software específico APC220 RF-Magic, para realizar configuración a través de puerto:

<http://www.appcon.com.cn/en/soft.php?cid=34>

Drivers

<https://www.silabs.com/developers/usb-to-uart-bridge-vcp-drivers>

Ejemplo construcción de una antena yagi, con muy buena funcionalidad:

<https://www.youtube.com/watch?v=oxbjPXNOey4>