

Wireless Transmitter Modules allow your Arduino to wirelessly communicate with other arduinos, or with radio frequency (RF) controlled devices that operate in the same frequency (433Mhz in this case).

They work in pairs, meaning you need both a receiver and a transmitter to communicate with each other.

The receiver has 4 pins, but we actually use 3 of them: GND (Ground), VCC (5V) and one DATA pin (2 pins). You will also find an hole near left corner to solder antenna.

Please note : to achieve the best signal reception you have to add an antenna, preferably straight, 50 Ohm $\frac{1}{4}$ wavelength. Wavelength = light speed / frequency. 433M : generally 17CM, single or multi-core wire.

Technical Details:

TX (Transmitter) Technical Specifications:

Working voltage: 3V~12V

Working current: max \leq 40mA (12V), min \leq 9mA(3V)

Resonance mode: sound wave resonance (SAW)

Modulation mode: ASK /OOK

Working frequency: 315MHz-433.92MHz, customized frequency is available.

Transmission power: 25mW (315MHz at 12V)

Frequency error: +150kHz (max)

Velocity: \leq 10Kbps

Self-owned codes: negative

RX (Receiver) Technical Specifications:

Working voltage: 5.0VDC +0.5V

Working current: \leq 5.5mA (5.0VDC)

Working principle: single chip superregeneration receiving

Working method: OOK/ASK

Working frequency: 315MHz-433.92MHz, customized frequency is available.

Bandwidth: 2MHz (315MHz, having result from testing at lowing the sensitivity 3dBm)

Sensitivity: excel -105 dBm (50 Ω)

Transmitting velocity: <9.6 Kbps (at 315MHz and -95 dBm)

Example of Use:

Materials:

2 Arduini

1 Receiver

1 Transmitter

6 Jumper Wires

Wiring Instructions:

On the Receiver wire the pin labeled "DATA" to pin 2, "GND" to GND, and "VCC" to 5V on the Arduino

On the Transmitter wire one of the pins labeled "DATA" (it doesn't matter which one, as long as you use only one of the "DATA" pins) to pin 10, "GND" to GND, and "VCC" to 5V on the second Arduino.

Sketch Instructions:

This code depends on the RC-Switch library, which can be downloaded here:

<https://code.google.com/p/rc-switch/>

After copying and uploading this code your Transmitter module should be sending wireless signals. After copying and uploading this code your Receiver module should be receiving the wireless signals your Transmitter is sending. If you want to modify the signals, simply change the line:

```
mySwitch.send("10000100011000000000010100"); //Replace the number inside
```

Transmitter Code

```
#include <RCSwitch.h>

RCSwitch mySwitch = RCSwitch();

void setup() {
  mySwitch.enableTransmit(10); // Using Pin #10
}

void loop() {
  mySwitch.send("10000100011000000000010100");
  delay(1000);
}
```

Receiver Code

```
/*
  Simple example for receiving

  http://code.google.com/p/rc-switch/

  Need help? http://forum.ardumote.com
*/

#include <RCSwitch.h>

RCSwitch mySwitch = RCSwitch();

void setup() {
  Serial.begin(9600);
  mySwitch.enableReceive(0); // Receiver on inerrupt 0 =>
  that is pin #2
}

void loop() {
  if (mySwitch.available()) {

    int value = mySwitch.getReceivedValue();

    if (value == 0) {
      Serial.print("Unknown encoding");
    } else {
      Serial.print("Received ");
      Serial.print( mySwitch.getReceivedValue() );
      Serial.print(" / ");
      Serial.print( mySwitch.getReceivedBitlength() );
      Serial.print("bit ");
      Serial.print("Protocol: ");
      Serial.println( mySwitch.getReceivedProtocol() );
    }

    mySwitch.resetAvailable();
  }
}
```

Open the serial monitor, with the Receiver board connected to the computer, and you should see some signals

Resources & Related Links:

<https://code.google.com/p/rc-switch/>