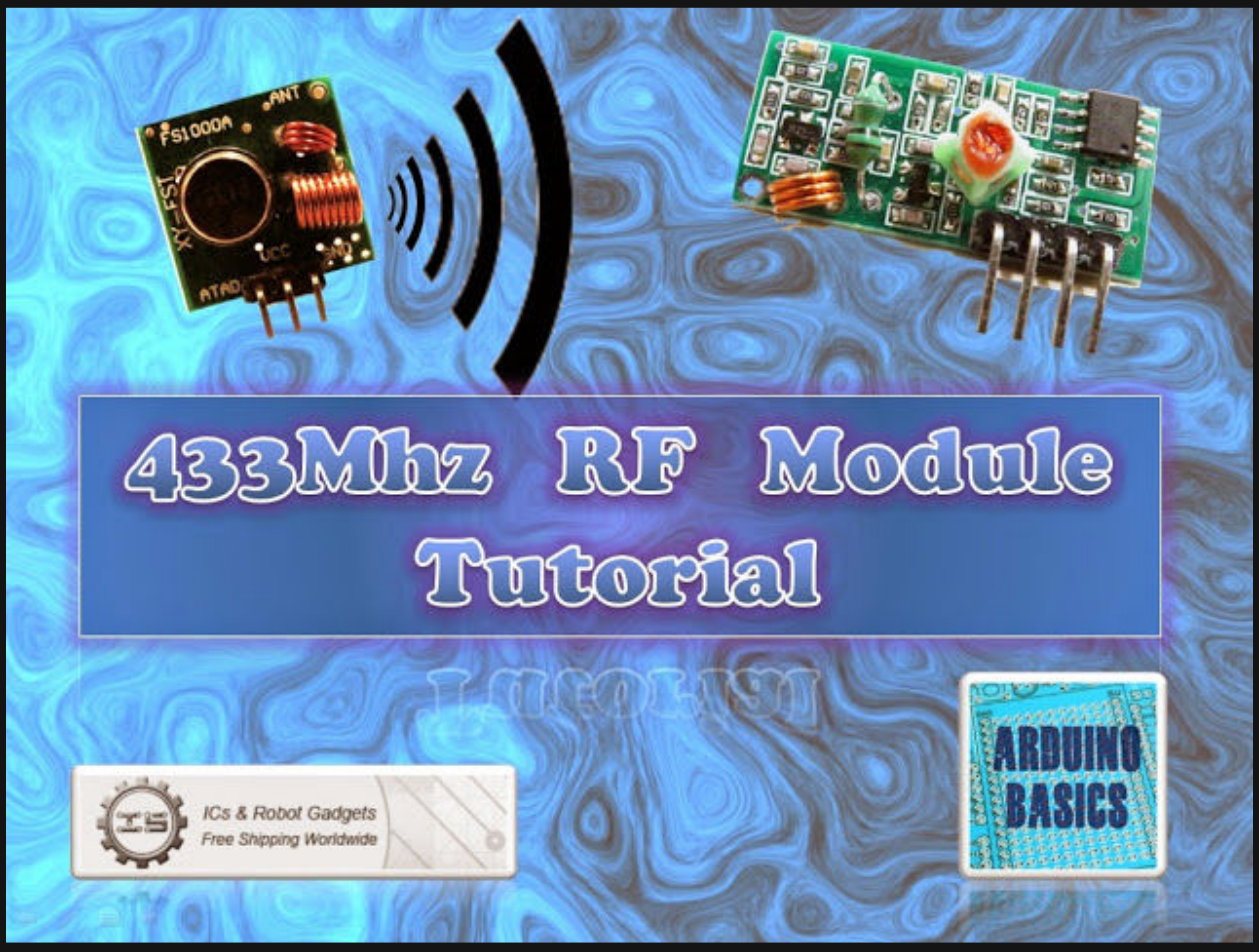
23 June 2014

# 433 MHz RF module with Arduino Tutorial 1




If you are looking for a way to communicate between Arduinos, but don't have much cash at your disposal, then look no further. These RF modules are not only affordable, but easy to use. They are much easier to set up than an XBee, plus you can use them without the need of a special shield. Before you rush out and buy a ton of these modules, make sure that you are not breaking any radio transmission laws in your country. Do your research, and buy them only if you are allowed to use them in your area. There are a few [OPTIONAL] libraries that can be used to help you and your particular project.

- Virtual Wire (at ICStation)
- RadioHead - which supercedes VirtualWire
- RC-Switch for communication with remote controls
- Ninjablocks 433 Utilities

I will mention at this point however, that I did NOT use any libraries in this particular tutorial. That's right. I will show how easy it is to transmit data from one arduino to another using these RF modules WITHOUT libraries.

Also if you are looking for an easy way to record the signals and play them back without a computer - then jump to this tutorial.

# Video

# Project 1- RF Blink

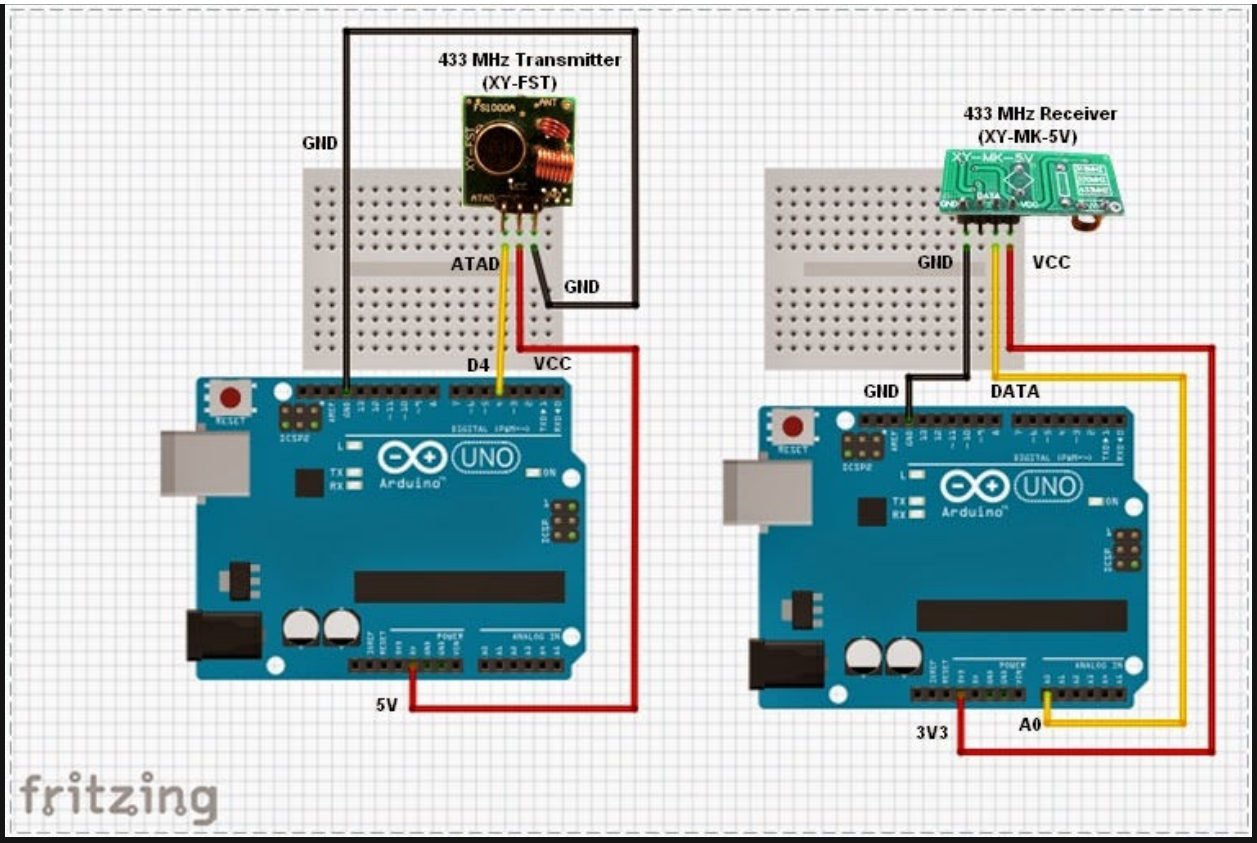
Firstly we need to test if the RF modules are working. So we will design a very simple transmit and receive sketch to test their functionality. We will use the Arduino's onboard LED to show when the transmitter is transmitting, and when the other Arduino is receiving. There will be a slight delay between the two Arduinos. You can solder an antenna onto these modules, however I did not do this, I just kept the modules close together (1-2cm apart). I also found that I was getting better accuracy when I used 3V instead of 5V to power the receiver. While using 5V for VCC on the receiver, I would get a lot of interference, however with 3V, I hardly got any noise. If you find you are getting unpredictable results, I would suggest you switch to 3V on the receiver and move the transmitter and receiver modules right next to each other. Remember this is just a check... you can experiment with an antenna or a greater distance afterwards.

Here are the parts that you will need to carry out this project:

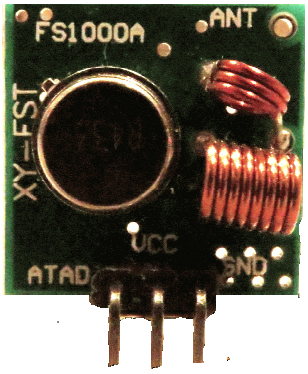
# Parts Required

- 2 x Arduino UNO or compatible boards
- Breadboard
- Wires
- RF Module (433 Mhz) - Transmitter and Receiver pair or the 315 Mhz version

# The Transmitter and Receiver Fritzing Sketch




## The Transmitter




| RF Transmitter FS1000A / XY-FST | Arduino UNO (or compatible) |
|---|---|
| ATAD | Digital Pin 4 |
| VCC | 5V |
| GND | GND |

Notice the pin called "ATAD". It took me a while to figure out what ATAD stood for, when I suddenly realised that this was just a word reversed. It should be DATA (not ATAD). Nevertheless, this is the pin responsible for transmitting the signal. We will make the Arduino's onboard LED illuminate when the transmitter pin is HIGH, and go off when LOW as described in the following table.

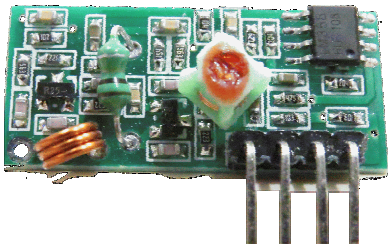
| Cycle | High | Low |
|---|---|---|
| 1 | 2000 | 4000 |
| 2 | 2000 | 2667 |
| 3 | 2000 | 1778 |
| 4 | 2000 | 1185 |
| 5 | 2000 | 790 |
| 6 | 2000 | 527 |
| 7 | 2000 | 351 |
| 8 | 2000 | 234 |
| 9 | 2000 | 156 |
| 10 | 2000 | 104 |
| 11 | 2000 | 69 |
| 12 | 2000 | 46 |
| 13 | 2000 | 31 |
| 14 | 2000 | 21 |
| 15 | 2000 | 14 |
| 16 | 2000 | 9 |
| 17 | 2000 | 6 |
| 18 | 2000 | 4 |

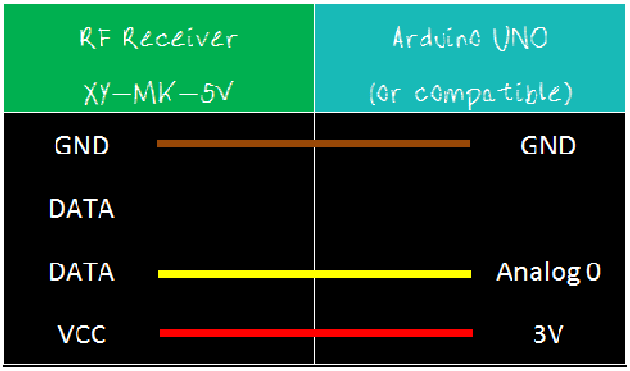
## Arduino sketch - Transmitter

```
/*
 RF Blink - Transmit sketch
    Written by ScottC 17 Jun 2014
    Arduino IDE version 1.0.5
    Website: http://arduinobasics.blogspot.com
    Transmitter: FS1000A/XY-FST
    Description: A simple sketch used to test RF transmission.
------------------------------------------------------------ */

#define rfTransmitPin 4  //RF Transmitter pin = digital pin 4
#define ledPin 13        //Onboard LED = digital pin 13

void setup(){
  pinMode(rfTransmitPin, OUTPUT);
  pinMode(ledPin, OUTPUT);
}

```

```
18 void loop(){
19   for(int i=4000; i>5; i=i-(i/3)){
20     digitalWrite(rfTransmitPin, HIGH);      //Transmit a HIGH signal
21     digitalWrite(ledPin, HIGH);             //Turn the LED on
22     delay(2000);                            //Wait for 1 second
23
24     digitalWrite(rfTransmitPin,LOW);       //Transmit a LOW signal
25     digitalWrite(ledPin, LOW);              //Turn the LED off
26     delay(i);                              //Variable delay
27   }
28 }
29
```

## The Receiver





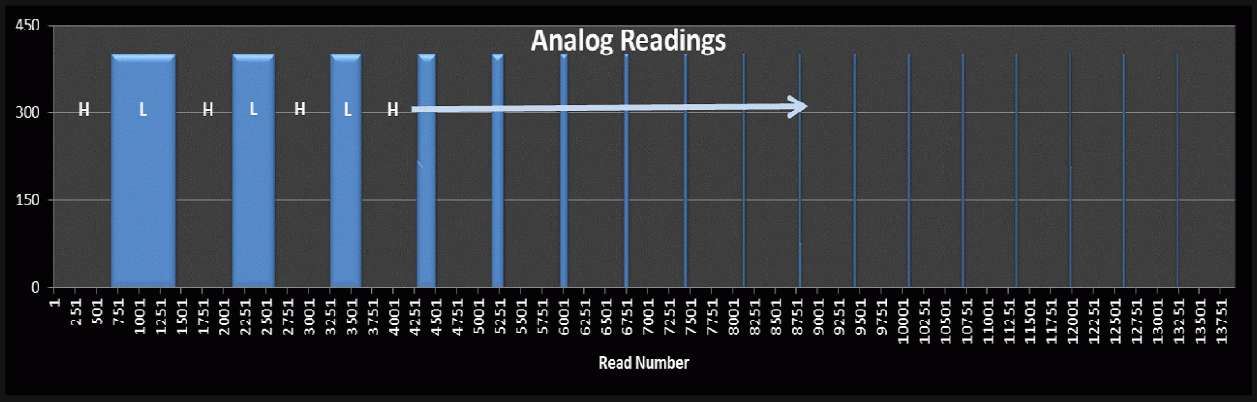

## Arduino sketch - Receiver

```
1 /*
```

```
 RF Blink - Receiver sketch
    Written by ScottC 17 Jun 2014
    Arduino IDE version 1.0.5
    Website: http://arduinobasics.blogspot.com
    Receiver: XY-MK-5V
    Description: A simple sketch used to test RF transmission/receiver.
------------------------------------------------------------ */

#define rfReceivePin A0  //RF Receiver pin = Analog pin 0
#define ledPin 13        //Onboard LED = digital pin 13

unsigned int data = 0;   // variable used to store received data
const unsigned int upperThreshold = 70;  //upper threshold value
const unsigned int lowerThreshold = 50;  //lower threshold value

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

void loop(){
  data=analogRead(rfReceivePin);    //listen for data on Analog pin 0

   if(data>upperThreshold){
    digitalWrite(ledPin, LOW);   //If a LOW signal is received, turn LED OFF
    Serial.println(data);
  }

  if(data<lowerThreshold){
    digitalWrite(ledPin, HIGH);   //If a HIGH signal is received, turn LED ON
    Serial.println(data);
```

```
33      }
34  }
```

When a HIGH signal is transmitted to the other Arduino. It will produce an AnalogRead = 0.
When a LOW signal is transmitted, it will produce an AnalogRead = 400.
This may vary depending on on your module, and voltage used.
The signals received can be viewed using the Serial Monitor, and can be copied into a spreadsheet to create a chart like this:




You will notice that the HIGH signal (H) is constant, whereas the LOW signal (L) is getting smaller with each cycle. I am not sure why the HIGH signal produces a Analog reading of "0". I would have thought it would have been the other way around. But you can see from the results that a HIGH signal produces a 0 result and a LOW signal produces a value of 400 (roughly).

# Tutorial 2

In tutorial 2, we will receive and display a signal from a Mercator RF Remote Controller for Fan/Light.

# Tutorial 3

In tutorial 3 - we use the signal acquired from tutorial 2, and transmit the signal to the fan/light to turn the light on and off.

# Tutorial 4

In tutorial 4 - we use the information gathered in the first 3 tutorials and do away with the need for a computer. We will listen for a signal, store the signal, and then play it back by pressing a button. Similar to a universal remote ! No libraries, no sound cards, no computer. Just record signal and play it back. Awesome !!

If you like this page, please do me a favour and show your appreciation :

Visit my ArduinoBasics Google + page.
Follow me on Twitter by looking for ScottC @ArduinoBasics.
Have a look at my videos on my YouTube channel.


However, if you do not have a google profile...
Feel free to share this page with your friends in any way you see fit.



Posted by Scott C at 12:47
Email ThisBlogThis!Share to TwitterShare to FacebookShare to Pinterest
Labels: 433Mhz, Arduino, ArduinoBasics, best arduino blog, communication, project, RF, Serial, tutorial

## 84 comments:

1.

**Rafael**21 July 2014 at 18:56

Great!

Reply

Replies

1.

**Rob**31 May 2015 at 01:45

Nice!!!
One question....
Say I wanted to boost the power on the Transmitter so the wave can penetrate a wall or some soil (like a planter box) how would I do it?

Is it possible?
Alright, I suppose that's two questions. One more....
Is the frequency of 433Mhz too high of a frequency for Soil penetration? Should it be lower?
Thanks for the help, and your time Scott.

2. 

**Scott C**31 May 2015 at 17:24

Hi Rob,
Your questions are way above my level of expertise.... I am by no means an expert in this field... However, I was able to transmit to the fan/light from another room. And from my quick search on google, I would say that the frequency is too high. You would have better success with lower frequencies. As for transmitter power, you would see what voltage you could apply. I found that a greater voltage gave me greater distance, however, you would have to look at the data sheet to see just how high you can go. Not sure why you need to penetrate the soil, would it be possible to bring the RF module up/out into open air?

3. 

**Rob**11 June 2015 at 10:02

Hi Scott,
Thanks for the reply!
I managed to connect a Home Made Yagi to the Receiver Module and received good results. The Yagi is built for a 433MHz Band, and is a Directional Antenna opposed to an Omnidirectional, so the energy transmitted/ received is directed.
The Gain is 7.1DB, so the Band is still wide but still focused, this way more distance can be gained with lower Transmit Power.
So far I've learned lots from this project, I have more notes on my website if interested.
Thanks for the help Scott.
let me know if you want the link.

4. 

**Scott C**11 June 2015 at 10:15

Hi Rob,
That is great news.
Yes I would love to see your project.
Feel free to post a link below

5. 

**Anonymous**21 June 2015 at 10:58

Here it is! http://www.whatisacnc.com/pages/Antenna-Radio.PHP

6. 

**Scott C**21 June 2015 at 17:10

That is cool. I could not tell if you managed to get it to find gold or silver ? I could not tell from the results at the end? Did it work ? Anyway - I thought the project concept and idea was brilliant. If I get time one day - I might try to replicate your project myself.

Thanks for posting your project link above.

**Reply**

2. 

**jawabiscuit**26 July 2014 at 08:26

thankyou for breaking it down so simply!

Reply

Replies

1. 

**jawabiscuit**26 July 2014 at 08:32

I got it working on trinket transmitter and arduino receiver exactly as you laid it out here.

**Reply**

3. 

**Fazlul Hoque**16 September 2014 at 14:48

RF tech dose not working with Tx and Rx ?

Reply

Replies

1. 

**Scott C**16 September 2014 at 16:05

Hi Fazlul,
Am not sure I understand your question.

**Reply**

4. 

**Anonymous**1 October 2014 at 22:38

why does the receiver have 2 data pins?

Reply

Replies

1. 

**Scott C**2 October 2014 at 17:38

I have no idea.

**Reply**

5. 

**youngota**3 October 2014 at 01:37

Hei, thanks for your nice tutorial. I was initially thinking about bluetooth communication between two Arduinos. How is it different when you use 433mhz rf module? (I know that you don't kneed a shield...)

Reply

Replies

1. 

**Scott C**4 October 2014 at 08:54

433mHz modules can have a longer range if you choose to use a wire antenna. I would say that using RF is easier than Bluetooth, however it depends on what you are trying to do. If you are trying to communicate between one Arduino and another, then I would personally choose the RF option. If you want to control an Arduino with your phone,

then I would choose Bluetooth. It really depends on what you are trying to do and the maximum distance required between devices.

2. 

**Anonymous**12 October 2014 at 23:56

Hi Scott,

thanks for the tutorial.

I tried things out but get some results I would not expect.

On my receiver the LED is constantly on and only goes off for a very short period of time (20-21 values in the serial monitor) when the transmitter starts to send.

I tried different RF modules and arduino boards but it is always the same behaviour. The transmitter and the receiver are very close to each other (about 1 cm).

My frequency counter (from gooit) shows a frequency of about 433.9MHz during the hole sending time of the transmitter and the LED is constantly on while the transmitter sends and constantly off when the transmitter does not send. So the transmitter should work well.

Do you have any idea what the problem could be.

Thanks and kind regard
Michael

3. 

**Scott C**11 November 2014 at 07:36

Check to make sure you are not near an interfering source. Keep the receiver away from other electronic items. You may want to try a transmitter and receiver of a different frequency (e.g. 315 MHz) - depending on where you are in the world, I think the 315Mhz frequency is more common in the USA...

**Reply**

6. 

**Mark Cassarly**7 November 2014 at 21:54

Have you tried changing the LOW output time to a constant, like the HIGH value is? rather than using i?

Reply

Replies

1. 

**Scott C**8 November 2014 at 08:27

Yes I tried - however, I designed the variable LOW output time to easily differentiate the two signals being received - that's all.

**Reply**

7. 

**Marc Quinton**28 November 2014 at 01:43

an idea about "DATA" vs "ATAD" : one is reverse of the other. Perhaps, this is because 1 level is UP (for DATA) and DOWN (for ATAD module).

Reply

Replies

1. 

**Scott C**1 December 2014 at 06:55

I am not sure what you mean ?

**Reply**

8. 

**Pances**2 December 2014 at 15:40

The tutorial very good...
But, how with use KYL 210
Please help me, i need coding and tutorial...
Thanks Master

Reply

Replies

1. 

**Scott C**2 December 2014 at 17:54

Hi Pances,
I don't have that module.
Best to ask that question in the Arduino Forums. Perhaps someone else has already gone through the process.

**Reply**

9. 

**Peter H** 16 December 2014 at 07:08

Hi Scott,
i have a 17 dof robot controlled by a torobot usc32 servo controller, running the torobot software through a usb cable. i have two arduino megas and the FS1000A tran/rec. i have fitted one mega to the robot servo controller,arduino tx to controller rx, a common gnd plus power to each, could you show me a mod to your sketch to run the servos through the FS1000A tran/rec. cutting the usb cable out of the picture. thanks Scott,
regards Peter.

Reply

Replies

1. 

**Scott C** 28 December 2014 at 00:14

Hi Peter,
Thank you for your question, but these requests are best served on an Arduino forum. I get many requests for help, and either don't have the required experience or the time. You will find the forums much more useful.

Regards
Scott

**Reply**

10. 

**Reiner christian raja** 31 December 2014 at 14:23

hi i understood what u have posted, and i need to use many reciever with one transmitter with the same arduino, how can i do this, explain

Reply

Replies

1. 

**Scott C**4 January 2015 at 22:49

It depends on what you are trying to do? And why you would use many receivers on the same Arduino... This does not really make sense to me??

**Reply**

11. 

**Samuel Ecks**7 January 2015 at 21:41

Hi Scott, a very clear tutorial. I understand you are tight for time, but I wondered if you thought it was possible for the LED connected to the receiver to get brighter after each signal from the transmitter?

Reply

Replies

1. 

**Scott C**8 January 2015 at 13:39

Yes - that would be possible. When the signal goes above or below a certain threshold, you could make the LED get brighter. You would have to use PWM to achieve this.

**Reply**

12. 

**TEK**18 January 2015 at 08:58

Hi
Do you know if these ca be used on the 868MHz band?

Reply

13. 

**Scott C**18 January 2015 at 21:22

I am not sure. It seems like it was made specifically for the 433MHz range,and I would not know how to change it.

Reply

14. 

**Anonymous**23 January 2015 at 03:03

Very helpful tutorial Scott. Thank you for sharing!

erco/ parallax & picaxe user

Reply

Replies

1. 

**Scott C**2 February 2015 at 14:58

No worries

**Reply**

15. 

**Yoyo**27 January 2015 at 09:26

Hi.

Your tutorial is very usefull
I was using serial communication by this module, and it's failed.
Could you tell me why you use analoug input?

One request:
Could you make a tutorial for nRF24L01(+) 2.4GHz Wireless Transceiver module?
It's using SPI communication

Thanks a lot
http://yoyosukardi.blogspot.com

Reply

Replies

1. 

**Scott C**2 February 2015 at 14:58

Hi Yoyo,
I use analog input so that I have full control of the threshold value.

Thank you for your request, however, I do not have those components. But if I ever buy them, then I will look into it.

Regards
Scott

**Reply**

16. 

**Anonymous**29 January 2015 at 00:33

Nice work. Thank you for putting this together for newbies like me.

Reply

Replies

1. 

**Scott C**2 February 2015 at 15:12

No worries. Thanks for the feedback.

**Reply**

17. 

**Steven Scott**30 January 2015 at 02:51

Hi, Can the Arduino boards be removed once the programme and link is established? Will it still work? Sorry if this sounds like a strange question...

Thanks, Ste

Reply

Replies

1. 

**Scott C**2 February 2015 at 15:16

Not sure what you mean... but if you are asking if the RF modules will still work without the Arduino boards, then the answer is no.

If you are asking if the sketch will still work if you disconnect the Arduino boards from the computer, the answer is yes and no. Yes, the code will execute, but there are Serial commands which will be useless. But you could write the signals to an SD card instead, or use some other method to capture the signals ??

Not sure if this answered your question.

2.

**Steven Scott**4 February 2015 at 02:21

Scott C. Thanks for that. I was asking if it will still work with arduino removed. Question answered, thank you.

**Reply**

18.

**gc9n**31 January 2015 at 08:00

Hi there...
This is the only SKETCH for RX and TX that worked for me...

I my trying to make this work with a VirtualWire library , but there is no way to do it....this is my code..

this is a link from the code that i am trying to get it work with the XY-MK-5V 433Mhz RF Receiver AND FS1000A/XY-FST RF 433Mhz Transmitter

http://forum.arduino.cc/index.php?topic=296564.new#new

Reply

Replies

1.

**Scott C**2 February 2015 at 15:28

I have no experience with the virtual library. However, my advice would be to start with the basics. Start small, and get the basic functionality working. Then start adding complexity step by step.
Right now, you have so many places where your sketch could be failing, so go back and start with a simple sketch and then go from there.

**Reply**

19.

**roshan balachandar**6 March 2015 at 12:57

I am working on a project-sign language translator.So I am using flex sensors to gesture various alphabets.Thus I need to transmit these alphabets wirelessly and display them in another ARDUINO UNO.Could you help me with the coding part for transmission and reception ?RF module:WX-TX-01 and WS-RX-02.

Reply

Replies

1. 

**Scott C**6 March 2015 at 14:05

Hi Roshan,
I don't have those sensors, but you can post this to my forum section and I will do my best.

**Reply**

20. 

**Nabeel Fattah**12 March 2015 at 17:55

Thank you Scott for the nice and helpful tutorial.
Can we use this device for two way communication? So the reciever will send back the data to the transmitter after reiceving it.
I need to measure time interval for the data transferring between them, to see how fast and how much packet size can be send.
to know about the maximum data and speed.

Thanks

Reply

Replies

1. 

**Scott C**13 March 2015 at 01:19

Hi Nabeel.
You would need two transmitters and two receivers to have two-way communication between two arduinos. The receiver module only receives - it does not transmit.
If you want a module that transmits and receives - you will need to search for a "433Mhz transceiver".

As for the speed and packet sizes that can be sent on the modules that I have - I am afraid that is beyond my knowledge level, and you would have to research the datasheet for that one.
However, I have read that the success rate of data transmission between Arduino's can be hit and miss (especially as the distance increases). Some people get really frustrated, and spend quite some time trying to sort this out.

As you can see from my 4 tutorials on these modules, my end goal was to copy the RF signal from my RF remote. And it worked very well for this project.

Reply

21. 

**wesley eeckhoudt**14 March 2015 at 03:02

Is t possible to send a fixed code to the receiver to eliminate the noise ( so the receiver only reacts on that code) I can't figure it out . Pleaze help

Reply

Replies

1. 

**Scott C**14 March 2015 at 19:55

Hi Wesley,
I am not sure how sending a code will eliminate noise.
My guess is that the noise will still be there.

Have you looked at the VirtualWire or RadioHead library ?
http://www.instructables.com/id/RF-315433-MHz-Transmitter-receiver-Module-and-Ardu/?ALLSTEPS

http://www.airspayce.com/mikem/arduino/RadioHead/

If these do not help and you want to start from scratch, then you could look to send a code to the receiver, but I would send it multiple times and maybe with some sort of checksum method to ensure that you have received all the data. The trick is how do you let the transmitter know whether you have been successful or not in receiving it's message.

Hard to know the best way to tackle your issue without knowing your setup/layout/requirements/equipment etc etc. These problems are always best served in a forum.
If you need further help - then either post in the ArduinoBasics forum - in the section called "Help me with my project" - and I will try my best to help you when I can. But if you really want some good help, then look to the Arduino forum

Reply

22. 

**Colin**21 March 2015 at 13:44

Hi, Im doing a project using RF module to track a parked vehicle. I need some assist.

Reply

Replies

1. 

**Scott C**21 March 2015 at 22:57

Hi Colin,
Post your query in the ArduinoBasics forum in the section "help me with my project". I get a lot of requests for help, and will try my best to help you in some form or fashion. But please do not expect me to write out your code for you. You will have to do most of the work/research/coding yourself. I will give advice or guidance where I can. If this is not suitable, then I recommend the Arduino Forum.

2. 

**Scott C**22 March 2015 at 21:25

Hi Colin,
Just letting you know that I rellocated your post in the forum to the "Help me with my project section"
Regards
Scott

3. 

**Colin**23 March 2015 at 00:11

Hi, Thank you

**Reply**

23. 

**Damien Vessiere**24 March 2015 at 02:17

Hello Scott,
Thanks you for this nice tutorial and the help that you provide in comments.
I have a Raspberry Pi, 2 ATtiny85 and the same 433 Mhz reciever/emitter as you.
I can programs my 8Mhz ATtiny throught the Raspberry with arduino libs (it works fine for blink led or Serial.println("Hello") with the raspberry at 2400bps) but when I use your program or other "Virtual wire" program found on the web it not work at all...
The reciever analog value is extremely noisy (0, 100, 600, 32, 200, ...) where ever I am. And whatever I send with the emiter 1 or 0, it dosent seem to affect the reciever.
Technicaly, both reciever and emiter VCC are plug on the 5V GPIO of Raspberry. GND on Raspberry

GND and data each on a pin of one of the two Attiny85.
Do you have any idea of the potential mistake ?

And could you please tell me if it works for you if you do this scheme without microcontroller or Arduino :
Emitter VCC (5V) GND (GND) DATA (5V)
Reciever VCC (5V) GND (GND) DATA (resistor -->LED --> GND)
Led must be ON 100% of the time, right ? (in my case led blink randomly)

And
Emitter VCC (5V) GND (GND) DATA (GND)
Reciever VCC (5V) GND (GND) DATA (resistor -->LED --> GND)
Led must be OFF major part of the time (noise) , right ? (in my case led blink randomly)

Thanks :)

Reply

Replies

1. 

**Scott C**24 March 2015 at 08:48

Hi Damien,
I am not sure what you mean that when you use my sketch, that it does not work at all? What doesn't work ? Are you getting high and low signals ? How is it that you can get a blink sketch to work and not this sketch (which alternates the high and low signals like a blink sketch).
Also, when using these modules, I too experienced some interference and noise. So I reduced the voltage of the receiver to 3.3V. I think the noise issue is a common one with these modules - plus it also depends on where you live. You may need to opt for a different frequency depending on your country.
From what I have read, and from what I have noticed, is that people are trying to use them in a way that is different from what they were designed to do.
When I transmitted signals from my remote control to my receiver, I had no issues. When I transmitted that recorded signal to my fan's receiver, I had no issues. But sending a high value or low value for an extended amount of time, seems to cause issues. So rather than try to see if it is working by sending a continuous high or continuous low signal... try sending a sequence of highs and lows in quick succession, and repeat this signal a number of times.
I might try and write a tutorial to explain this better, but it won't be for a couple of months, because I am in the middle of some other projects.

2. 

**Damien Vessiere**24 March 2015 at 16:30

Thanks for your quick answer, I'm from France.
When I talk about "blink" it's just to say that I succed to program my ATtiny with the "hello world" program for a microcontroller : flash a LED on a pin...
So when I use your program my LED flash very quickly without any reason... (Same effect if I disconnect the emiter, and same effect if I connect the LED directly to the data of the receiver). So I get high and low signals but randomly.
So I will try to turn ON and OFF the emiter quickly, witch delay do you advise me, 200ms ?
I will try also to reduced the voltage of the receiver to 3.3V.

Can I do something on these modules to change frequency ?
One more important thinks... I am not using antenna because module are 5 cm far away and I am not very good with soldering iron. Do you think it could be the problem ?

3. 

**Scott C**24 March 2015 at 21:09

Ok - I understand.
Instead of modifying the frequency of the module, I meant for you to get a different frequency module to try out (eg 315Mhz or something else)... but not sure if that will help you or not.
You will get interference on the receiver due to the variable gain. You might benefit from reading this site.
However, I overlooked that you were using an ATtiny. I don't have one of them and have never used one of them. Do you have an Arduino UNO that you could use to test the RF modules ? This may eliminate some of the guess work you are doing at the moment???
And I just read this blog post from a person who might be having the same problems you are - and my guess is that you might want to reconsider your choice of microcontroller.

If you want to communicate between the two modules at a distance greater than 5cm, then yes - I would highly recommend for you to use an antenna on the receiver. My transmitter was able to transmit a signal to my fan at quite a distance without an antenna, however if I were to add an antenna, I am sure the distance would be greater.

As for the the amount of delay to choose... I would say, start low (20ms) and then increase it. If you have a potentiometer or button that you can use to modify the time in your program, it will save you the hassle of uploading and re-uploading for every change in time. The serial monitor is also very useful for this type of troubleshooting.

4. 

**Scott C**24 March 2015 at 21:14

However - you might also want to look at this instructable. They seemed to get it to work.

5. 

**Damien Vessiere**25 March 2015 at 20:04

No arduino for the moment... just a Raspberry, 2x Attiny and some cable, led, ...
Thanks you for all this stuff, I will try all this !

6. 

**Damien Vessiere**20 April 2015 at 21:53

All works fine !
Problems was :
My first ebay receiver was broken (maybe my mistake :-) )
Second, the antenna is mandatory... without, even few centimeters, it not work at all.
I have bought it here : http://r.ebay.com/zyjuge 10 for 1.5 € delivery fee included.

7. 

**Scott C**21 April 2015 at 11:08

Good to hear that you got it to work. I have heard a few people mention that their receivers did not perform, however when purchased elsewhere it worked fine (like you have just experienced)... I think there must be a quality issue in manufacturing these cheap devices.

**Reply**

24. 

**kooroosamy kavinien**28 March 2015 at 01:50

Hi scott c , I highly appreciate your video , i have used the same code but including the virtual wire library to it and i have use one arduino uno and an arduino mega the code did compile everything but at the it didn't work .
well i need a bit of help , can you guide me how i can use RFID to trigger an servo onto another arduino , your guidance would highly be appreciated.

Reply

Replies

1. 

**Scott C**28 March 2015 at 08:46

I have made my reply on the forum post you submitted.

**Reply**

25. 

**Anonymous**15 April 2015 at 01:54

Thank you for this tutorial. I have been using the virtual wire libraries, but i believe that a project should be as simple as possible without sacrificing functionality.

Reply

26. 

**bharat ahir**23 April 2015 at 13:10

if two rm module is used in same location then is there may b interfearence?

Reply

Replies

1. 

**Scott C**23 April 2015 at 20:08

Yes. So you would have to either identify the transmitter in some way or use different frequencies.

**Reply**

27. 

**Niranga Shalutha**15 July 2015 at 23:02

here is it possible to catch a signal if the distance is 10m? we are going to transmit different data signals from several transmitters? These data signals should be caught by one transmitter. one proper data signal is enough to be caught from one transmitter How should i do that? Please Help..!! Thank you

Reply

Replies

1. 

**Scott C**15 July 2015 at 23:58

Hi Niranga,
I assume you mean to catch the signal using a receiver ?
Is it possible to catch a signal if the distance is 10m? The answer is: it depends.

It depends on:
1) Frequency being used.
2) Objects/walls/obstructions or clear path
3) Antenna being used or not (on transmitter and/or receiver)
4) Signals being sent at different times or at the same time
5) Strength of the signal being transmitted (ie. power of the transmitter)

6) Length of the signal being transmitted
7) Interfering signals from undesired sources
I am sure there are other factors, but I think you get the point.

How should you do that? First try with one transmitter. If that works, then try more.
Please help: well I'm not sure if that helped, but my tutorials might. Try them and see how you go.

**Reply**

28. 

**Mitchell Currie** 3 August 2015 at 09:26

Hi,
Quick question, what's the point of the rolling LOW signal value, are you experimenting with some sort of analogue signal or just seeing how it reacts? (it seems overly complex for the simple on/off scheme I thought it required).

How do you handle the clock speed of the signal? it seems like you would need to make sure that RX and TX devices both clock bits out at the same speed?

Reply

Replies

1. 

**Scott C** 3 August 2015 at 13:46

Hi Mitchell,

This was just an experiment to test the transmitter and receiver.
Without really knowing how these modules worked, I thought that I would just send a signal using the transmitter and just listen with the receiver to see what would happen.
By sending a typical blink pattern, I noticed that when I sent a high signal, the receiver would provide an analog reading of zero, and a low signal would produce an analog reading of around 400. This seemed very anti-intuitive to me. So then I thought, maybe there is a delay, and that the high signal is actually producing a value of 400 ?? So how could I test this ??
I decided to change one variable - and keep the rest constant. I changed the duration of the low value, and kept the high value duration the same... and you can see the results in the chart above....
My end goal was to receive a signal from my RF remote (store this signal), and then play it back. I managed to do it in the end, but you have to read the other tutorials to see how.

**Reply**

29. 

**Anonymous**5 August 2015 at 02:51

Hi,
Thanks for the sharing!

I've followed your idea exactly, however the receiving LED is flashing when the transmitter LED is off, Do you know how can I avoid it as it's sign of noise.

Suli

Cheers!

Reply

Replies

1. 

**Scott C**5 August 2015 at 11:45

Hi Suli,
You may get some noise with this sketch. The Receiver will increase it's gain automatically until it receives a signal. So to reduce noise, you should transmit at a faster rate or perhaps change your location and move away from interfering signals.

**Reply**

30. 

**Anonymous**5 August 2015 at 23:16

Hello! Thanks for your tutorial!
I´m kind new with arduinos, but I was thinking of using just one arduino with the transmitter, and none with the receiver. The receiver would simply connect to a led, and depending on the information that arrived it would blink or just turn on and off, for example. Is that possible? Thanks in advance!

Reply

Replies

1. 

**Scott C**6 August 2015 at 11:09

If I only had one arduino, I would do it the other way around. I would transmit without the Arduino and attach the receiver to the Arduino. I am not sure if the receiver would be able to drive an LED... it might be able to, but I haven't tried. I guess it depends on what you are trying to transmit, and why you would want to do this?

2. 

**Anonymous**7 August 2015 at 07:33

I´m trying to communicate wirelessly a pushbutton to an electric toy. But I wanted the pushbutton to work under different modes, for example working while pressed, or starting to work when pushed, stopping x minutes later.... But I´m not sure if the receiver would receive that information if i don´t use an arduino...

3. 

**Scott C**7 August 2015 at 08:28

As you can see from my tutorial, when you apply voltage to the transmitter, the receiver will respond accordingly. If the electric toy has an RF receiver built in, then I would use the Arduino on the transmitter side. If you wanted to trigger a push button on the toy, you could possibly do all of this without an Arduino altogether, however, you may get false triggers through interference. While my tutorial makes it look like the receiver will only trigger when the transmitter sends information, this is not true. It will also trigger when it picks up stray signals from other devices or electrical equipment. I don't think these modules were designed to work in the way you imagine it to... as the receiver will increase the gain and become more sensitive to stray signals if it hasn't received anything for some time... So I personally think you would need two Arduino's for this to work properly, and you would have to send a sequence of commands which the receiving Arduino would have to interpet, decode and then act in the way you want it to.

4. 

**Anonymous**7 August 2015 at 19:33

I was just thinking of use one, to reduce costs and space in the toy.. But I understand what you´re saying. Thank you very much for your answer!

**Reply**

31. 

**Kadir Haldenbilen**16 August 2015 at 16:22

Thanks Scott, for such a simple solution. I have noticed that It works better with: else if(data<lowerThreshold) { ... rather than if(data<lowerThreshold) { ...

for the receiving end for lower threshold check, eliminating some noise.

Reply

Replies

1. 

**Scott C**17 August 2015 at 10:15

Hi Kadir,

Thanks for the tip, however, can you explain how your modification would change the outcome?
For example, if data=0, both statements would trigger
If data=40, neither statement would trigger...

If I were looking to eliminate noise, I would modify the range of the upper and lower Threshold values, but I can't see how your modification changes anything?? Maybe I am missing something? But I am interested in what you have to say.

**Reply**

32. 

**Sergio Cantoro**10 September 2015 at 22:00

Hello , are these rf modules compatible with Arduino 2 (3,3V) or UDOO? Thank you

Reply

Replies

1. 

**Scott C**10 September 2015 at 23:52

Hi Sergio, As per the manufacturer website, it appears that it needs at least 3.5V, but normal operating voltage is 5V.
Having said that, I found that the receiver worked fine at 3.3V, the transmitter may be a different story.

**Reply**

33. 

**Niranga Shalutha**17 September 2015 at 12:14

Can you please tell me if we use multiple transmitters to transmit a signal to a single receiver can there be any interference or data loss? can you give me an idea about how many channels are in transmitters and how they work in an above mentioned scenario ?

Reply

Replies

1. 

**Scott C**17 September 2015 at 17:49

Hi Niranga,

The receivers are indiscriminate. Meaning -> If you send a signal at that frequency, it will receive it. It will receive stray signals from other sources also...
So if you you are using multiple transmitters, you will need to ensure that they are not transmitting at the same time or are using different frequencies..... similar to a walkie-talkie. If you are transmitting at 2 different frequencies, you will need 2 matching receivers.

Can there be data loss - yes - of course... there can be data loss even with one transmitter. Will it get worse with two ? Depends on whether the signals over-lap or not.

2. 

**Niranga Shalutha**18 September 2015 at 14:04

Thank you very much for the information. This means a lot to me
i also want to know that is there a specific number of channels in the transmitter. I saw somewhere that there are 4 channels. if there are channels. so transmitters can use different channels to transmit. Am i right ? (I'm not sure)
So if i transmit the signal with different delays in different transmitters. Will it solve the problem ?

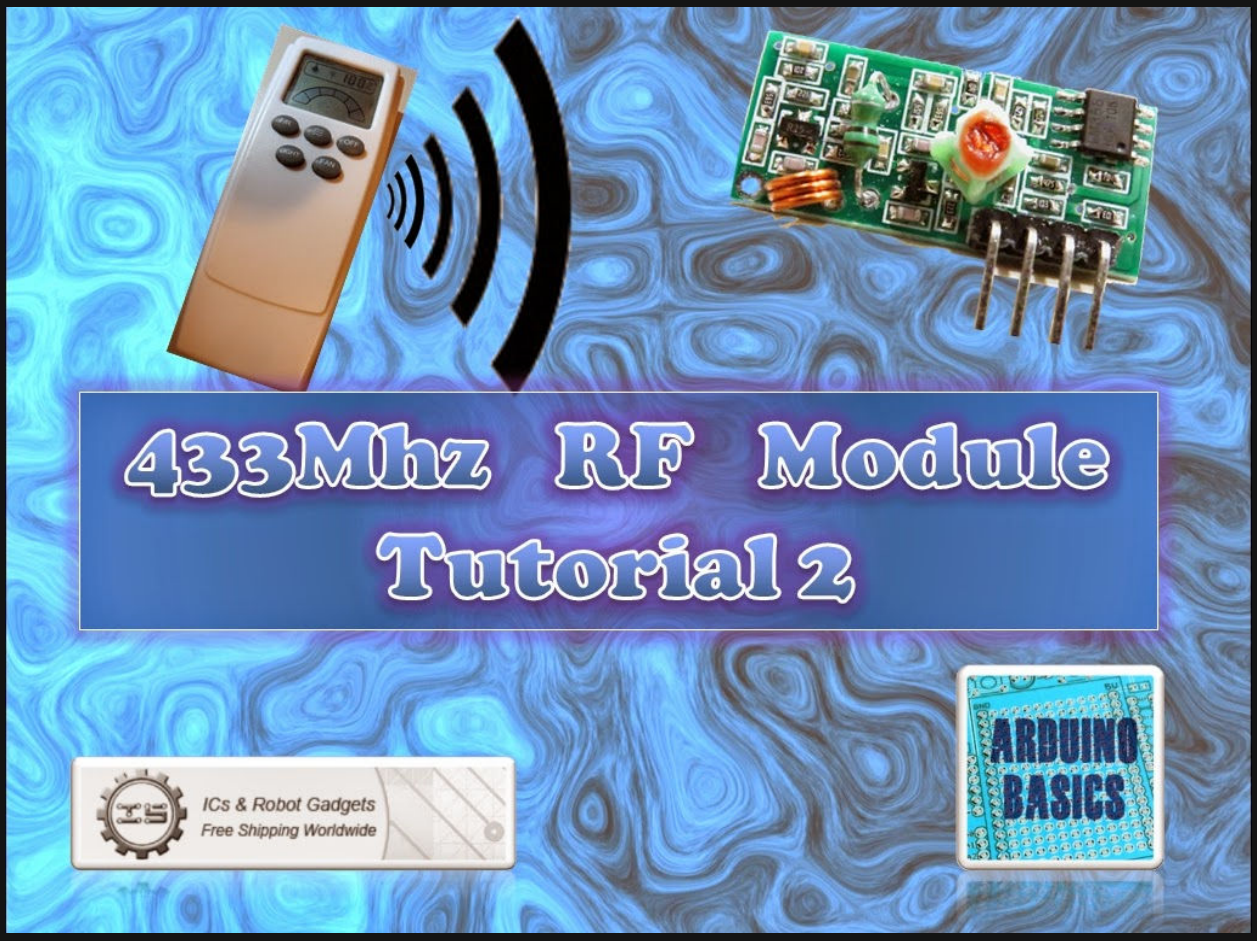
3. 

**Scott C**19 September 2015 at 09:15

I don't know anything about the number of channels. You would have to ask the site where you read that... I am guessing there would some additional hardware required, but that is just a guess. With regard to the delays, I am not sure what you are doing or how you are hooking these transmitters up? But as long as each transmitter transmits their complete signal un-interrupted by any other transmitter/signal you should be fine. The way that you choose to do this will depend on your particular set-up and personal preference.

27 June 2014

# 433 MHz RF module with Arduino Tutorial 2




## Project 2: RF Remote Copy

In the previous project, we transmitted a signal wirelessly from one Arduino to another. It was there to help troubleshoot communication between the modules. It was important to start with a very short distance (1-2 cm) and then move the RF modules further apart to test the range. The range can be extended by soldering an antenna to the module, or by experimenting with different voltage supplies to the modules (making sure to keep within the voltage limits of the modules.)

In this project - we aim to receive a signal from an RF remote. The remote that I am using is a Mercator Remote Controller for a Fan/Light. (Remote controller code is FRM94). It is important that you use a remote that transmits at the same frequency as your receiver. In this case, my remote just happens to use a frequency of 433MHz. I was able to receive RF signals from from a distance of about 30cm without an antenna (from my remote to the receiver).

## Video

Here are the parts that you will need to carry out this project:

## Parts Required

- 1 x Arduino UNO or compatible board
- Breadboard
- Wires

- Mercator FRM94 Remote Controller for Fan/Light (Transmitter Model: TR107A)
- RF Module (433 Mhz) - Receiver or the 315 Mhz version

## Remote Controller

You can quickly test your remote, by pressing one of the buttons in close proximity to the RF receiver (using the same sketch as in Project 1), and you should see the LED flicker on an off in response to the button press. If you don't see the LED flickering, then this project will not work for you.

Here is a picture of the remote controller that I am using:


## Arduino Sketch - Remote Receiver

The following sketch will make the Arduino wait until a signal is detected from the remote (or other 433 MHz RF device). Once triggered, it will turn the LED ON, and start to collect and store the signal data into an array.

I did my best to keep the signal reading section of the sketch free from other functions or interruptions.The aim is to get the Arduino to focus on reading ONLY... and once the reading phase is complete, it will report the signal data to the Serial monitor. So you will need to have the Serial monitor open when you press the remote control button.

The remote control signal will be made up of HIGH and LOW signals - which I will try to illustrate later in the tutorial. But for now, all you need to know is that the Signal will alternate between HIGH and LOW signals, and that they can be different lengths.

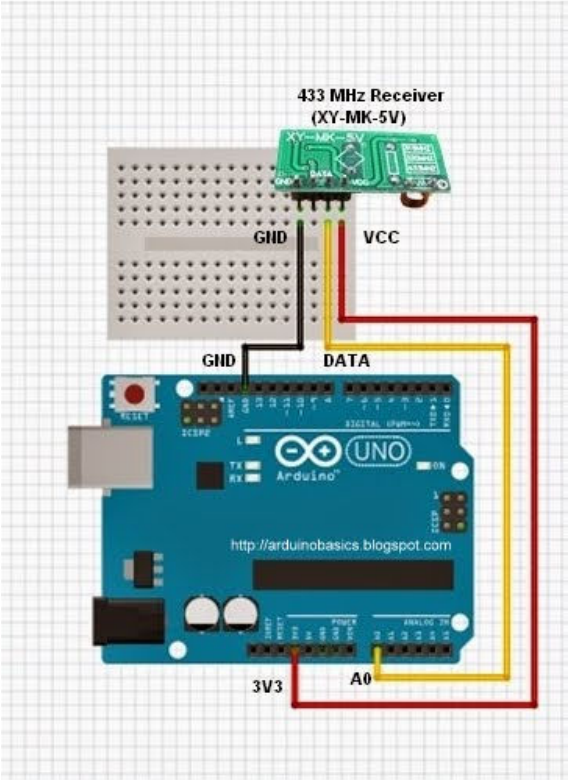
This sketch aims to identify how long each LOW and HIGH signal is (to make up the complete RF remote signal). I have chosen to capture 500 data points(or 250 LOW/HIGH combinations).You may wish to increase or decrease the dataSize variable to accomodate your specific RF signal. In my case, I only really needed 300 data points, because there was a "flat" signal for the last 200 data points (characterised by 200 repetitions of a LOW signal length of 0 and HIGH signal length of 255)

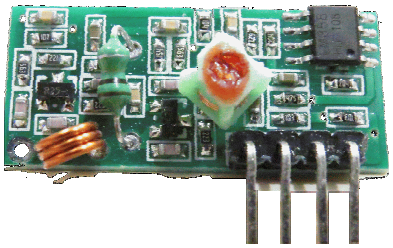
---

```
 2
 3 /*
 4   RF Remote Capture sketch
 5      Written by ScottC 24 Jun 2014
 6      Arduino IDE version 1.0.5
 7      Website: http://arduinobasics.blogspot.com
 8      Receiver: XY-MK-5V
 9      Description: Use Arduino to Receive RF Remote signal
10 ------------------------------------------------------------ */
11
12  const int                  //Arduino memory is limited (max=1700)
13  byte                        //Create an array to store the data
14                              //Onboard LED = digital pin 13
15                              //RF Receiver data pin = Analog pin 0
16  const unsigned int                         //upper threshold value
17  const unsigned int                        //lower threshold value
18  int                          //Set the maximum length of the signal
19  int                     //Variable to measure the length of the signal
20  unsigned long               //Variable to record the start time
21  unsigned long               //Variable to record the end time
22  unsigned long                   //Variable to record signal reading time
23
24
25  void setup
26   Serial begin
27   pinMode         OUTPUT
28
29   /* The following code will only run ONCE --------------
30   ---Press the reset button on the Arduino to run again-- */
31
32   while analogRead
33       //Wait here until a LOW signal is received
34                 micros     //Update start time with every cycle.
35
36   digitalWrite         HIGH    //Turn LED ON
37
38
39   //Read and store the rest of the signal into the storedData array
40   for int
41
42     //Identify the length of the LOW signal---------------LOW
43                     //reset the counter
44     while analogRead
45
46
47
48
49     //Identify the length of the HIGH signal---------------HIGH
50                   //reset the counter
51     while analogRead
```

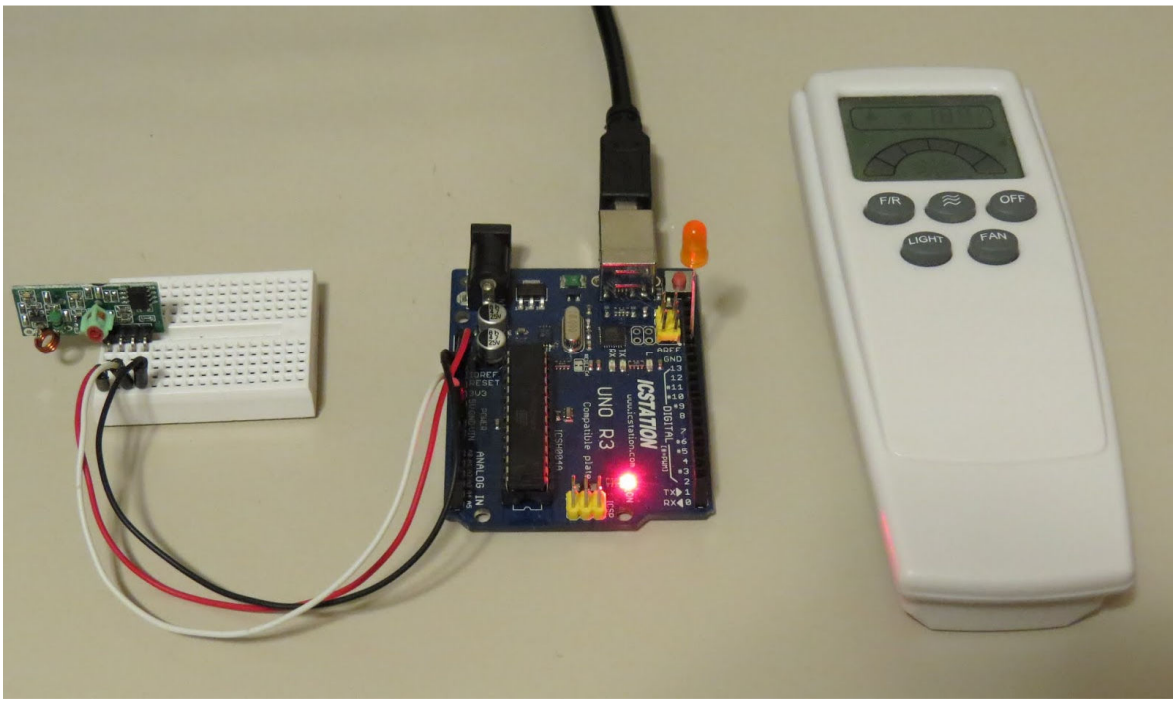
```
52
53
54
55
56      //Any readings between the two threshold values will be ignored.
57      //The LOW or HIGH signal length must be less than the variable "maxSignalLength"
58      //otherwise it will be truncated. All of the HIGH signals and LOW signals combined
59      //must not exceed the variable "dataSize", otherwise it will be truncated.
60      //The maximum number of signals is 1700 - if you try to extend this variable to a
61 higher
62      //number than 1700 - then the Arduino will freeze up and sketch will not work.
63      //-------------------------------------------------------------
64
65
66            micros      //Record the end time of the read period.
67
68
69   digitalWrite          LOW  //Turn LED OFF
70
71   //Send report to the Serial Monitor
72   Serial println "====================="
73   Serial print "Read duration: "
74   Serial print
75   Serial println " microseconds"
76   Serial println "====================="
77   Serial println "LOW,HIGH"
78   delay
79   for int
80     Serial print
81     Serial print ","
82     Serial println
83     delay
84
85
86
87  void loop
      //Do nothing here
```

## Receiver Fritzing Sketch



| RF Receiver XY-MK-5V | Arduino UNO (Or compatible) |
|---|---|
| GND | GND |
| DATA | |
| DATA | Analog 0 |
| VCC | 3V |

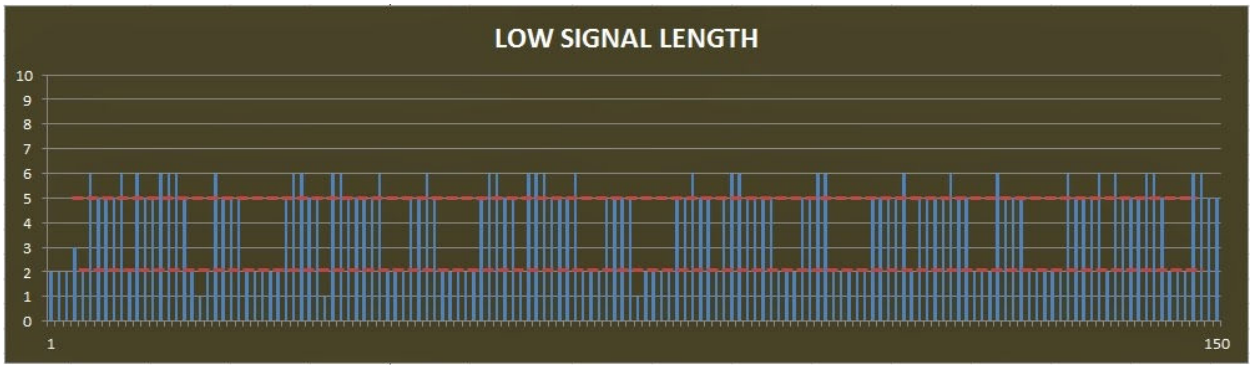



## Results

After pressing the button on the RF remote, the data signal is printed to the Serial Monitor. You can copy the data to a spreadsheet program for review. This is an example of the signal produced after pushing the button on the remote for turning the fan/light on.

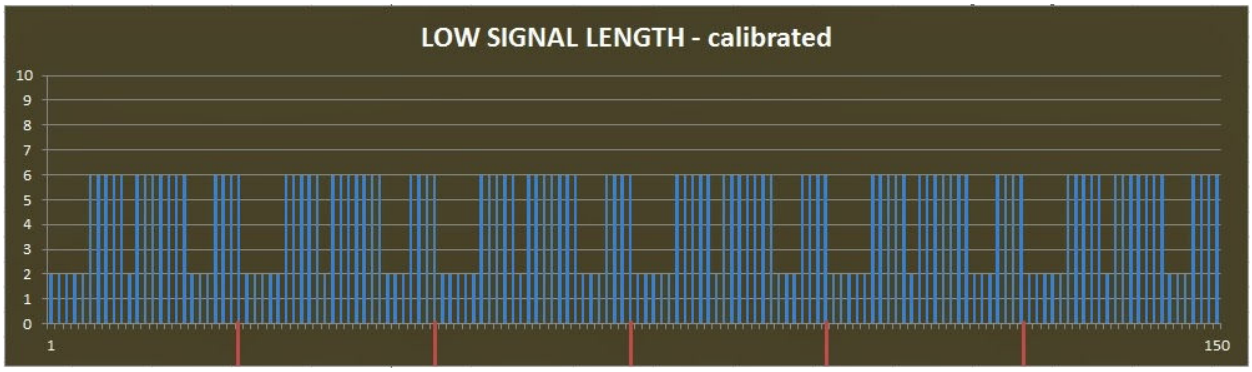
| ========== | |
|---|---|
| Read duration: 3126148 microseconds | |
| ========== | |
| LOW | HIGH |
| 2 | 6 |
| 2 | 6 |
| 2 | 7 |
| 1 | 7 |
| 2 | 3 |
| 5 | 3 |
| 5 | 3 |
| 5 | 3 |
| 5 | 3 |
| 6 | 6 |
| 2 | 3 |
| 5 | 3 |
| 5 | 3 |
| 5 | 3 |
| 5 | 3 |
| 6 | 3 |
| 5 | 3 |
| 5 | 6 |
| 2 | 7 |
| 1 | 7 |
| 1 | 4 |
| 5 | 3 |
| 5 | 3 |
| 5 | 3 |
| 5 | 92 |

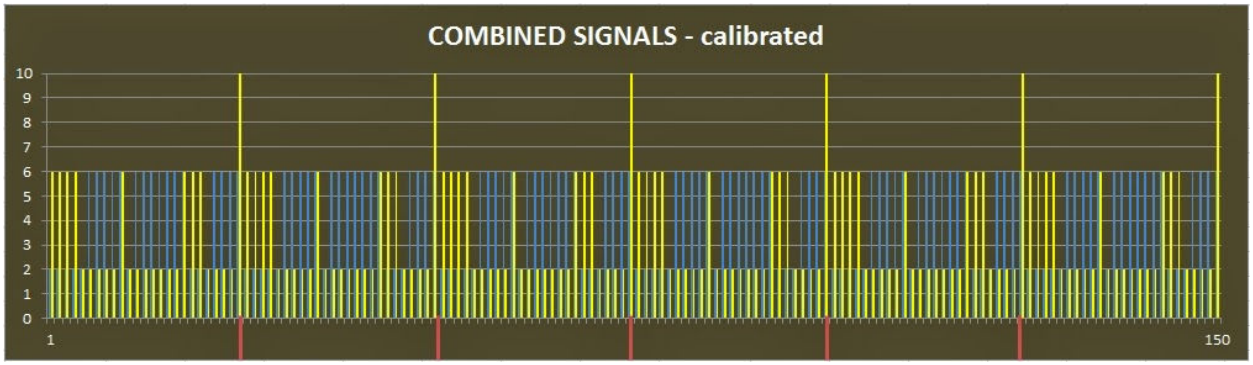
| ========== | |
|---|---|
| Read duration: 3126148 microseconds | |
| ========== | |
| LOW | HIGH |
| 3 | 5 |
| 3 | 6 |
| 2 | 6 |
| 2 | 6 |
| 2 | 3 |
| 5 | 3 |
| 6 | 2 |
| 6 | 2 |
| 6 | 2 |
| 6 | 6 |
| 2 | 3 |
| 5 | 3 |
| 6 | 2 |
| 6 | 2 |
| 6 | 3 |
| 5 | 3 |
| 5 | 3 |
| 6 | 6 |
| 2 | 6 |
| 2 | 6 |
| 2 | 6 |
| 2 | 6 |
| 2 | 7 |
| 2 | 6 |
| 2 | 91 |

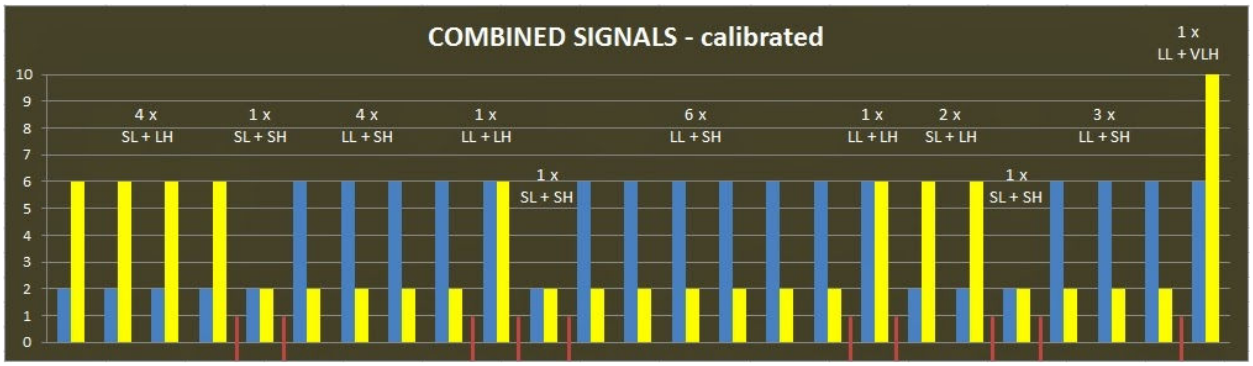



| LOW LENGTH | Frequency |
|---|---|
| 1 | 3 |
| 2 | 50 |
| 3 | 1 |
| 4 | 0 |
| 5 | 62 |
| 6 | 34 |
| 7 | 0 |
| 8 | 0 |
| 9 | 0 |
| 10 | 0 |
| SUM | 150 |

| HIGH LENGTH | Frequency |
|---|---|
| 1 | 0 |
| 2 | 12 |
| 3 | 84 |
| 4 | 0 |
| 5 | 0 |
| 6 | 38 |
| 7 | 10 |
| 91 | 1 |
| 92 | 3 |
| 101 | 1 |
| SUM | 149 |

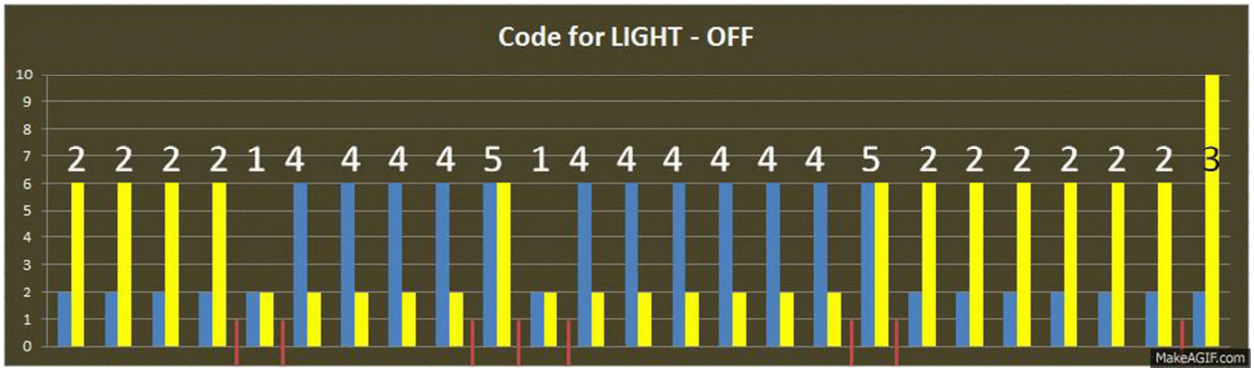


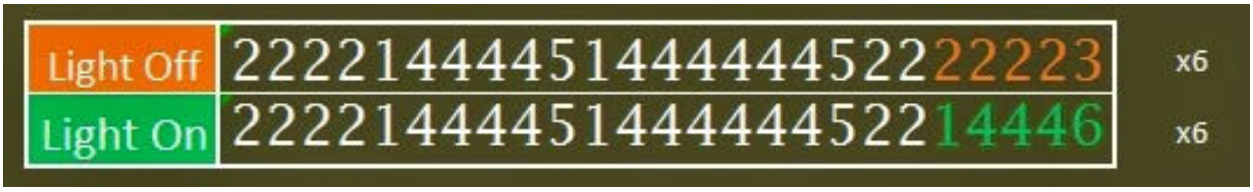



| | SHORT HIGH (SH) | LONG HIGH (LH) | VERY LONG HIGH (VLH) |
|---|---|---|---|
| SHORT LOW (SL) | 1 | 2 | 3 |
| LONG LOW (LL) | 4 | 5 | 6 |

| eg. | | | |
|---|---|---|---|
| | SL + SH = 1 | SL + LH = 2 | SL + VLH = 3 |
| | LL + SH = 4 | LL + LH = 5 | LL + VLH = 6 |




| | |
|---|---|
| 2 | 7 |
| 2 | 6 |
| 2 | 6 |
| 2 | 6 |
| 2 | 6 |
| 2 | 255 |
| 0 | 255 |
| 0 | 255 |
| 0 | 255 |
| 0 | 255 |
| 0 | 255 |
| 0 | 255 |
| 0 | 255 |
| 0 | 255 |
| 0 | 255 |
| 0 | 255 |
| 0 | 255 |
| 0 | 255 |
| 0 | 255 |
| 0 | 255 |
| 0 | 255 |

TimeOut

This is the end of tutorial 2. In the next tutorial, we will use the code acquired from the remote to turn the FAN LIGHT on and off (using the 433 MHz RF transmitter).

**Click here for Tutorial 3**

Posted by Scott C at 01:05

Email ThisBlogThis!Share to TwitterShare to FacebookShare to Pinterest
Labels: 433Mhz, Arduino, ArduinoBasics, best arduino blog, Capture, code, project, Receiver, Remote, RF

# 20 comments:

1.

**caroline & Jeroen**31 July 2014 at 23:12

Thanks for this tutorial.

it's easy to follow, good explanation,

i have managed to retrieve the code from a pir sensor.

please can you give a hint or better write a tutorial how to receive the code found in tutorial 2 to start a sub in the sketch?

thanks for this great work,

Jeroen,

Reply

Replies

1.

**Scott C**1 August 2014 at 00:13

Hi Jeroen,
Can you please explain what you mean by "Start a sub in the sketch" ??
If you want to know how to transmit the code you acquired in tutorial 2, then continue on to tutorial 3.
Please be aware that somewhere in tutorial 1&2 I have managed to get LOWs and HIGHs mixed up. Or maybe I haven't. Anyway - you will find that when you are transmitting the code, that you have to transmit them the other way around (LOWs become HIGHs and vice versa).
See how you go.

2.

**caroline & Jeroen**1 August 2014 at 05:50

Hi Scott, thanks for your reaction,
i have also read tutorial 3 where you send the code with the Arduino.

but that is not my question, i want to receive de code with the arduino!

i have a pir sensor sending 433Mhz codes.

as i say, i have found the code.
Now i want to receive and let the arduino make a sound or something like that..

i have no idea how to start arduino listen only to the found code ?

i hope my question is now better.
(sorry my english is not very good)

3. 

**Scott C** 1 August 2014 at 08:29

Hi Jeroen,
I think I understand your question. You want the Arduino to listen for a specific code, and then get it to do something else ONLY when it receives that code.
This kind of question is always best served within an Arduino forum - where multiple respondents can provide relevant info and code snippets etc, plus the flow of information is specific to your question.
Having said that - your best bet is to have a look at tutorial 4 - which will show you how to get the Arduino to listen for the signal. This signal will be stored within the array called storedData[].
You can then use a function to clean up the signal using "IF" statements. eg.

```
if(storedData[i]<4){
storedData[i]=2;
}
```

You will need to do this also for values above or equal to 4.
Once you have cleaned up the code, you may only be interested in the first 10 or first 1000 values..... And would have to find a way to match these values in the storedData[] array with the pre-defined code you speak of. I can't go through the details here - because it would almost be a whole tutorial in itself - but please post your question on the Arduino Forum - and feel free to come back a leave a link to your forum query in the comments.
Good Luck

4. 

**Miroslav Jernejšek Perec** 19 August 2014 at 19:31

Hi Scott,

This. IS. Amazing.

Thank you for your tutorial a 1000 times...I've been busting my brain with the rcswitch library as I have 2 different remote socket sets and they work with different protocols and I was able to control one, but not the other set.

YOU made it work for me! Thank you again.

So i made it a bit simpler for me, but there is a lot of room for improvement, including the scanner that would display a code to use, perhaps even with calculated timeDelay.

What I did was to use M$ Excel for code calculation, where I used the combination table, upgraded it according to my previously non-working socket (added High medium low) set and used it to calculate combinations.

I must point out, that the pairs (HI,LO) in your sketch are correct, but not also in the table, as you already mentioned. I figured out from received timeouts, that pair 0,255 should be 0 for HI and 255 for LO as it should be "silent" for a while.

So at this point I have 3+4 sockets, I can control with your code fragments, but it also lets me believe we could replicate almost any "as simple as these" control signal...I still have a wireless doorbell to try with.

As you already assumed, there should be different timeDelay lengths as in my case one is 115 and the other is 150 (middle values).

Keep up the good work and keep in touch.

Best regards,
Miroslav J. Perec (mirojp)

5. 

**Scott C**23 August 2014 at 12:46

Thanks for the feedback Miroslav !!
I know I have somehow managed to reverse HIGHs and LOWs somewhere in my investigation process, and may need to go back and revisit where I have gone wrong... Tutorial 4 is where I wanted to end up, whereas Tutorial 1-3 just show how I got there. I wanted a quick way to record the signals and play them back without the need of Excel or a computer... and I managed to do this is in tutorial 4.
Glad to hear that it worked for you !!

6. 

**Scott C**23 August 2014 at 12:48

Here is the link for tutorial 4:
http://arduinobasics.blogspot.com/2014/07/433-mhz-rf-module-with-arduino-tutorial_30.html

**Reply**

2. 

**tj**9 October 2014 at 20:09

Hi Scott,
great tutorial — thank you for the work. I'm trying to read a weather station's signal (and ultimately decode it), and I'll give this method a go.

So far I've often seen the suggestion to hook up the receiver to your computer's Mic In and record the signal with Audacity, and then simply analyze the waveform visually. The audio ADC's sample rate is high enough to produce a very detailed graph.

Anyway, I'm thinking of increasing the Arduino's analogread speed to improve the resolution. (In the hope of getting the best of both worlds.) Is your code available under an open license? I'd throw the result on Github (assuming I produce something sensible).

Tom

Reply

Replies

1. 

**Scott C** 9 October 2014 at 23:48

Hi Tom,
When I started this project, I did not know if it was going to work or not. When it did work, I did a little happy dance. Not sure if it works for anyone else.

Providing that your project is legal, I don't mind how you use my code. I would be interested to hear how your project turns out, and would be happy with just a reference link back to this blog somewhere in your code or Github.

**Reply**

3. 

**Jakub Krieger** 11 January 2015 at 20:33

Hi, I found this tutorial and it's great job!
I've got a question:
how set the receiver to work on 433mhz, because now I receive a lot of noise, and don't know how can I receive clear signal?

Jakub

Reply

Replies

1. 

**Scott C** 12 January 2015 at 14:51

The receiver is already set to work at 433mhz. I found that using 3V instead of 5V worked better for me as it reduced the range of the receiver. The greater the range, then the more likely you are to receive stray/interfering signals from other sources. You could use something to boost the signal of the transmitter to penetrate through the noise. However,

you would need to seek advice to make sure that this is lawful in your country. I used trial an error.
The other alternative is use a different frequency (e.g.315mhz) - once again you will need to check that this (or any) frequency is acceptable in your country.
I was able to work without an antenna without any real issues.

2. 

**Jakub Krieger**13 January 2015 at 04:31

Thank You, yesterday I finally read the code using Your program above. I've set the receiver to 433mHz using a screw on the receiver =D

**Reply**

4. 

**Paul Alcock**18 January 2015 at 17:03

Absolutely awesome set of tutorials Scott.
I now have a beautiful Excel graph for one of my remote control signals. I have tweaked the settings in your sketch that transmits the signal but no joy. With my DVB-T system I now see virtually identical data from the signal from the remote control as I do from the Arduino but no response from the RF device. It is a fairly basic RF socket and I have all the code details but I wanted to test your system before I went on to a more complex device. I have reduced the 10 microseconds down to 3 so there are several attempts within the zone I know I need to work at.

The graph (for off signal) and the comparison between remote signal and arduino signal (for on signal) can be seen at http://www.raspberrypi.org/forums/viewtopic.php?f=37&t=96807

Any ideas why the RF socket is not activating with your excellent code?

Reply

Replies

1. 

**Scott C**19 January 2015 at 21:41

See tutorial 4 for my response to your question there

**Reply**

5. 

**Paul Alcock**18 January 2015 at 21:30

I think you are based in Australia Scott so I thought I would do further tests whilst you sleep with the more 'difficult' RF device I have. Followed your procedure and retained my tweaks in the sketch and it worked FIRST time.
Strange thing is with yesterdays tweaks for the 'easy' RF device the signals reported from DVB-T system for todays 'difficult' device show quite different readings (between actual remote and Arduino signal). Yet the Arduino controls the device just fine.

So I changed most of the sketch back to your original settings and the Arduino continues to control the device. As you have said in your tutorials there is quite a bit of flexibility in what is an acceptable signal for the RF devices.

I have 2 of the 'difficult' RF devices, which I will now rename as type 2 and they 'work' as opposed to the 'easy' RF devices which I would have expected to work as type 1, but don't work.

Type 2 devices have DIP switches and the 2 units I have are set to different codes. The working Arduino sketch will only activate 1 of the devices even when I send the revised code to the unit. The revised code being worked out as I know the binary sequence to send and DVB-T confirms it is sent correctly.

I have just changed the DIP switches on the none working type 2 device to match those of the working type 2 device. Arduino now controls both units. I did this just to check that one unit didn't have a different tolerance to the other.

So there is something I (or the Arduino) doesn't quite understand. Why will type 1 units not work even though the signals seem to match very well and why will type 2 units not work when set to a different code?

Reply

Replies

1. 

**Scott C**19 January 2015 at 21:41

See tutorial 4 for my response to your question there

**Reply**

6. 

**Filippo**21 January 2015 at 06:51

What about using interrupt instead of polling for signal change? I think this post (http://forum.arduino.cc/index.php?topic=129201.0) does exactly this. Would you get a better measurement?

Reply

7. 

**Scott C**21 January 2015 at 08:21

Hi Filippo,
Better measurement ? Not sure.
Feel free to give it a go and let me know.
The method used above worked fine for me, so I had no reason to look any further.

Reply

8. 

**emre ılıksu**26 May 2015 at 04:44

Hi Scott

Am i still able to use the system if i have 5 meters to communicate?

Thx.

Reply

Replies

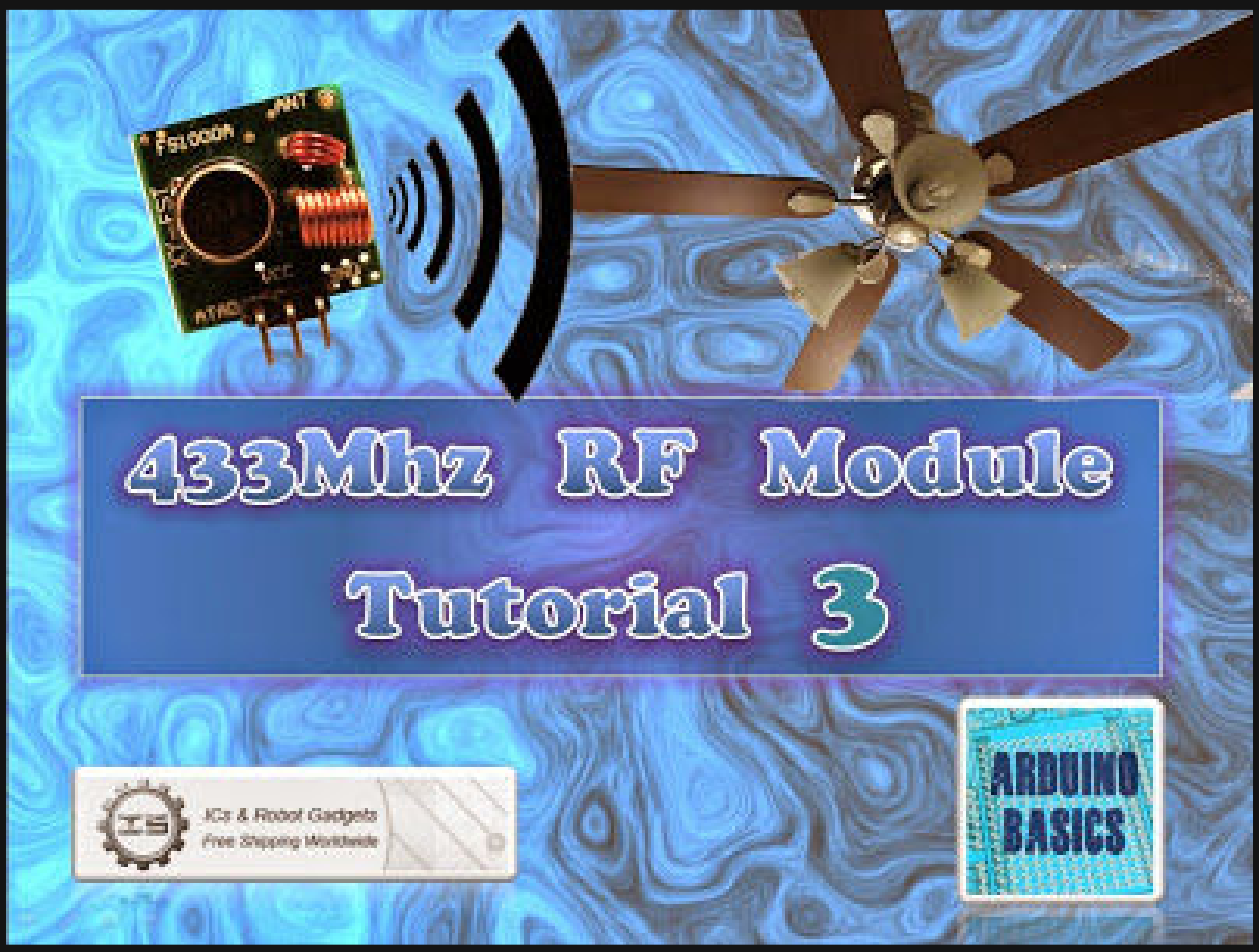
1. 

**Scott C**26 May 2015 at 17:41

Hi emre - if you are talking about communicating between Arduinos, then this is not the right tutorial. If you are talking about using this to communicate with a fan/light receiver over a distance of 5m, then yes - I was able to send signals to the light/fan from a distance of 5m. Performance may vary depending on your module, the power being supplied, other interfering devices etc etc

20 July 2014

# 433 MHz RF module with Arduino Tutorial 3




## Project 3: RF Remote Control Emulation

In the first tutorial, I introduced the 433 MHz Transmitter and Receiver with a simple sketch to test their functionality. In the second tutorial, the 433MHz receiver was used to receive a signal from an RF remote. The RF remote signal was coded based on the pattern and length of its HIGH and LOW signals. The signals received by the remote can be described by the code below:

Code comparison table

| | | |
|---|---|---|
| Light Off | 222144445144444522222223 | x6 |
| Light On | 222144445144444522214446 | x6 |

The RF remote that I am using transmits the same signal 6 times in a row. The signal to turn the light on is different from that used to turn the light off. In tutorial 2, we were able to "listen to" or receive the signal from the RF remote using the RF receiver. I thought it would be possible to just play back the signal received on the Arduino's analogPin, but the time it takes to perform a digital write is different to the time it takes to do an AnalogRead. Therefore it won't work. You need to slow down the digitalWrite speed.
I would like to find out if it is possible to apply this delay to all 433 MHz signal projects, however, I only have one 433 MHz remote.

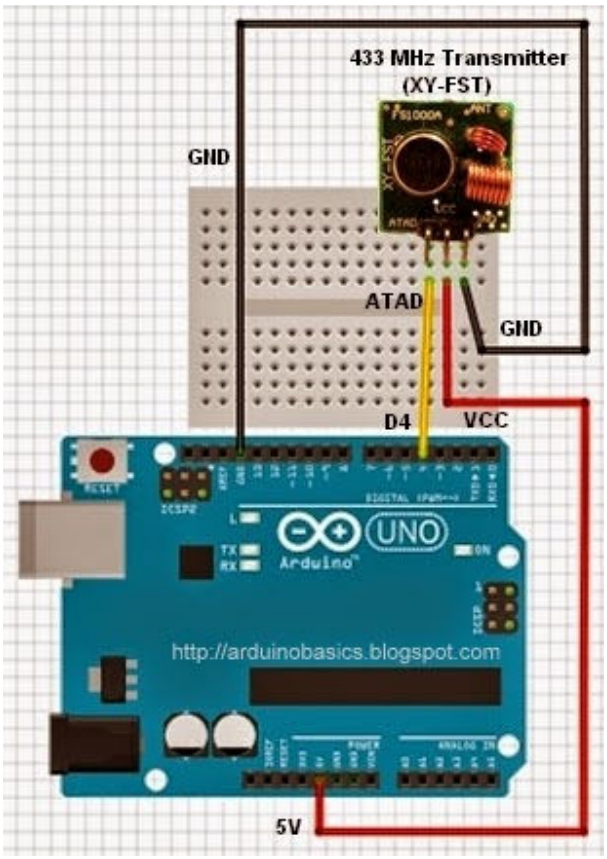
If the delay in your project is the same as mine (or different) I would be keen to know - please leave a comment at the end of the tutorial.

We are going to use trial and error to find the optimal digitalWrite delay time. We will do this by slowly incrementing the delay until the transmission is successful. The transmission is considered successful if the fan-light turns on/off. All we have to do is count the number of transmissions until it is successful, then we should be able to calculate the delay.

## Parts Required

- Arduino UNO or compatible board
- Breadboard
- Wires
- RF Module (433 Mhz) - Transmitter and Receiver pair or the 315 Mhz version
- Mercator Ceiling Fan/Light with Remote

## The Transmitter Fritzing Sketch



# RF Calibration - Arduino Sketch

```
/*
  Transmit sketch - RF Calibration
     Written by ScottC 17 July 2014
     Arduino IDE version 1.0.5
     Website: http://arduinobasics.blogspot.com
     Transmitter: FS1000A/XY-FST
     Description: A simple sketch used to calibrate RF transmission.
 ------------------------------------------------------------ */

                          //RF Transmitter pin = digital pin 4
                          //Onboard LED = digital pin 13

 const int                     //The size of the code to transmit
 int                           //The array used to hold the RF code
 int                                                          //The RF code that
will turn the light ON
 int                                                           //The RF code
that will turn the light OFF
 int                   //Used to switch between turning the light ON and OFF
 int                   // The variable used to calibrate the RF signal lengths.



 void setup
   Serial begin               // Turn the Serial Protocol ON
   pinMode                OUTPUT     //Transmit pin is an output
   pinMode          OUTPUT

 //LED initialisation sequence - gives us some time to get ready
  digitalWrite          HIGH
  delay
  digitalWrite          LOW
  delay



  void loop
                      // switch between light ON and light OFF
                      // transmit the code to RF receiver on the Fan/Light

                       //Increment the timeDelay by 10 microseconds with every
transmission
    delay              //Each transmission will be about 2 seconds apart.
```

```
49
50   /*-----------------------------------------------------------------
51      toggleCode(): This is used to toggle the code for turning
52                    the light ON and OFF
53   -----------------------------------------------------------------*/
54   void
55     if
56        for int
57
58
59
60       else
61       for int
62
63
64
65
66
67
68
69
70
71   /*-----------------------------------------------------------------
72     transmitCode(): Used to transmit the signal to the RF receiver on
73                     the fan/light. There are 6 different HIGH-LOW signal combinations.
74
75                     SH = short high   or  LH = long high
76                                      PLUS
77           SL = short low    or    LL = long low    or    VLL = very long low
78
79   -----------------------------------------------------------------*/
80    void
81     // The LED will be turned on to create a visual signal transmission indicator.
82     digitalWrite         HIGH
83
84    //initialise the variables
85     int
86     int
87
88     //The signal is transmitted 6 times in succession - this may vary with your
89 remote.
90     for int
91       for int
92         switch
93           case    // SH + SL
94
95
96           break
97           case    // SH + LL
98
```

```
99
100             break
101             case    // SH + VLL
102
103
104             break
105             case    // LH + SL
106
107
108             break
109             case    // LH + LL
110
111
112             break
113             case    // LH + VLL
114
115
116             break
117
118
119           /* Transmit a HIGH signal - the duration of transmission will be determined
120              by the highLength and timeDelay variables */
121           digitalWrite                HIGH
122           delayMicroseconds
123
124           /* Transmit a LOW signal - the duration of transmission will be determined
125              by the lowLength and timeDelay variables */
126           digitalWrite                LOW
127           delayMicroseconds
128
129
130       //Turn the LED off after the code has been transmitted.
131       digitalWrite          LOW
```

I used an array to hold the RF code for light ON and light OFF. Each number within the code represents a specific sequence of HIGH and LOW lengths. For example, 2 represents a SHORT HIGH and a LONG LOW combination. A short length = 3, a long length = 7, and a very long length = 92. You need to multiply this by the timeDelay variable to identify how much time to transmit the HIGH and LOW signals for.
The short and long lengths were identified from the experiments performed in tutorial 2 (using the RF receiver).
Each code is transmitted 6 times. The LED is turned on at the beginning of each transmission, and then turned off at the end of the transmission. The timeDelay variable starts at 5 microseconds, and is incremented by 10 microseconds with every transmission.
In the video, you will notice that there is some flexibility in the timeDelay value. The Mercator Fan/Light will turn on and off when the timeDelay variable is anywhere between 75 and 135 microseconds in length. It also seems to transmit successfully when the timeDelay variable is 175 microseconds.
So in theory, if we want to transmit a signal to the fan/light, we should be able to use any value between 75 and 135, however in future projects, I think I will use a value of **105**, which is right about the middle of the range.

## Video

Now that I have the timeDelay variable, I should be able to simplify the steps required to replicate a remote control RF signal. Maybe there is room for one more tutorial on this topic :)

Update: Here it is - tutorial 4
Where you can record and playback an RF signal (without using your computer).

Posted by Scott C at 02:30
Email ThisBlogThis!Share to TwitterShare to FacebookShare to Pinterest
Labels: 433Mhz, Arduino, ArduinoBasics, best arduino blog, Fan, Light, Mercator, project, Remote, RF, Transmitter, tutorial

## 22 comments:

1.

**Rafael**21 July 2014 at 19:02

How far range did you get? May I suggest a tutorial 4: antennas (home made). Thanks for the tutorials!

Reply

Replies

1.

**Scott C**21 July 2014 at 20:10

Hi Rafael,
Without an antenna, I was able to turn the light on and off from the next room. But this distance is much further than I need. I am not interested in transmitting beyond the boundaries of my house (or that room for that matter). Thank you for the suggestion, I would have to seek advice as to whether the use of an antenna would break any laws in Australia... Perhaps somebody can help me out here and provide some advice. I am not interested in breaking laws. But if I am legally allowed to, I might do a tutorial on antennas. What about antennas do you want to know?

**Reply**

2.

**Anonymous**28 July 2014 at 00:26

I must congratulate you on such a fine blog. I discovered it today and i can't stop reading it. I wish you all the luck in the future works

Reply

3. 

**Miroslav Jernejšek Perec** 19 August 2014 at 19:32

Hi Scott,

This. IS. Amazing.

Thank you for your tutorial a 1000 times...I've been busting my brain with the rcswitch library as I have 2 different remote socket sets and they work with different protocols and I was able to control one, but not the other set.

YOU made it work for me! Thank you again.

So i made it a bit simpler for me, but there is a lot of room for improvement, including the scanner that would display a code to use, perhaps even with calculated timeDelay.

What I did was to use M$ Excel for code calculation, where I used the combination table, upgraded it according to my previously non-working socket (added High medium low) set and used it to calculate combinations.

I must point out, that the pairs (HI,LO) in your sketch are correct, but not also in the table, as you already mentioned. I figured out from received timeouts, that pair 0,255 should be 0 for HI and 255 for LO as it should be "silent" for a while.

So at this point I have 3+4 sockets, I can control with your code fragments, but it also lets me believe we could replicate almost any "as simple as these" control signal...I still have a wireless doorbell to try with.

As you already assumed, there should be different timeDelay lengths as in my case one is 115 and the other is 150 (middle values).

Keep up the good work and keep in touch.

Best regards,
Miroslav J. Perec (mirojp)

Reply

4. 

**Anonymous** 2 December 2014 at 09:22

Hello Scott,

thank you very much for this tutorial and code. I am able to control my RC sockets which I got from Maplin with the Arduino. Well written tutorial and great effort on your part. Thank you very much, I can't wait to read trough your blog. Keep up the great work,

best regards,
Vlada

Reply

Replies

1. 

**Scott C**2 December 2014 at 10:16

Thanks for your feedback Vlada. I am glad it helped.

**Reply**

5. 

**Anonymous**29 December 2014 at 21:22

Thank you so much for this! I was sooo close about giving up my project, spent night over night reading tutorial after tutorial but yours really saved my day =D I used it to program a receiver for a specific remote control. So simple and yet so brilliant. Works perfectly and accurate! Adding code while developing shifted the timings slightly (of course) but splitting the signal into high and low peaks (1 and 0) made it very tolerant. As my remote has so many buttons I separated the big remote code into a codeA (common code) and a button specific code B part which keeps the button code small (Attiny..) and makes it easy to blink out the button-codes with poor mans display in debugging mode. Absolutely cool *thumbs up* definitely my best Christmas gift! Thx again! BR Martin

Reply

Replies

1. 

**Scott C**31 December 2014 at 09:56

Merry Christmas BR Martin, I am glad it worked for you.

**Reply**

6. 

**I Fonteyn**6 January 2015 at 17:36

Hello,

what a well written and easy tot understand tutorial.
I stumbled on this, looking for a soultion to make my remote controlled sockets automated. The idea is to make my sockets go ON at 4 pm (lets say) and OFF at 11 pm, without me pushings buttons every day. Can your scheme work with some sort of clock? I am kind of a noob when it comes to arduino, but have some knowledge of programming.
thanks in advance and again congrats with the nice blog.

I. Fonteyn

Reply

Replies

1. 

**Scott C**7 January 2015 at 10:27

Hi I. Fonteyn,
Thank you for your feedback.
I recently got myself a Real Time Clock (RTC). There are many different types out there, however, I got mine from EpicTinker
You should be able to use something like this in conjunction with the RF transmitter to do what you want to do. I might try this myself sometime. However don't wait for me, as I usually take a while to get these tutorials out. But I like the idea.

2. 

**edwin**18 February 2015 at 14:05

I am working on an 'at home' simulator to be published on instructables and what you say is in fact quite easy.
In a loop you check for the time and if it is your desired time you send the command to switch your light on or off.
I use standard libraries for that for the RTC as well asfor the 433 transmitter.

3. 

**Edwin**18 February 2015 at 14:13

hee u find a section of code to switch a device on at 6:10 a.m.:
if (now.hour()==6 && now.minute()==(10) ){
ab440Switch.sendSignal(29,'A',true);
}
that is a bit simplified as i do also check if the code already has been send and i store that in NVRAM but you do not really need that

**Reply**

7. 

**carlo carli**25 January 2015 at 00:07

*This comment has been removed by a blog administrator.*

Reply

Replies

1. 

**Scott C**2 February 2015 at 15:04

Hi Carlo - have a look at tutorial 4. This may help you.
I have deleted your post - due to the mere size of the data you posted. Best to post in a forum, and then link to that post - otherwise the comments section on this blog post will be ridiculously long. Also, others may be able to help.

**Reply**

8. 

**Edwin**18 February 2015 at 13:59

very interesting. with regard to sending the data u take a different approach than what i usually see: highs and lows being defined by a long and short combination, rather than having in yr case 6 patterns defined. But as long as it works thatis great

Reply

9.

**gopikrishna.chowdhury@gmail.com**10 May 2015 at 17:17

Hi Scott,

This is the only blog all over the internet which explains on how to use RF sensors. I appreciate your efforts.

I tried to use your code and my RF receiver has generated the following output without any transmitters around.

0 1 0 0 774 0 773 774 0 0 0 0 0 0 1 773 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 773 0 0 774 773 0 0 0 773 0 0 0 584
0 0 0 0 0 717 0 0 0 0 0 774 773 0 774 0 771 774 0 0 0 0

Now when I press my Car lock/unlock Remote control it has the below pattern only.

0 0 0 0 0 0 0 0 702 0 0 0 0 0 553 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 771 0 0 775 11 0 0 474 0 0 0 0 773 1 773
0

But I am not able to analyze how to decode the above code which is generated when a particular button is pressed on my RF remote .

Are there any tools or are there any better ways to analyze what it is trying to send ?

My main aim is to replicate the codes sent by my TV RF remote.

Reply

Replies

1. 

**Scott C** 11 May 2015 at 00:28

Have you tried tutorial #4?
This is probably the one that is most likely to work.
Also please make sure that when you press a button on the RF remote, that you have the remote as close to to the RF receiver as possible.

Also , this is not the only tutorial around. There are plenty other examples out there. You just have to find them...

2. 

**Anonymous** 5 June 2015 at 23:53

Hi, I had a similar problem when I tested the concept and had lots of queer results without any transmitter. The problem was my wireless router, when i turned it off everything worked fine.
Hope this helps.

**Reply**

10. 

**Marc P.** 8 August 2015 at 10:14

Hey man. Thank you so much i can clone the keyremote for my mom's car, and garage's... :))

You are sooo amazing..

thanks again!
Marc.

Reply

Replies

1. 

**Scott C** 8 August 2015 at 18:17

I appreciate the feedback Marc, and I am glad that it worked for you.
I would be interested in seeing a YouTube video of your project in action :)
Feel free to leave a link below if you want to.

Reply

11. 

**Marc P.**9 August 2015 at 00:45

Hi Scott C. My garage remote is at 15 MHz, unfortunally doesnt work :(

i dont know the frequency of the remote of the car of my mom.. i will try it..

thank you!
Marc.

Reply

Replies

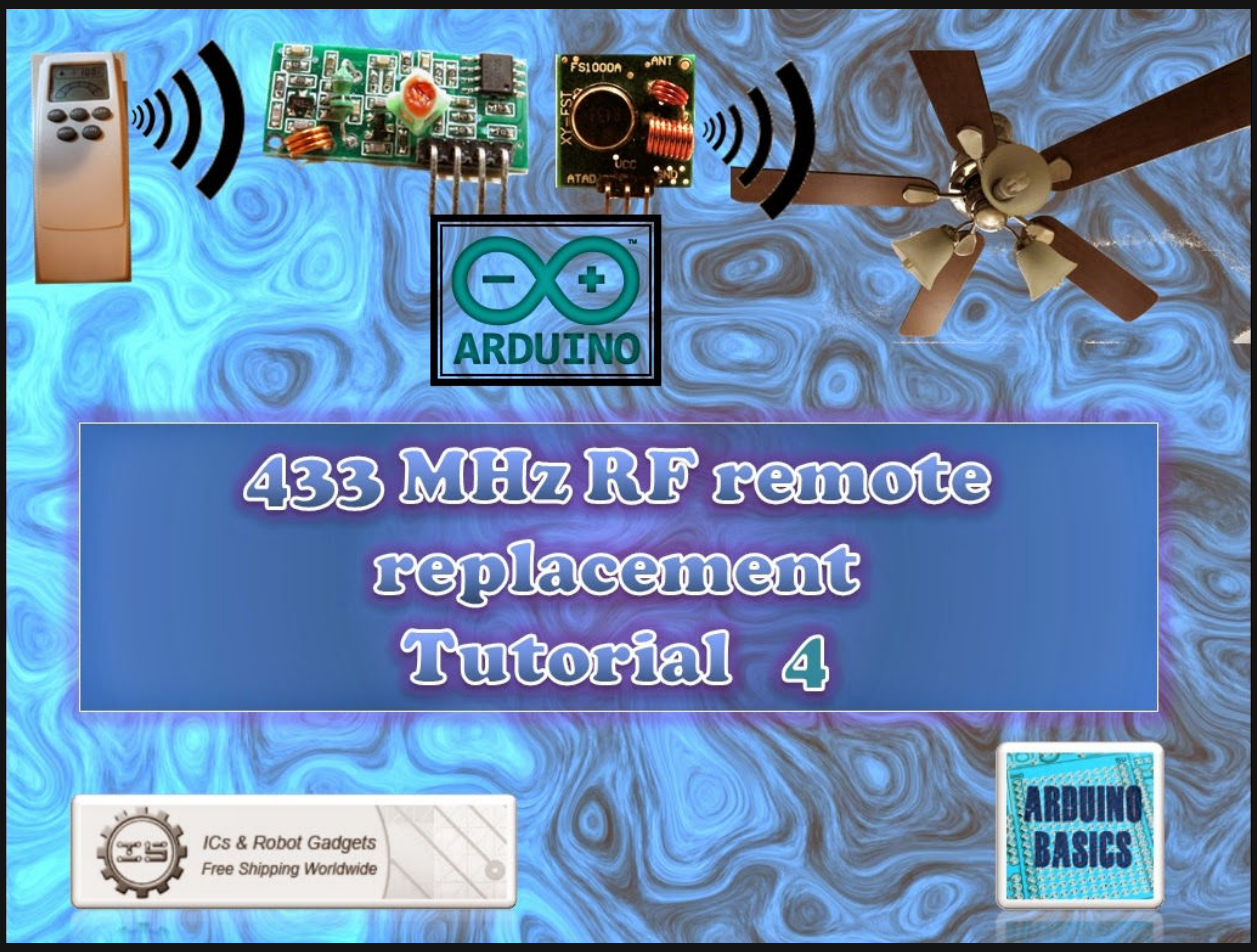
1. 

**Scott C**9 August 2015 at 08:30

You have to have a matching transmitter/receiver. My remote had a frequency of 433MHz, so was able to copy the signals without issue. For garage remotes and car remotes, you may come across rolling codes, so even if you have the correct signal frequency, this method may still not be successful. You can tell by the receiving signal. If you consistently receive the same code over and over and over, every time you transmit from the remote, then this method is much more likely to work for you. Good luck.

30 July 2014

# 433 MHz RF module with Arduino Tutorial 4:




WARNING: Please check whether you can legally use RF transmitters and receivers at your location before attempting this project (or buying the components). This project is aimed at those who are looking to automate their home.

Carrying on from my previous "433MHz transmitter and receiver" tutorials (1,2 & 3): I have thrown away the need to process the signal with a computer. This means that we can now get the Arduino to record the signal from an RF remote (in close proximity), and play it back in no time at all.

The Arduino will forget the signal when powered down or when the board is reset. The Arduino does not have an extensive memory - there is a limit to how many signals can be stored on the board at any one time. Some people have opted to create a "code" in their projects to help maximise the number of signals stored on the board. In the name of simplicity, I will not encode the signal like I did in my previous tutorials.

I will get the Arduino to record the signal and play it back - with the help of a button. The button will help manage the overall process, and control the flow of code.

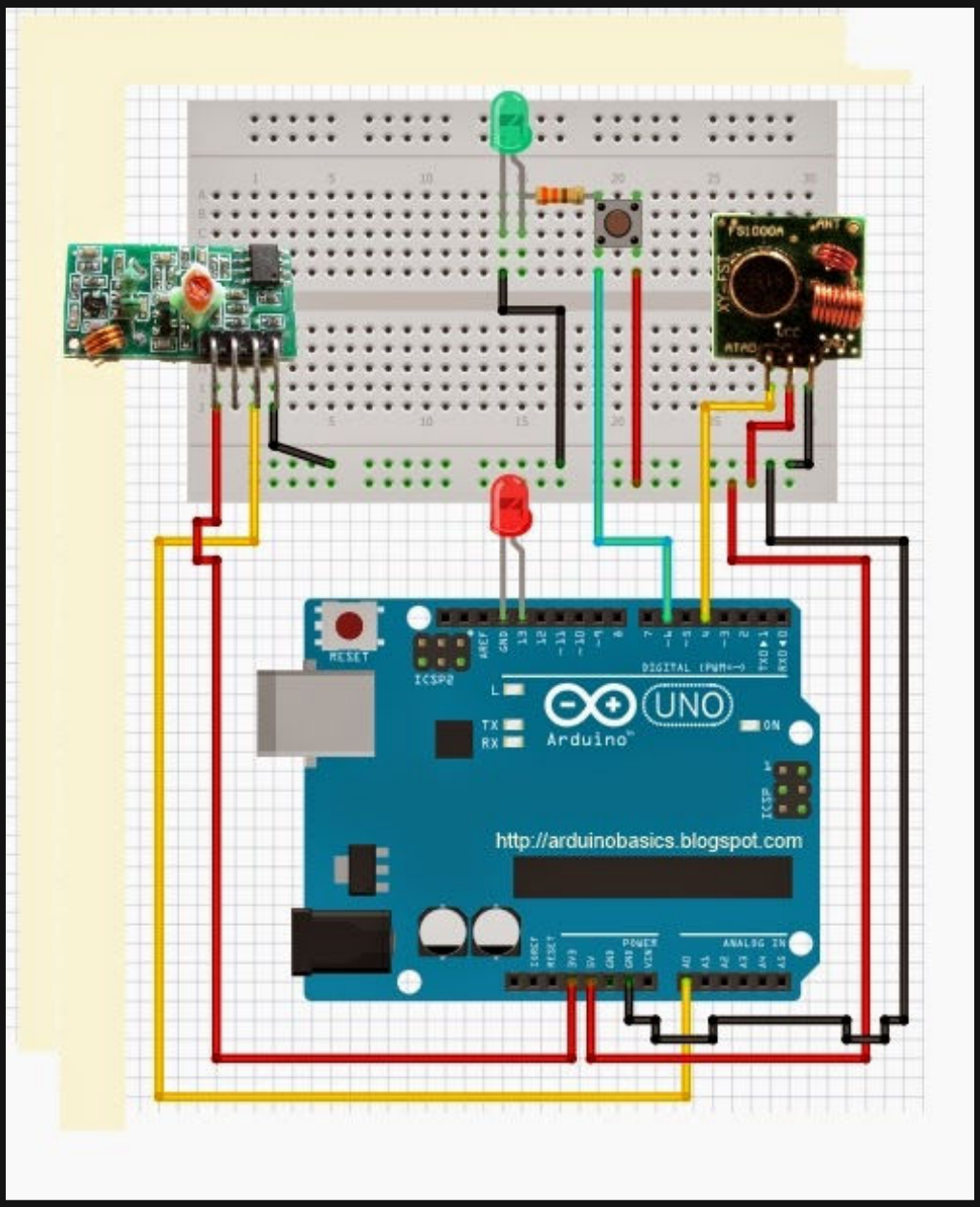
Apart from uploading the sketch to the Arduino, this project will not require the use of a computer. Nor will it need a sound card, or any special libraries. Here are the parts required:

## Parts Required:

- Arduino UNO or compatible board
- Breadboard

- Button
- Red and Green LED
- 330 ohm resistor(s)
- Wires
- RF Module (433 Mhz) - Transmitter and Receiver pair or the 315 Mhz version
- Mercator Ceiling Fan/Light with Remote

## Fritzing Sketch




## Arduino Sketch

```
1
2
3 /*
```

```
 4   433 MHz RF REMOTE REPLAY sketch
 5      Written by ScottC 24 Jul 2014
 6      Arduino IDE version 1.0.5
 7      Website: http://arduinobasics.blogspot.com
 8      Receiver: XY-MK-5V      Transmitter: FS1000A/XY-FST
 9      Description: Use Arduino to receive and transmit RF Remote signal
10  ------------------------------------------------------------- */
11
12                              //RF Receiver data pin = Analog pin 0
13                            //RF Transmitter pin = digital pin 4
14                              //The button attached to digital pin 6
15                            //Onboard LED = digital pin 13
16
17  const int                   //Arduino memory is limited (max=1700)
18  byte                         //Create an array to store the data
19  const unsigned int                    //signal threshold value
20  int                           //Set the maximum length of the signal
21  int                         //Variable to measure the length of the signal
22  int                         //Variable to control the flow of code using button presses
23  int                         //Variable to hold the state of the button
24  int                         //Used to slow down the signal transmission (can be from 75 -
25 135)
26
27  void setup
28    Serial begin            //Initialise Serial communication - only required if you
29 plan to print to the Serial monitor
30    pinMode                 OUTPUT
31    pinMode          OUTPUT
32    pinMode          INPUT
33
34
35  void loop
36                digitalRead
37
38    if                              HIGH
39      //Serial.println("Listening for Signal");
40
41
42
43
44                digitalRead
45
46    if                              HIGH
47      //Serial.println("Send Signal");
48
49
50
51    delay
52
53
```

```
 54
 55 /* ----------------------------------------------------------------------------
 56     Initialise the array used to store the signal
 57    ----------------------------------------------------------------------------*/
 58 void
 59   for int
 60
 61
 62
 63
 64
 65
 66 /* ----------------------------------------------------------------------------
 67     Listen for the signal from the RF remote. Blink the RED LED at the beginning to
 68 help visualise the process
 69     And also turn RED LED on when receiving the RF signal
 70    ---------------------------------------------------------------------------- */
 71 void
 72   digitalWrite         HIGH
 73   delay
 74   digitalWrite        LOW
 75   while analogRead
 76     //Wait here until an RF signal is received
 77
 78   digitalWrite         HIGH
 79
 80   //Read and store the rest of the signal into the storedData array
 81   for int
 82
 83      //Identify the length of the HIGH signal---------------HIGH
 84                     //reset the counter
 85      while analogRead
 86
 87
 88                                    //Store the length of the HIGH signal
 89
 90
 91      //Identify the length of the LOW signal----------------LOW
 92                    //reset the counter
 93      while analogRead
 94
 95
 96                                    //Store the length of the LOW signal
 97
 98
 99                        //Account for the first AnalogRead>threshold = lost while
100 listening for signal
101     digitalWrite         LOW
102
103
```

```
104
105  /*-------------------------------------------------------------------------
106     Send the stored signal to the FAN/LIGHT's RF receiver. A time delay is required to
107 synchronise
108     the digitalWrite timeframe with the 433MHz signal requirements. This has not been
109 tested with different
110     frequencies.
111     ------------------------------------------------------------------------- */
112  void
113    digitalWrite          HIGH
114    for int
115        //Send HIGH signal
116        digitalWrite                HIGH
117        delayMicroseconds
118        //Send LOW signal
119        digitalWrite                LOW
120        delayMicroseconds
121
122    digitalWrite          LOW
123    delay
124
125
126    /*-----View Signal in Serial Monitor
127    for(int i=0; i<dataSize; i=i+2){
        Serial.println("HIGH,LOW");
        Serial.print(storedData[i]);
        Serial.print(",");
        Serial.println(storedData[i+1]);
      }
      -------------------------------- */
```

Now let's see this project in action !

Have a look at the video below to see the Arduino turning a light and fan on/off shortly after receiving the RF signal from the RF remote. The video will also show you how to put this whole project together - step by step.

## The Video

This concludes my 433MHz transmitter and receiver tutorials (for now). I hope you enjoyed them. Please let me know whether this worked for you or not.
I have not tested this project with other remotes or other frequencies - so would be interested to find out whether this technique can be used for ALL RF projects ??

Loading...

If you like this page, please do me a favour and show your appreciation :

Visit my ArduinoBasics Google + page.
Follow me on Twitter by looking for ScottC @ArduinoBasics.
Have a look at my videos on my YouTube channel.


However, if you do not have a google profile...
Feel free to share this page with your friends in any way you see fit.

Posted by Scott C at 02:09
Email ThisBlogThis!Share to TwitterShare to FacebookShare to Pinterest
Labels: 433Mhz, Arduino, ArduinoBasics, best arduino blog, project, Remote, Replay, RF

## 28 comments:

1.

**Boss Boss**27 August 2014 at 07:11

Great

Reply

2.

**Rahul Poddar**9 September 2014 at 16:21

Hello, Nice explanation. This helps to understand the basic principle for receiver and transmitter using arduino. Just another query it is possible to take a note of the sniffed data from existing remote and play the same data via another transmitter using arduino. Basically I want o first read the data from each button of the existing RF remote and use the data in my transmitter to play it.

Reply

Replies

1. 

**Scott C**9 September 2014 at 21:32

At the end of the Arduino code - you will see a section which allows you to transmit the signal to the serial monitor. You could send the signal to an SD card, or just store them on your computer using the Processing IDE . I assume you have already looked at tutorials 1 - 3.
Once you have the stored signal, you can just send it back to another Arduino, and transmit the signal as required.

**Reply**

3. 

**Anonymous**3 October 2014 at 02:09

Hei, thanks for your interesting tutorial. I'm a newbie to Arduino and I wonder how come the LED on pin 13 and ground wouldn't burn without any resistor. Thank you :)

Reply

Replies

1. 

**Scott C**4 October 2014 at 09:15

That is a good question. Some people say that there is an onboard resistor attached to pin13, while others say that newer boards don't have this resistor. While I have used this technique numerous times without any problems, I would recommend that you either stick with the onboard LED, or attach a resistor in series (just in case).
Excellent question !!

2. 

**Scott C**4 October 2014 at 09:27

Here is an article that supports the use of a resistor in series. But in the comments of that article, there are people that say that some Arduino boards have an onboard resistor in series with pin13.

Use this technique at your own risk. But thank you for this question. Because I think I will recommend a resistor in future tutorials.

Reply

4. 

**Anonymous**27 October 2014 at 11:50

Hello,

Thanks for your tutorials, however on tutorial #4 your push button is floating so it will turn on at random, it's much better to make use of the internal pull-up resistor by adding the following to void setup() :-

digitalWrite(button, HIGH);

Then change the coding so buttonVal==HIGH becomes buttonVal==LOW and wire the button so when pressed it goes to GND.

In this state the button is always pulled up HIGH and GND pulls it LOW when the button is pressed, otherwise the button will float and trigger randomly.

Thanks.

Reply

Replies

1. 

**Scott C**7 January 2015 at 10:45

Ha Ha, I always forget about this floating problem... Well picked up. Thank you.

2. 

**Paul Alcock**19 January 2015 at 21:01

'Anonymous' or Scott can please you provide a few more details on the code and circuit changes to remove the floating button issue. Also for the serial print to be effective as it is hit and miss at the moment with the floating button.

As an interim I worked out that the best way to get a reliable recording is to have the button switched on when the Arduino is reset. This is giving excellent results and I'm truly amazed at how effective this is.

It is managing to record and playback all ID's for 'type 2' devices that were failing as highlighted in my post on the tutorial 2 page. It is also working for some RF activated relays that I will call type 3 devices. Strangely it is still not working for type 1 devices. Type 1 are Status remote sockets as shown at http://www.amazon.co.uk/Status-SREMSOC3PK3-Remote-Control-Socket/dp/B003XOXAVG

The protocol for the Status remote sockets is covered in Geoff Johnson's article at http://www.hoagieshouse.com/RaspberryPi/RCSockets/RCPlug.html

Scott is there any reason why your sketch isn't able to record and playback this protocol?

For reference I have included below an on and an off signal for the Status remote sockets as recorder by rtl_433.exe (rtl recording software for windows).

3. 

**Scott C** 19 January 2015 at 21:39

*This comment has been removed by the author.*

4. 

**Paul Alcock** 19 January 2015 at 21:53

Sorry Scott, forget to paste in the on off signals. See below.

```
signal_len = 85221, pulses = 174
Iteration 1. t: 156 min: 79 (76) max: 234 (98) delta 125
Iteration 2. t: 156 min: 79 (76) max: 234 (98) delta 0
Pulse coding: Short pulse length 79 - Long pulse length 234

Short distance: 65, long distance: 221, packet distance: 2378

p_limit: 156

[00] {25} 04 db fb 00 : 00000100 11011011 11111011 00000000
[01] {25} 04 db fb 00 : 00000100 11011011 11111011 00000000
[02] {25} 04 db fb 00 : 00000100 11011011 11111011 00000000
[03] {25} 04 db fb 00 : 00000100 11011011 11111011 00000000
[04] {25} 04 db fb 00 : 00000100 11011011 11111011 00000000
[05] {25} 04 db fb 00 : 00000100 11011011 11111011 00000000
[06] {24} 04 db fb 00 : 00000100 11011011 11111011 00000000

*** signal_start = 312812343, signal_end = 312878215
signal_len = 65872, pulses = 124
Iteration 1. t: 156 min: 79 (59) max: 234 (65) delta 212
Iteration 2. t: 156 min: 79 (59) max: 234 (65) delta 0
Pulse coding: Short pulse length 79 - Long pulse length 234

Short distance: 66, long distance: 222, packet distance: 2379

p_limit: 156

[00] {25} 04 db f3 00 : 00000100 11011011 11110011 00000000
[01] {25} 04 db f3 00 : 00000100 11011011 11110011 00000000
[02] {25} 04 db f3 00 : 00000100 11011011 11110011 00000000
[03] {25} 04 db f3 00 : 00000100 11011011 11110011 00000000
[04] {24} 04 db f3 00 : 00000100 11011011 11110011 00000000
```

5. 

**Scott C** 19 January 2015 at 22:07

Hi Paul,
Wow - you seem to be knee deep in RF signals :)
I was just about to respond to you, to ask if you had a chance of looking at Tutorial 4, when I came across this message... so looks like you have managed to work your way through most of your problems....
I would like to say that I did have to go through a bit of trial and error, and sometimes got a bit confused by the RF results that I was getting. There are few sites that explain that there is a delay when you write to the pin, and then go on explain the frequency calculations.

To solve the Floating push button - read this tutorial.

I am not seeing your on/off signals recorded by rtl_433.exe
You may want to expand the dataSize variable to a higher value, and once you know the length of the signal, bring it down to what-ever number you discover. I think the maximum number you can use for this variable is 1700 (on an Arduino UNO) - due to memory restrictions. But see how you go.
And you may need to play around with the timeDelay variable as I did in Tutorial 3. You might want to incorporate the calibration process into this tutorial??

And finally, I think Geoff Johnson noted that HIGHs and LOWs were reversed in his sketch. I think I had a similar issue, so you may want to try reversing them to see if that helps.

6. 

**Scott C** 19 January 2015 at 22:13

Sorry Paul,

I re-posted my message because I needed to change something and add something. And now I see you have posted your signals... but I don't know if it actually helps me to figure out what is happening on your side...

7. 

**Scott C** 19 January 2015 at 22:23

Hi Paul,

Geoff Johnson uses a timeDelay of around 222 Microseconds ??

8. 

**Paul Alcock**20 January 2015 at 20:55

Hi Scott,

Spent a few hours yesterday trying to 'crack' the Status sockets but no joy. Strange thing is the relays I use have a RF signal 'learn' facility and when I use the Status remote to 'teach' them the RF they are controlled by your sketch. The same applies when I use another well done RF library.

Cloned another RF device earlier today. It is a 'security' device which I will call type 4. So types 2, 3 and 4 working well but not type 1, yet.

9. 

**Scott C**21 January 2015 at 00:43

Paul - Glad to hear that this is working for so many of your devices - but there is obviously something different about your "type 1" device. Check to make sure that it is not using a rolling code (after some time)??
I will also advise you to post your question to the Arduino Forums... tell them what you are trying to do, what you have done, and see if the "experts" can get you on the right track. Show the code you are using - as this may help narrow down the issue.

I would be interested in the outcome - so please post a link to your forum post.

Regards
Scott

10. 

**Paul Alcock**22 January 2015 at 02:46

Hi Scott,
The Status sockets (type 1 device) do not use rolling codes of any type. Looking at the devices that your sketch controls compared with the one type that is doesn't (excluding rolling codes of course) the common factor appears to be signal length. Your sketch works where the long pulse is double the short pulse but Status sockets have long pulses three times the length of the short pulses. This is the only factor I can see different, plus maybe the reversing of the signal (low becomes high and high becomes low).

For reference Pilight has no trouble controlling the Status sockets.

Can you please contact me via Facebook.

**Reply**

5. 

**Anonymous**5 January 2015 at 23:27

Hi there,
best wishes from Germany. I started the other way round. Found these cheap modules on Ebay where they are available for less than 1 Euro, if you take five of them. That's only about half the amount you have to pay for a beer in a pub here. Looking for information after the purchase I found your tutorials. It was a really good idea to keep them simple. I am still a beginner with the Arduino and those explanations on the internet where the authors try to impress the reader with their knowledge are not helpful. So yours on the contrary is very helpful.
Thanks says Georg

Reply

Replies

1. 

**Scott C**7 January 2015 at 10:44

Possibly because I don't have much knowledge :)

Thanks for the feedback Georg

2. 

**Paul Alcock**19 January 2015 at 21:03

You have plenty of knowledge Scott. Thanks for a great set of tutorials on RF.

3. 

**Arjan**24 February 2015 at 05:52

Great stuff. Works like a charm. Since I am a beginner at Arduino stuff too, this post was really helpful to start getting up to speed and have fun at the same time :)

4. 

**Scott C**24 February 2015 at 10:03

+Arjan - am glad it worked for you. Thanks for the feedback.

**Reply**

6. 

**Matt Watson**2 June 2015 at 02:44

Hi Scott,

Thank you for creating this tutorial! I am new to electronic/Arduino and have decided to use your tutorial to build my first real project for the home.

The ceiling fans in my home seem to operate on 303.9MHz according to the FCC docs (https://apps.fcc.gov/oetcf/eas/reports/ViewExhibitReport.cfm?mode=Exhibits&RequestTimeout=500&calledFromFrame=N&application_id=u9yrEznUvQJtU3fTqdh%2BxA%3D%3D&fcc_id=L3HFAN-9T). Do you think the 315MHz receiver/transmitter would work for me?

Reply

Replies

1. 

**Scott C**2 June 2015 at 09:25

Hi Matt,
I don't think the 315MHz transmitter will work with your fan - but I'm not 100% sure. However, the modules are so cheap, that it might be worth trying. If you want to make your own transmitter, then this article may be useful: http://www.talkingelectronics.com/projects/27MHz%20Transmitters/27MHzLinks-3.html

2. 

**Scott C**2 June 2015 at 09:30

This discussion may also enlighten you. https://community.particle.io/t/303-9-mhz-rf-transmitter/9062/9

I did not read all of it - so don't know if they figured it all out in the end. But looks like they were trying to do the same thing as you.

3. 

**Matt Watson**3 June 2015 at 06:11

Thanks for your reply, Scott. I think I will go ahead and buy the transmitters on the off chance that they do work (like you said, incredibly cheap). I'll be sure to report back with my findings just in case anyone else is looking for this information in the future. Thanks again!

Reply

7. 

**Unknown**28 June 2015 at 06:48

Forgive me for being a slow follower and thank you so much for this tutorial. I have been studying your tutorial for a few days now and still confused. I enabled the "view signal serial monitor". and got the log (listed below) when pressed the channel 1 o the ZAP Outet switch transmitter.

http://www.amazon.com/Etekcity-Programmable-Electrical-Converter-Appliances/dp/B00DQELHBS/ref=sr_1_sc_1?ie=UTF8&qid=1435444721&sr=8-1-spell&keywords=ZAP+Outet+switch

@Scott, Can you please guide me what is the next steps to do to get the RF Calibration Table for on/off switch?

Reply

Replies

1. 

**Scott C**28 June 2015 at 12:52

Hi Unknown,

I have moved your query to the ArduinoBasics forum - which is a much better place to discuss this type of thing: http://arduinobasics-forum.1116184.n5.nabble.com/Quick-questions-about-ArduinoBasics-projects-f3.html

Thank you very much Paul Alcock for your response to Unknown's question. Just letting you know that I moved your response to the forum also.

Regards,
Scott