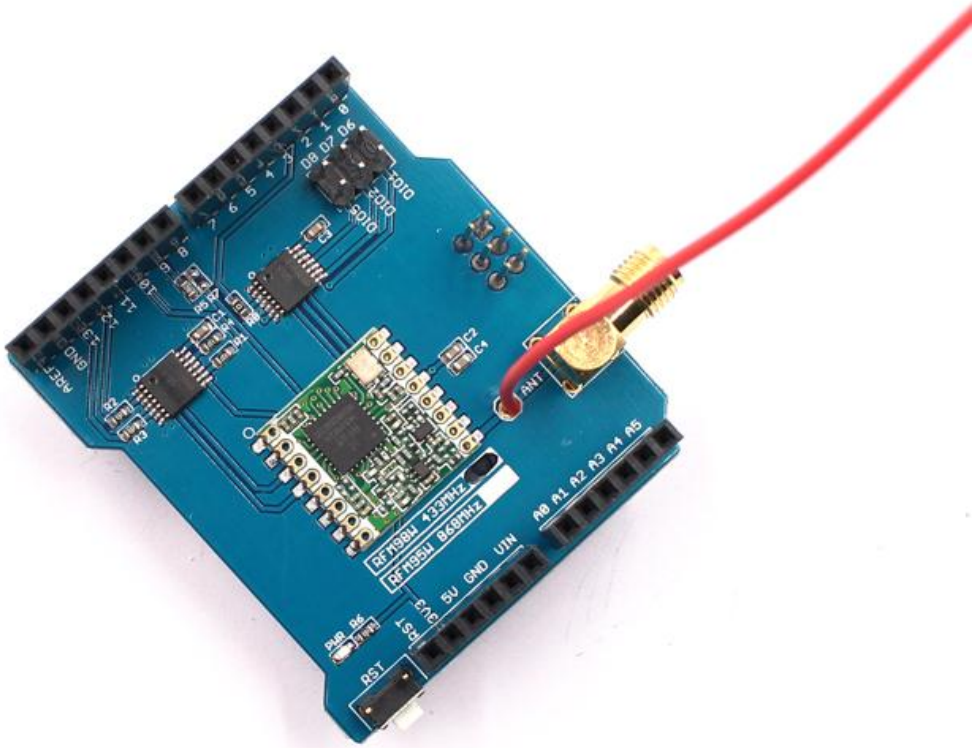
Usage of LoRa Radio Shield 433MHz /868MHz

The LoRa Shield allows the user to send data and reach extremely long ranges at low data-rates. It provides ultra-long range spread spectrum communication and high interference immunity whilst minimising current consumption.



Pins usage on Arduino

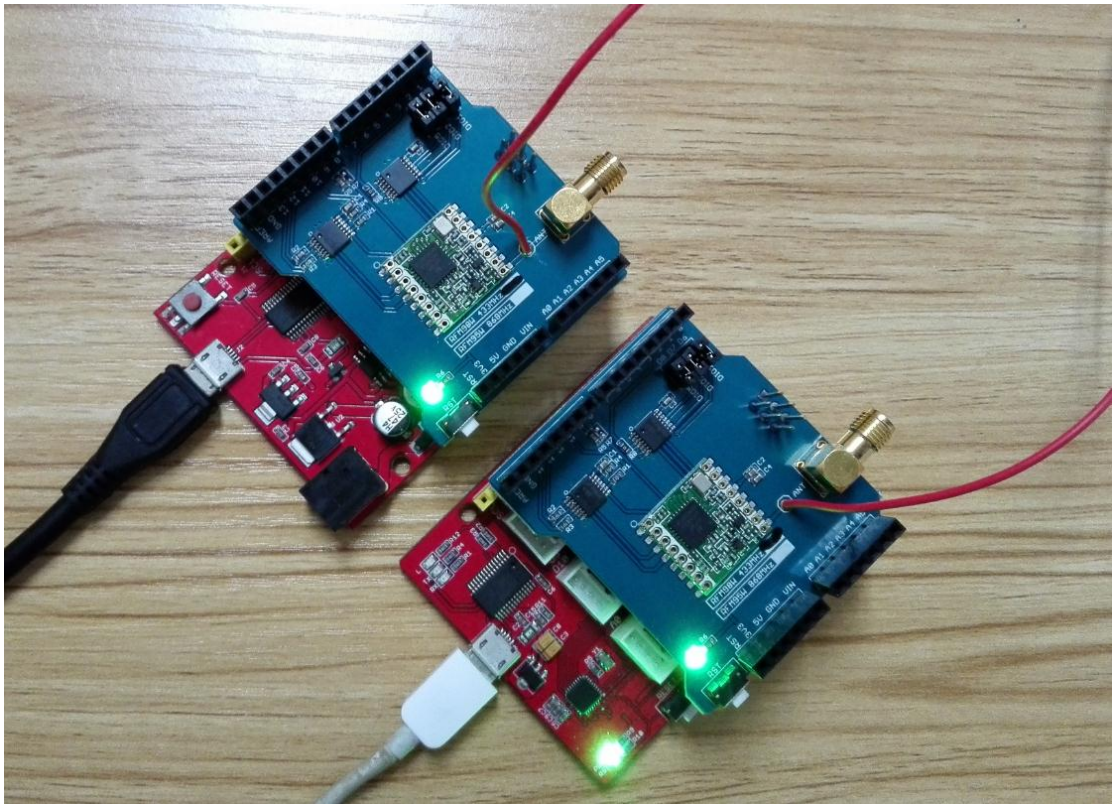
- **D2 - RFM95_INT**
- **D9 - RFM9x_RS**
- **D10 - RFM9x_CS**
- **D11 - RFM9x MOSI**
- **D12 - RFM9x MISO**
- **D13 - RFM9x SCK**

Arduino Library

Download the demo code form our website:
433MHz:
http://www.makerfabs.com/fabs/index.php?route=product/product&path=90&product_id=130
868MHz:
http://www.makerfabs.com/fabs/index.php?route=product/product&path=90&product_id=131
Place the **RadioHead** library folder your **arduino sketch folder/libraries/** folder.

Hardware connection:

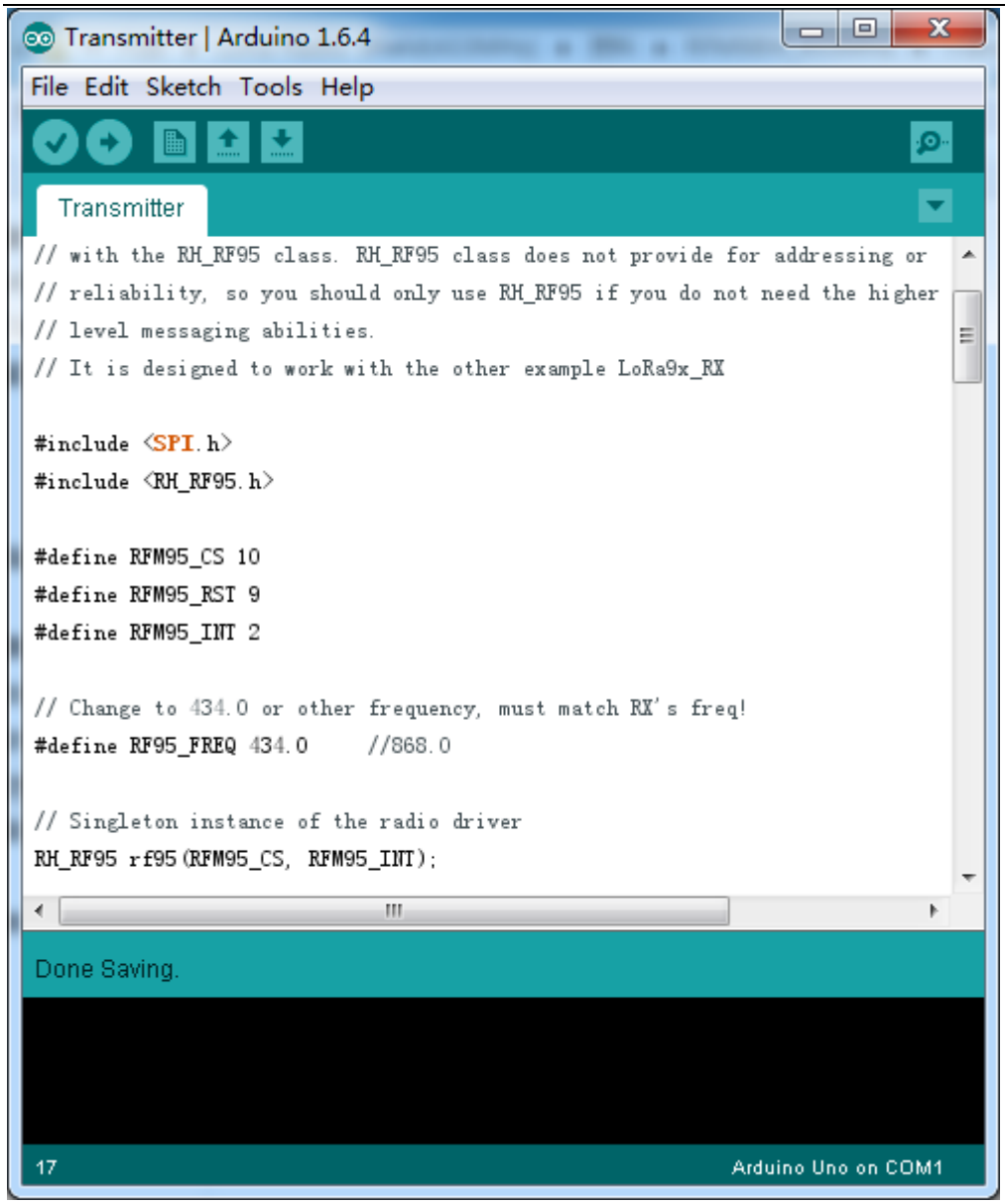
With this test, you need two Aruduino main boards. Just plug it to you Arduino or maduino. And upload the code, it will work ok.



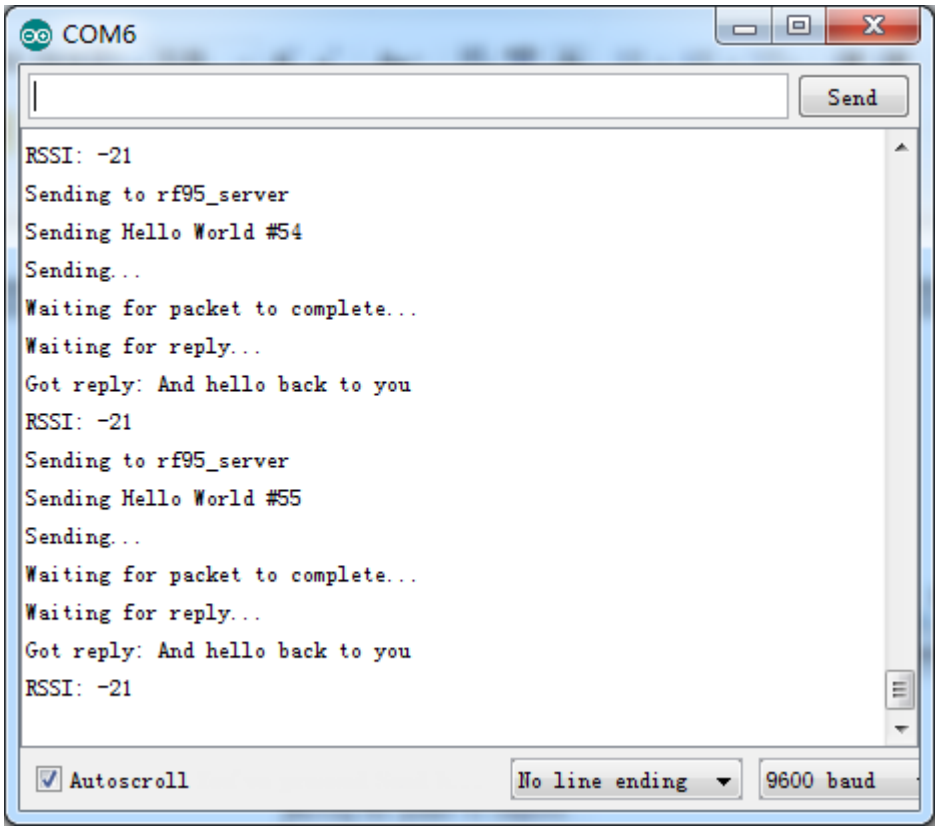
Basic RX & TX example

Let’s get a basic demo going, where one Arduino transmits and the other receives.

Transmitter example code

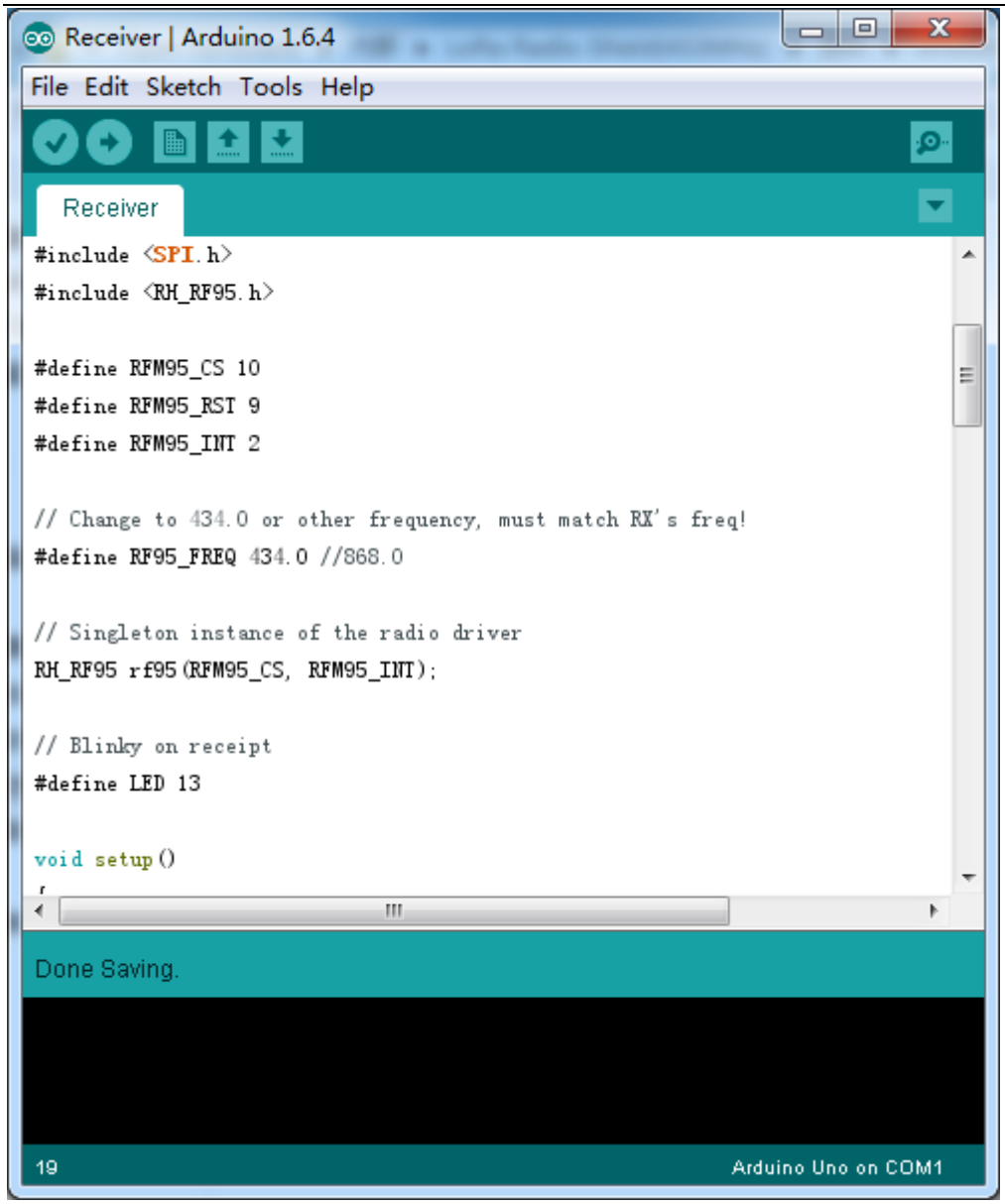


Once uploaded you should see the following on the serial console



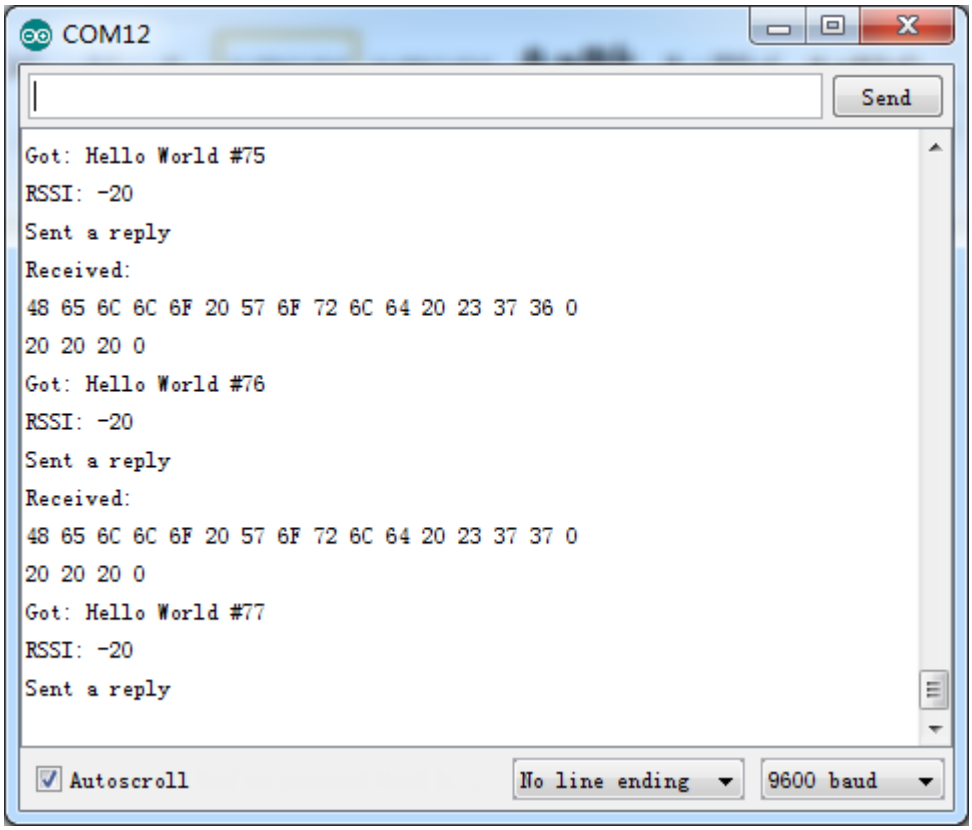
Now open up another instance of the Arduino IDE - this is so you can see the serial console output from the TX Arduino while you set up the RX Arduino.

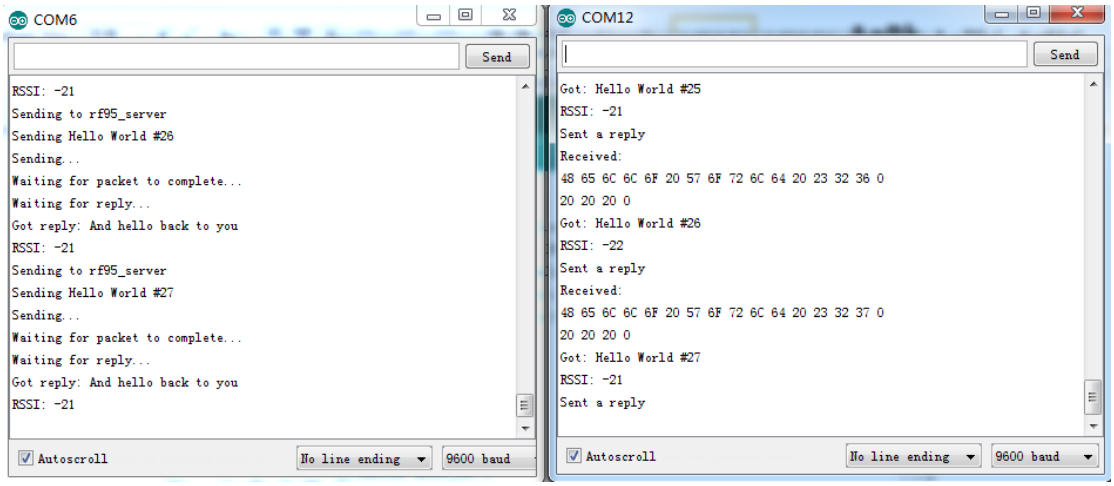
Receiver example code

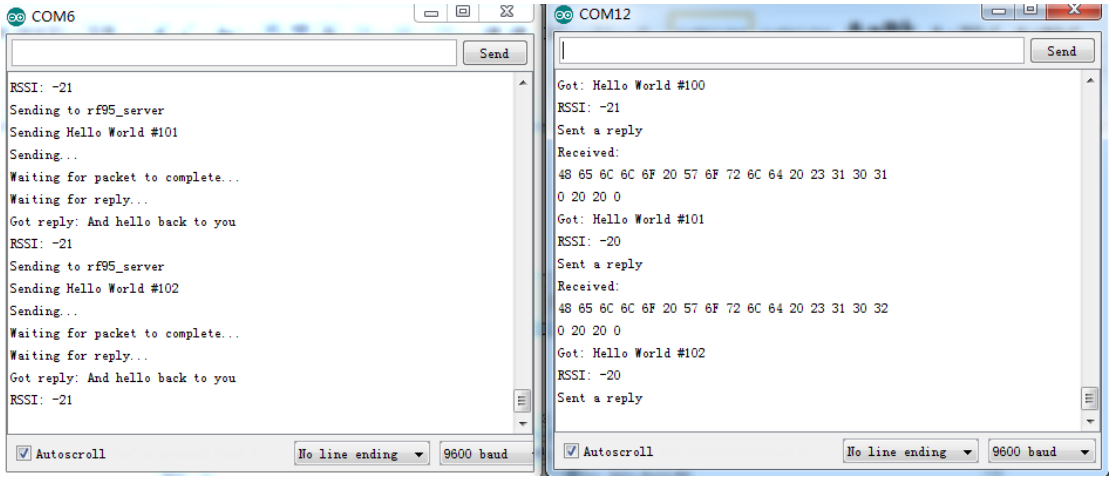


Upload the code to receiver maduino.

Now open up the Serial console on the receiver, while also checking in on the transmitter's serial console. You should see the receiver is...well, receiving packets


COM12
Send
Got: Hello World #75
RSSI: -20
Sent a reply
Received:
48 65 6C 6C 6F 20 57 6F 72 6C 64 20 23 37 36 0
20 20 20 0
Got: Hello World #76
RSSI: -20
Sent a reply
Received:
48 65 6C 6C 6F 20 57 6F 72 6C 64 20 23 37 37 0
20 20 20 0
Got: Hello World #77
RSSI: -20
Sent a reply
Autoscroll
No line ending
9600 baud


COM6
Send
RSSI: -21
Sending to rf95_server
Sending Hello World #26
Sending...
Waiting for packet to complete...
Waiting for reply...
Got reply: And hello back to you
RSSI: -21
Sending to rf95_server
Sending Hello World #27
Sending...
Waiting for packet to complete...
Waiting for reply...
Got reply: And hello back to you
RSSI: -21
Autoscroll
No line ending
9600 baud
COM12
Send
Got: Hello World #25
RSSI: -21
Sent a reply
Received:
48 65 6C 6C 6F 20 57 6F 72 6C 64 20 23 32 36 0
20 20 20 0
Got: Hello World #26
RSSI: -22
Sent a reply
Received:
48 65 6C 6C 6F 20 57 6F 72 6C 64 20 23 32 37 0
20 20 20 0
Got: Hello World #27
RSSI: -21
Sent a reply
Autoscroll
No line ending
9600 baud



You can see that the library example prints out the hex-bytes received **48 65 6C 6C 6F 20 57 6F 72 6C 64 20 23 35 37 32 0 20 20 0**, as well as the ASCII 'string' **Hello World.** Then it will send a reply.

And, on the transmitter side, it is now printing that it got a reply after each transmisssion **And hello back to you** because it got a reply from the receiver