

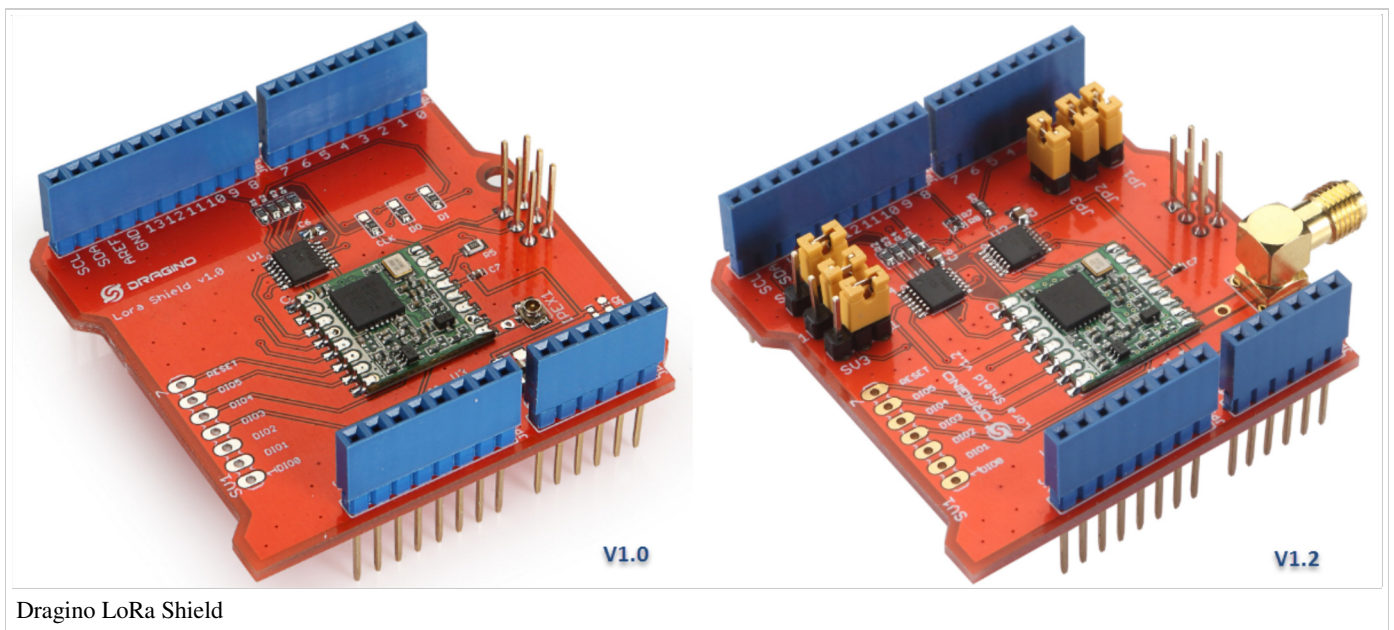
Lora Shield

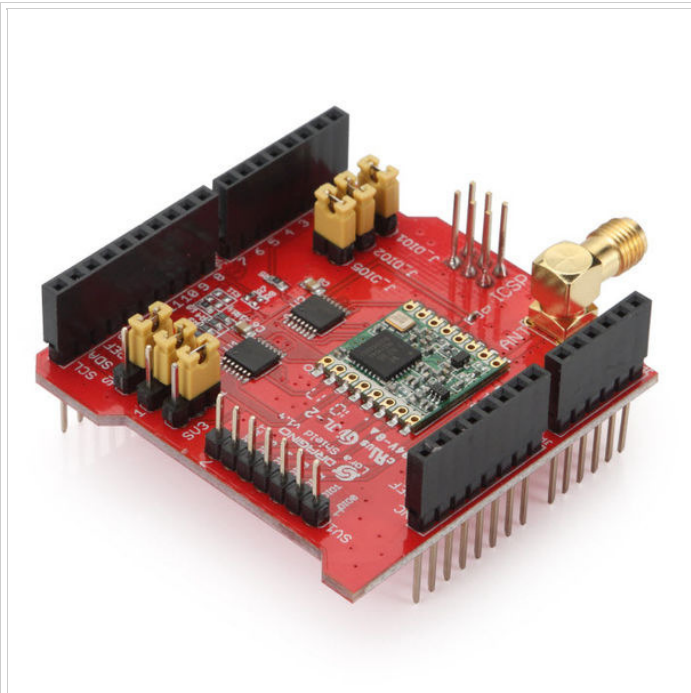
From Wiki for Dragino Project

Contents

- 1 INTRODUCTION
 - 1.1 What is the Dragino LoRa Shield
 - 1.2 Hardware version info
 - 1.3 Pin Mapping and Unused Pins
 - 1.4 Wireless Specifications of RFM95W
 - 1.5 Features
 - 1.6 Power Consumption
 - 1.7 Dimension
- 2 Use Examples
 - 2.1 Example1 -- Use with LMIC library for LoRaWAN compatible
 - 2.2 Example2 -- Use the RadioHead Library With Arduino Boards
 - 2.3 Example3 -- How to get sensor data from a remote Arduino via Wireless LoRa Protocol
 - 2.4 Example4 -- LMIC-RAW-915
 - 2.5 Example5 -- Set up as a LoRa Gateway using the LowCostLoRaGw library
 - 2.6 Example6 -- Upstream/Downstream to ThingSpeak IoT Server
 - 2.7 Example7 -- use the Dragino LoRa shield with Arduino MKR 1010
 - 2.8 Usage Notice
- 3 Order Infomation
- 4 Schematic
- 5 Frequent Ask Questions (FAQs)
 - 5.1 I have ordered LoRa Shield v95-868, while the chip on LoRa shield shows RF96?
 - 5.2 I want to use the LoRa Shield to connect to the Things Network by OTAA,how can I get the DevEUI?
- 6 Reference

INTRODUCTION





Dragino LoRa Shield v1.4

What is the Dragino LoRa Shield

The Dragino LoRa Shield is a long range transceiver on a Arduino shield form factor and based on Open source library. 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.

The LoRa Shield based on SX1276/SX1278 targets professional wireless sensor network applications such as irrigation systems, smart metering, smart cities, smartphone detection, building automation, and so on.

Using Semtech's patented LoRa™ modulation technique the LoRa Shield can achieve a sensitivity of over -148dBm using a low cost crystal and bill of materials. The high sensitivity combined with the integrated +20 dBm power amplifier yields industry leading link budget making it optimal for any application requiring range or robustness. LoRa™ also provides significant advantages in both blocking and selectivity over conventional modulation techniques, solving the traditional design compromise between range, interference immunity and energy consumption.

These devices also support high performance (G)FSK modes for systems including WMBus, IEEE802.15.4g. The LoRa Shield deliver exceptional phase noise, selectivity, receiver linearity and IIP3 for significantly lower current consumption than competing devices.



LoRa Shield with Leonardo

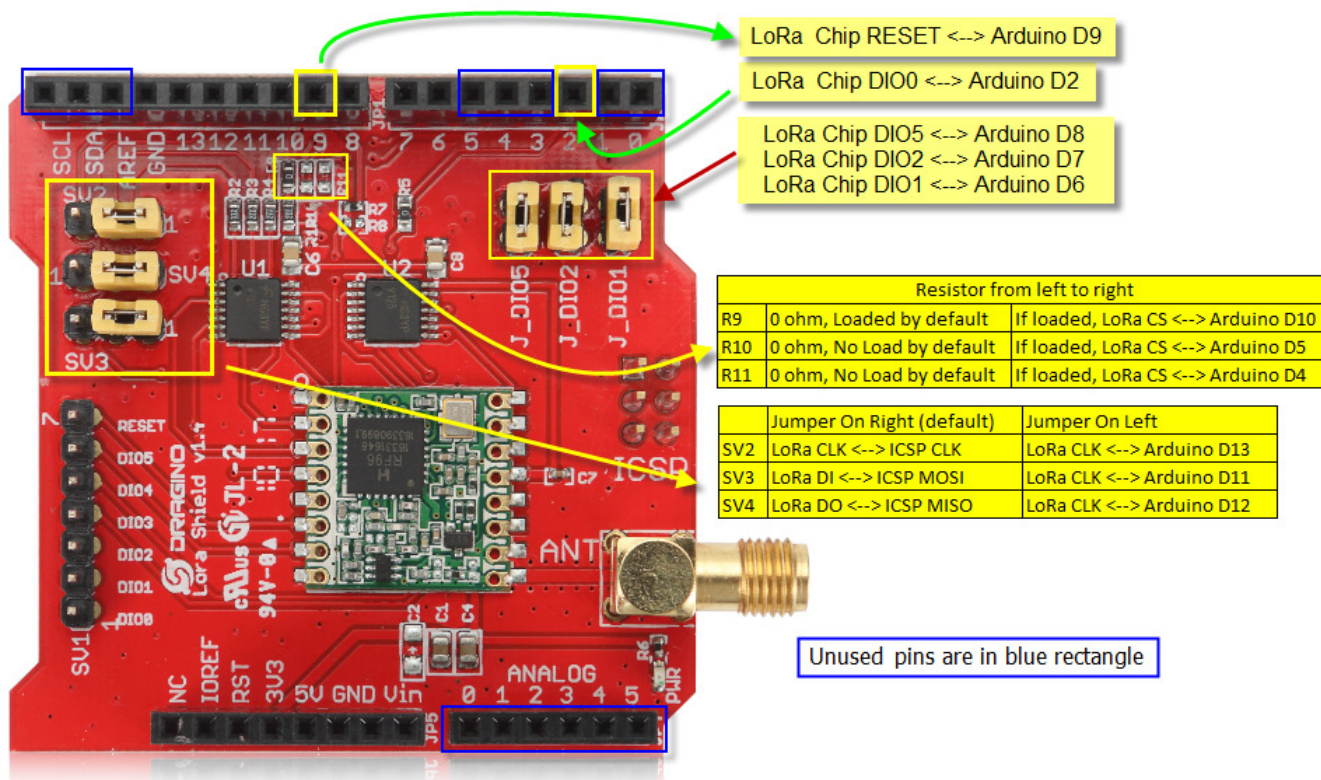
Hardware version info

Hardware Source is in: LoRa Shield hardware source (<https://github.com/dragino/Lora/tree/master/Lora%20Shield/hardware>).

- LoRa Shield v1.0: The first hardware release for the LoRa Shield.
- LoRa Shield v1.2: Connect the DIOs to Arduino IO. Change i-pex connector to SMA connector.
- LoRa Shield v1.3: Remove 5v to 3.3v LDO, use 3.3 directly from Arduino +3.3v pin.
- LoRa Shield v1.4: Add 3 resistors R9/R10/R11. Purpose is to provide possibility to change the Arduino pin connect to LoRa CS pin, so Arudino can easier to connect to multi SPI devices. by default, R9 is populated and R10/R11 is not populated. So LoRa CS is connected to Arduino D10.

Pin Mapping and Unused Pins

Pin Mapping For LoRa



LoRa Shield Pin Mapping

Wireless Specifications of RFM95W

- 168 dB maximum link budget.
- +20 dBm - 100 mW constant RF output vs.
- +14 dBm high efficiency PA.
- Programmable bit rate up to 300 kbps.
- High sensitivity: down to -148 dBm.
- Bullet-proof front end: IIP3 = -12.5 dBm.
- Excellent blocking immunity.
- Low RX current of 10.3 mA, 200 nA register retention.
- Fully integrated synthesizer with a resolution of 61 Hz.
- FSK, GFSK, MSK, GMSK, LoRaTM and OOK modulation.
- Built-in bit synchronizer for clock recovery.
- Preamble detection.
- 127 dB Dynamic Range RSSI.
- Automatic RF Sense and CAD with ultra-fast AFC.
- Packet engine up to 256 bytes with CRC.
- Built-in temperature sensor and low battery indicator.

Features

- Compatible with 3.3v or 5v I/O Arduino Board.
- Frequency Band: 915 MHZ/868 MHZ/433 MHZ (Pre-configure in factory)
- Low power consumption
- Compatible with Arduino Leonardo, Uno, Mega, DUE
- External Antenna via I-Pex connector

Power Consumption

Table 6 Power Consumption Specification

Symbol	Description	Conditions	Min	Typ	Max	Unit
IDDSL	Supply current in Sleep mode		-	0.2	1	uA
IDDIDLE	Supply current in Idle mode	RC oscillator enabled	-	1.5	-	uA
IDDST	Supply current in Standby mode	Crystal oscillator enabled	-	1.6	1.8	mA
IDDFS	Supply current in Synthesizer mode	FSRx	-	5.8	-	mA
IDDR	Supply current in Receive mode	<i>LnaBoost</i> Off, band 1	-	10.8	-	mA
		<i>LnaBoost</i> On, band 1	-	11.5	-	
		Bands 2&3	-	12.0	-	
IDDT	Supply current in Transmit mode with impedance matching	RFOP = +20 dBm, on PA_BOOST	-	120	-	mA
		RFOP = +17 dBm, on PA_BOOST	-	87	-	mA
		RFOP = +13 dBm, on RFO_LF/HF pin	-	29	-	mA
		RFOP = + 7 dBm, on RFO_LF/HF pin	-	20	-	mA

Power Consumption

Dimension

- Device Dimension: 62 x 43 x 23 mm
- Device Weight: 22g
- Package Dimension: 111 x 70 x 36 mm
- Package Weight: 53g

Use Examples

Example1 -- Use with LMIC library for LoRaWAN compatible

LMIC library is LoRaWAN compatible.

This demo include:

- How to use LoRa Shield to set up single channel gateway for TTN network.
- How to use LoRa Shield to set up a LoRa Node.
- How is the communication between the LoRa Node and LoRa Gateway.

Use the LMIC with **868MHZ** Dragino LoRa boards:

The full instruction is in this link: [1] (<http://www.instructables.com/id/Use-Lora-Shield-and-RPi-to-Build-a-LoRaWAN-Gateway/>)

Use the LMIC with **915MHZ** Dragino LoRa boards:

The operations similar to "Use the LMIC with 868MHZ Dragino Lora boards (<http://www.instructables.com/id/Use-Lora->

Shield-and-RPi-to-Build-a-LoRaWAN-Gateway/)".

You need to make following changes here:

1.Copy this code (https://github.com/dragino/Lora/blob/master/Lora%20Shield/Examples/lora_shield_ttn-915-fix-frequency/lora_shield_ttn-915-fix-frequency.ino) to you IDE and upload the sketch to your Arduino board:

There,you can also add below code to your sketch to check the **current emission frequency**:

```
Serial.println(LMIC.freq);
```

2.Edit "config.h"

change

```
#define CFG_eu868 1
```

to

```
#define CFG_us915 1
```

3.Edit "lmic.c",about on the line 760,

change

```
void LMIC_disableChannel (ul_t channel) {
    if( channel < 72+MAX_XCHANNELS )
        LMIC.channelMap[channel/4] &= ~(1<<(channel&0xF));
}
```

to

```
void LMIC_disableChannel (ul_t channel) {
    if( channel < 72+MAX_XCHANNELS )
        LMIC.channelMap[channel/16] &= ~(1<<(channel&0xF));
}
```

4.Edit the "main.cpp" to "Set center frequency" on your Raspberry Pi.

change

```
uint32_t freq = 868100000; //in Mhz! (868.1)
```

to

```
uint32_t freq = 915000000; //in Mhz! (915.0)
```

The other operations please refer to Use the LMIC with 868MHZ Dragino LoRa boards (<http://www.instructables.com/id/Use-Lora-Shield-and-RPi-to-Build-a-LoRaWAN-Gateway/>).

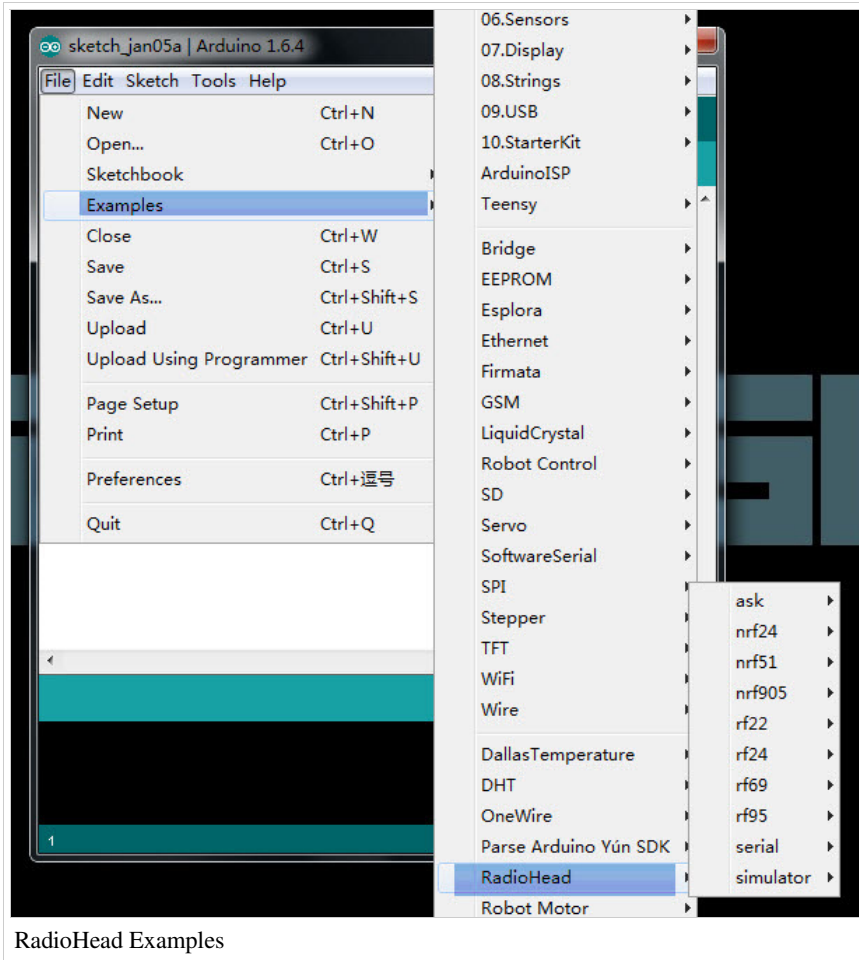
Example2 -- Use the RadioHead Library With Arduino Boards

The RadioHead Library simplifies communication between the Arduino Board and the LoRa Shield.We can find it here: <http://www.airspayce.com/mikem/arduino/RadioHead/> To use it in your project you just have to copy it in your Arduino project folder (next to your main Arduino project file .ino).

Compile Notice:

- Check if you have set the right frequency: After putting the library in the right place, you have to also modify the frequency to the frequency you want to use, the position of this is `setFrequency()` in the file: `arduino-xxx\libraries\RadioHead\RH_RF95.cpp`;
- The RF95-server sketch use the `led=9` to show the status. please change it to other LED , such 8. because 9 is connected to the LoRa module's RESET pin. you will see 'init fail' in the default server sketch

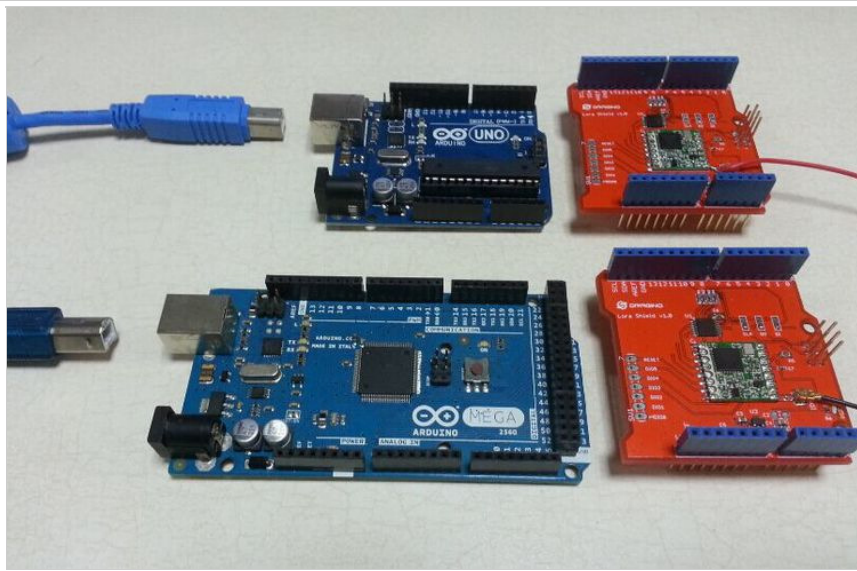
Then ,we can easily to use it to do some experimentation.



In this example, we use two LoRa Shields to transmit and receive signal, there are some project in the RadioHead Library, the operation is as follows:

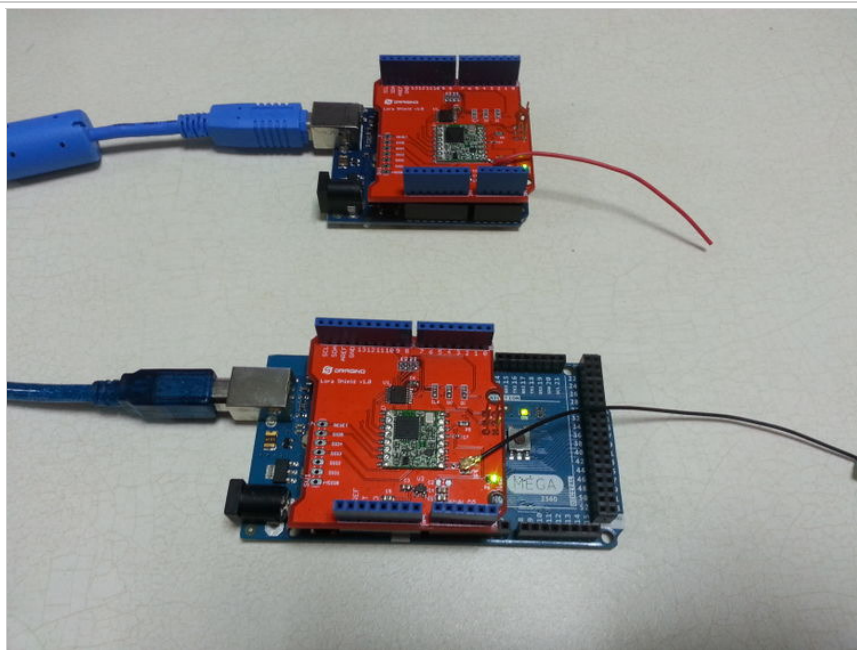
- Connect two LoRa Shields with Arduino board and connect them to computer via the USB cable;

We used the LoRa Shield+Arduino UNO as the server to transmit signal and the LoRa Shield+Arduino MEGA2560 as the client to receive signal here.



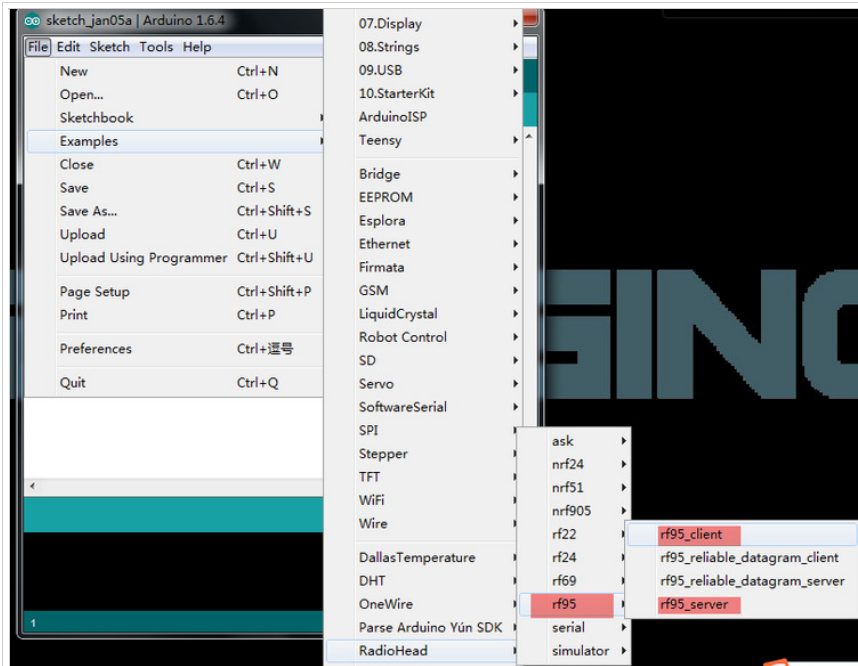
Lora Shields and Arduino boards

Connect the LoRa Shield and Arduino, and connect them to computer.



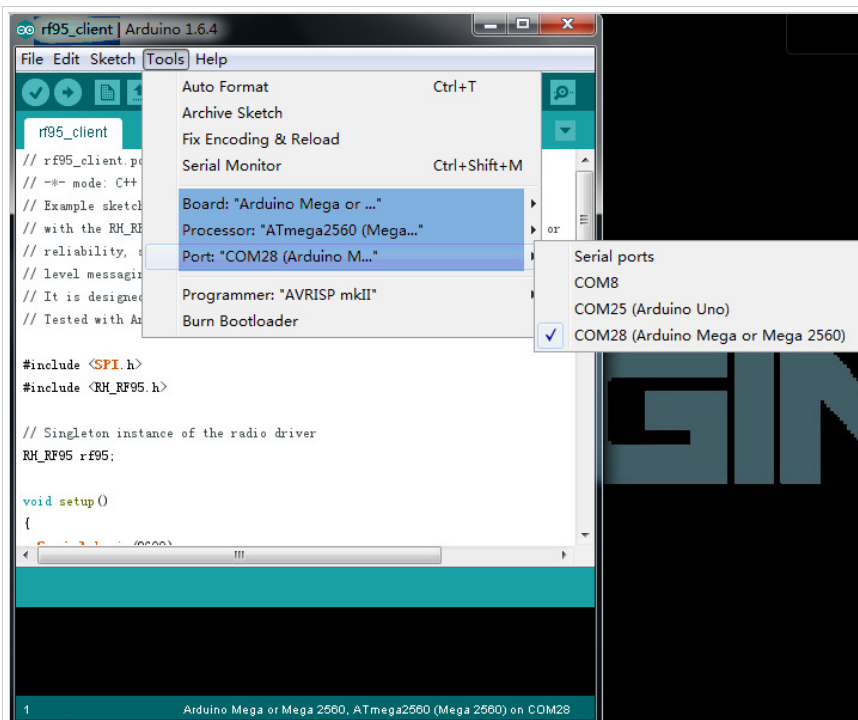
Connection

- Open the IDE and choose the right example;



Choose the example

- Configure the IDE settings and upload the two skteches to Aduino.



Configure the IDE settings of MEGA



The screenshot shows the Arduino IDE interface. The top window is titled "rf95_client | Arduino 1.6.4" and contains the sketch code. The bottom window is titled "COM28 (Arduino Mega or Mega 2560)" and shows the serial monitor output. The code in the top window is as follows:

```
// rf95_client.pde
// ... (code is partially obscured) ...
```

The serial monitor output shows a series of "got reply: And hello back to you" and "Sending to rf95_server" messages, indicating a successful communication between the two devices. The status bar at the bottom indicates the board is "Arduino Mega or Mega 2560, ATmega2560 (Mega 2560) on COM28".

Running result of MEGA

The screenshot shows the Arduino IDE interface. The top window is titled 'rf95_server | Arduino 1.6.4' and contains the following code:

```

File Edit Sketch Tools Help

[Icons]

rf95_server
// It is designed to work with the other example rf95_client
// Tested with Arduino MiniMicro:LoRa

```

The bottom window is titled 'COM25 (Arduino Uno)' and shows the serial monitor output. The text in the monitor is as follows:

```

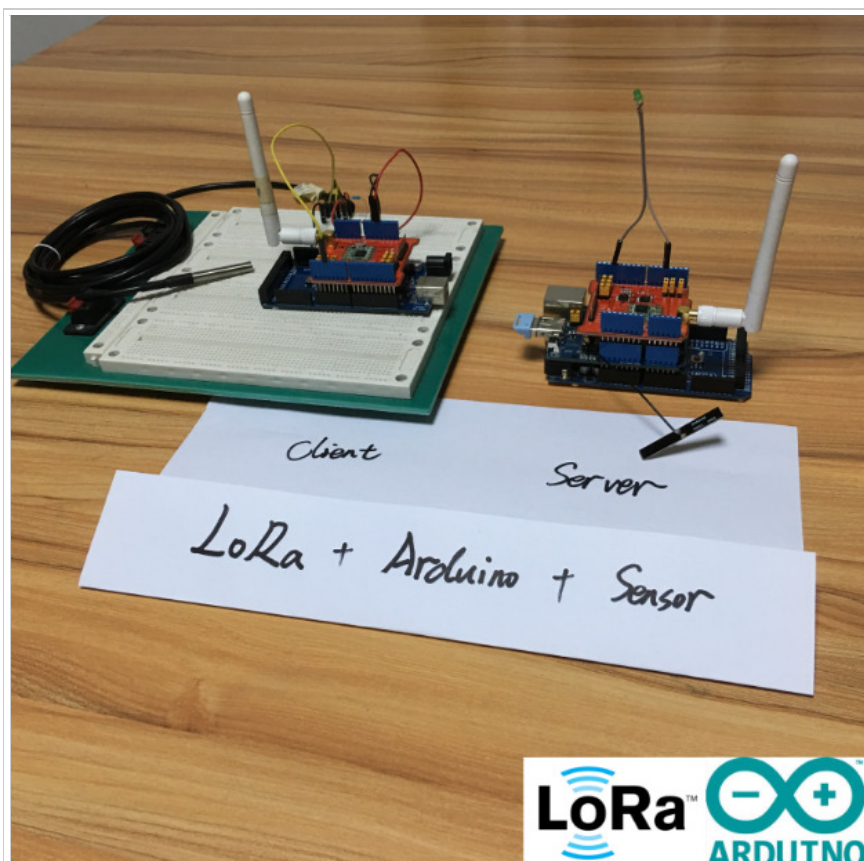
Sent a reply
got request: Hello World!
Sent a reply
got request: Hello World!
Sent a reply
got request: Hello World!
Sent a reply
got request: Hello World!
Sent a reply
got request: Hello World!
Sent a reply
got request: Hello World!
Sent a reply
got request: Hello World!
Sent a reply
got request: Hello World!
Sent a reply

```

At the bottom of the serial monitor window, there are settings for 'Autoscroll' (checked), 'No line ending', and '9600 baud'.

Running result of UNO

Example3 -- How to get sensor data from a remote Arduino via Wireless LoRa Protocol



LoRa Temperature Project

This is an example to show how to get sensor data from a remote Arduino via Wireless LoRa Protocol. The examples requires below hardwares:

- 1) Client Side: Arduino + LoRa Shield (868Mhz) + DS18B20 (Temperature Sensor).

2) Server Side: Arduino + LoRa Shield (868Mhz) + Yun Shield + USB flash.

make sure the USB flash has this file datalog.csv in the data directory of root.

Requie below software: Radiohead library from: [http://www.airspayce.com/mikem/arduino/RadioHead/...](http://www.airspayce.com/mikem/arduino/RadioHead/)

Client side will get the temperature and keep sending out to the server via LoRa wireless. Server side will listin on the LoRa wireless frequency, once it get the data from Client side, it will turn on the LED and log the sensor data to a USB flash.

The full instruction is in this link: <http://www.instructables.com/id/How-to-Get-Sensor-Data-From-a-Remote-Arduino-Via-W/>

Example4 -- LMIC-RAW-915

This example is for the **LoRa Shield 915** users. We use the library "arduino-lmic" in this project. Over here, we use the things below to complete this project.

- Dragino LoRa Shield(915) x 2
- Arduino board x 2
- Antenna (915MHZ) x 2
- USB cable
- Power supply

Connected the hardware, edit the "config.h" of LMIC library, change the "#define CFG_eu868 1" to "#define CFG_us915 1". This example requires DISABLE_INVERT_IQ_ON_RX to be set. **Uncomment** "#define DISABLE_INVERT_IQ_ON_RX" (line 59 of config.h), save settings.


```
7
8 // #define CFG_eu868 1
9 #define CFG_us915 1
```

LMIC config 1

```
57 // and/or when talking to the radio dll
58 // example).
59 #define DISABLE_INVERT_IQ_ON_RX
60 #endif // _lmic_config_h_
61
```

LMIC config 2

Open the IDE(1.6.8 here), choose the correct board and correct port, then upload the sketch "lmic-raw-915 (<https://github.com/dragino/Lora/blob/master/Lora%20Shield/Examples/lmic-raw-915/lmic-raw-915.ino>)" to the two arduino boards.



The screenshot shows the Arduino IDE interface with the 'raw-915' sketch loaded. The code is for an LMIC (LoRaWAN MAC) gateway. It includes a preprocessor check for `DISABLE_INVERT_IQ_ON_RX`, a comment about the LMIC duty cycle, and a pin mapping for the LMIC pins. The code is as follows:

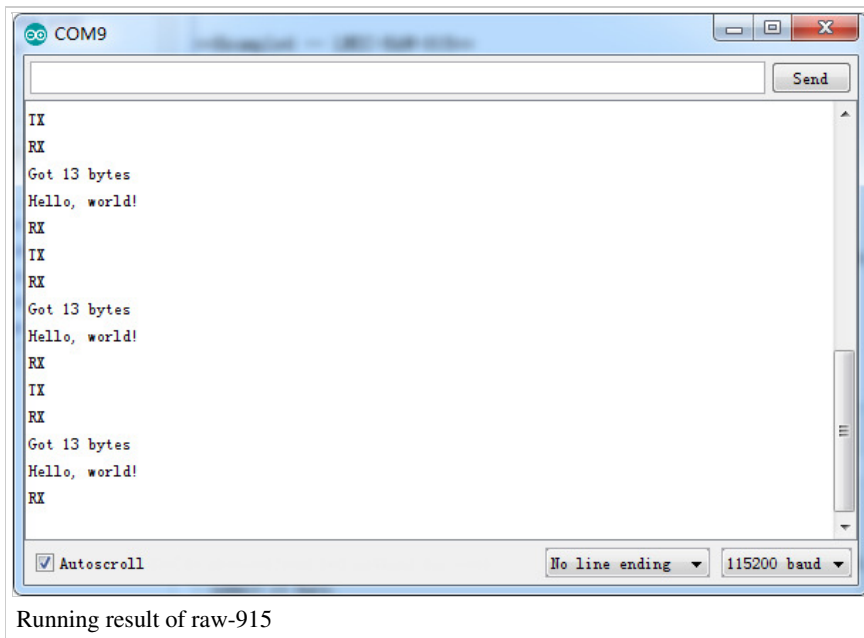
```
#if !defined(DISABLE_INVERT_IQ_ON_RX)
#error This example requires DISABLE_INVERT_IQ_ON_RX to be set. Update \
      config.h in the lmic library to set it.
#endif

// How often to send a packet. Note that this sketch bypasses the normal
// LMIC duty cycle limiting, so when you change anything in this sketch
// (payload length, frequency, spreading factor), be sure to check if
// this interval should not also be increased.
// See this spreadsheet for an easy airtime and duty cycle calculator:
// https://docs.google.com/spreadsheets/d/1voGAtQAjC1qBmaVuP1ApIKs1ekgUjavHuVQIXy7SvNc
#define TX_INTERVAL 2000

// Pin mapping
const lmic_pinmap lmic_pins = {
  .nss = 10,
  .rxtx = LMIC_UNUSED_PIN,
  .rst = 9,
  .dio = {2, 6, 7},
};
```

LMIC_RAW_915

Open the Serial Monitor and you can see the running result as below:



The screenshot shows the Serial Monitor window for COM9. The output shows the following sequence of events:

```
TX
RX
Got 13 bytes
Hello, world!
RX
TX
RX
Got 13 bytes
Hello, world!
RX
TX
RX
Got 13 bytes
Hello, world!
RX
```

The Serial Monitor settings are: Autoscroll (checked), No line ending, and 115200 baud.

Running result of raw-915

Example5 -- Set up as a LoRa Gateway using the LowCostLoRaGw library

The LowCostLoraGw source code (<https://github.com/CongducPham/LowCostLoRaGw>) provides good examples to shows how to use RPi to set data to the IoT provider and Dropbox. The LoRa Shield and LoRa and LoRa/GPS Hat can work with

these examples, by doing some modifications as below:

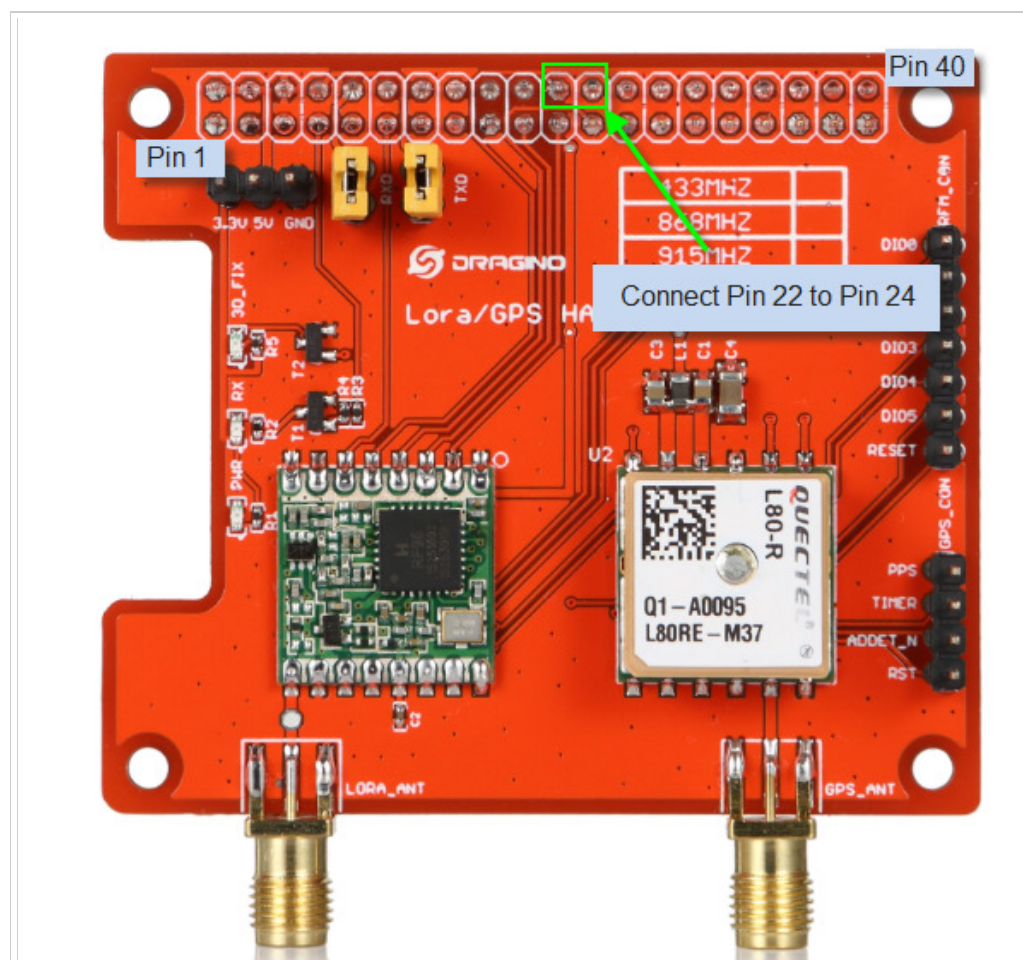
When use LoRa Shield with Arduino, what need to modify is:

```
In the sketch:
// uncomment if your radio is an HopeRF RFM92W or RFM95W
#define RADIO_RF92_95
```

```
In the code: SX1272.h
change #define SX1272_RST 3 to #define SX1272_RST 9
change #define SX1272_SS 2 to #define SX1272_SS 10
```

When use LoRa/GPS HAT with RPi 2/3, what need to modify is:

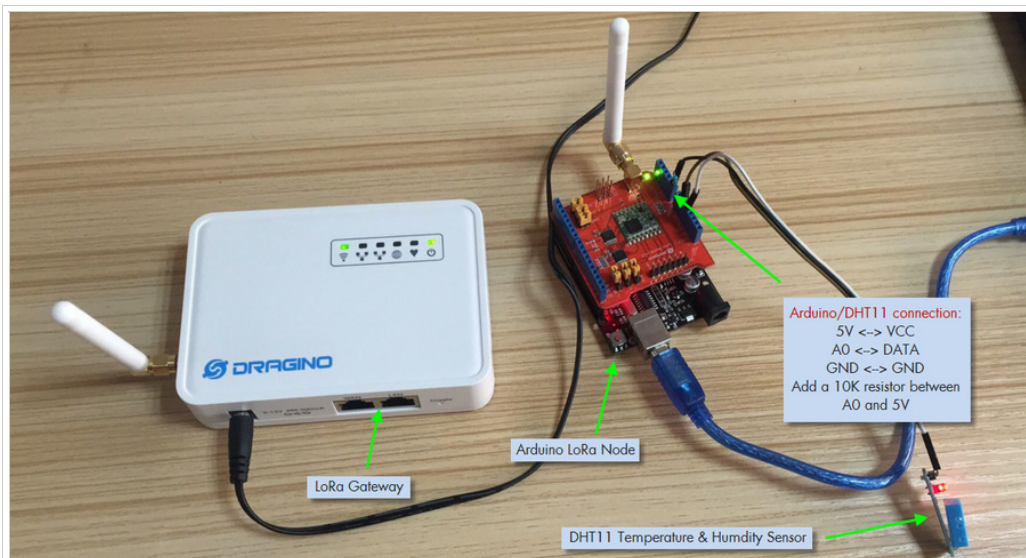
```
Solder the pin22 and pin24. for CS pin use in this library.
otherwise you will see below error:
pi@raspberrypi:~/lora_gateway$ sudo ./lora_gateway
Unrecognized transceiver.
....
```



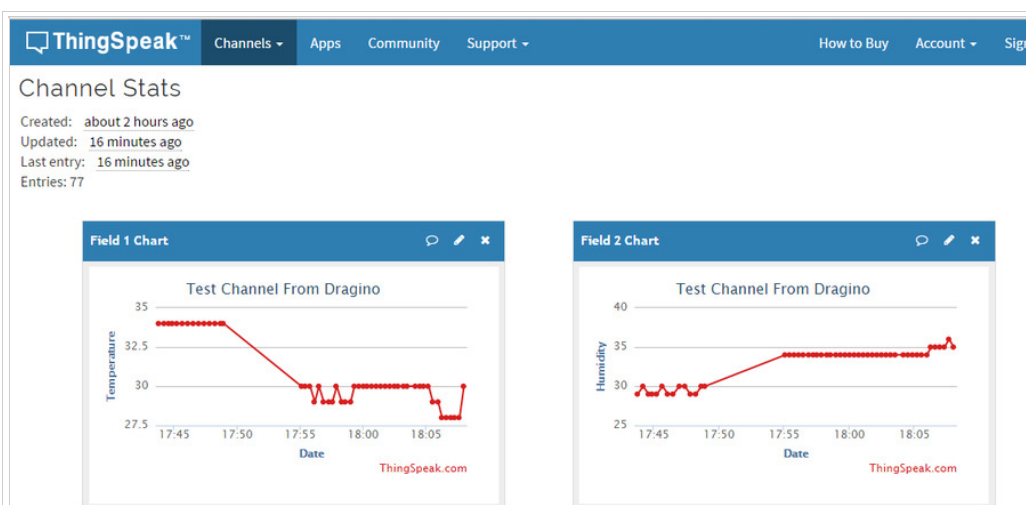
Use LoRa /GPS HAT with LowCostLoraGw source code

Example6 -- Upstream/Downstream to ThingSpeak IoT Server

Examples for ThingSpeak are in the LG01 user manual. Please click here (http://www.dragino.com/downloads/index.php?dir=UserManual/&file=LG01_LoRa_Gateway_User_Manual.pdf) for details.



ThingSpeak SetUp



ThingSpeak Test Result

The Examples (Source Code (<https://github.com/dragino/Arduino-Profile-Examples/tree/master/libraries/Dragino/examples/IoTServer/ThingSpeak>)) Include:

- How to upload get data from remote LoRa wireless node and upload it to ThingSpeak
- How to get commands in ThingSpeak and control a remote LoRa wireless node's relay

Example7 -- use the Dragino LoRa shield with Arduino MKR 1010

Example provided by:Robin Kluth

See the link from: use the Dragino LoRa shield with Arduino MKR 1010 (<http://www.dragino.com/downloads/index.php?dir=LoraShield/&file=How%20to%20use%20the%20Dragino%20LoRa%20shield%20with%20Arduino%20MKR%201010.docx>)

Usage Notice

You have to be aware that Radio link quality and performances are highly dependent of environment.

Better performances can be reached with:

- Outdoor environment.
- No obstacles.
- No high level radio interferer in the ISM 868MHz band.
- At least 1 meter above the ground.

Radio performances are degraded with:

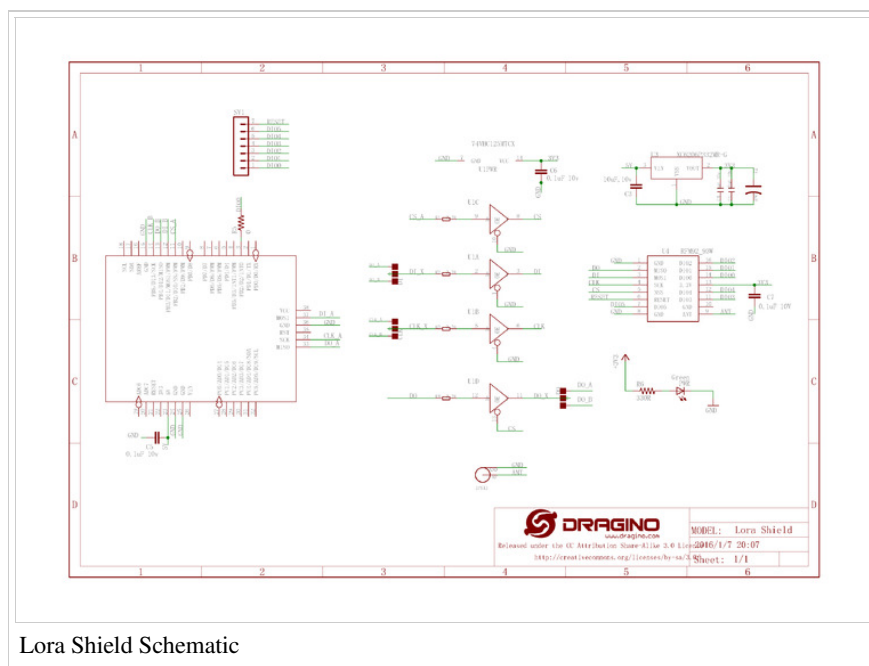
- Obstacles: buildings, trees...
- Inner buildings environments.
- High ISM 868MHz band usage by other technologies.

Radio communication are usually killed with bad topographic conditions. It is usually not possible to communicate through a hill, even very small.

Order Infomation

- LoRa Shield 868: support 868M frenquency
- LoRa Shield 915: support 915M frenquency
- LoRa Shield 433: support 433M frenquency

Schematic



Lora Shield Schematic

Frequent Ask Questions (FAQs)

I have ordered LoRa Shield v95-868, while the chip on LoRa shield shows RF96?

The LoRa Shield v95-868 is equied with HopeRF RFM95W-868-S2. This module has a mark RF96 on it. it doesn't mean the chip is RFM96w, but it means the chip is using the SX1276 chip.

Mean While, The LoRa Shield v95-915 S2 with mark RF96 on the IC. The LoRa Shield v98-433 S2 with mark RF98 on the IC and use the sx1278 chip.

I want to use the LoRa Shield to connect to the Things Network by OTAA,how can I get the DevEUI?

In the Arduino sketch with OTAA you need to setup APPEUI and DEVUI with LSB (less significant bit first), but APPKEY with MSB (more significant bit first). In the example skecth for OTAA the comments says that it doesn't matter if APPKEY if LSB or MSB, but it does, you must set it up with MSB!

And regarding DevEUI, I found that it depends on the LoRa transmitter: t RFM95W cannot give a unique DEVEUI so it's ok to use a random one.

Reference

- Hardware Source: <https://github.com/dragino/Lora/tree/master/Lora%20Shield>
- RoHS Report: http://www.dragino.com/downloads/index.php?dir=datasheet/RoHS/&file=Lora_Shield_RoHS_Report.pdf

Retrieved from "http://wiki.dragino.com/index.php?title=Lora_Shield&oldid=7208"

-
- This page was last modified on 18 October 2020, at 06:40.