

Lora Radio Shield

Contents

Introduction

Required

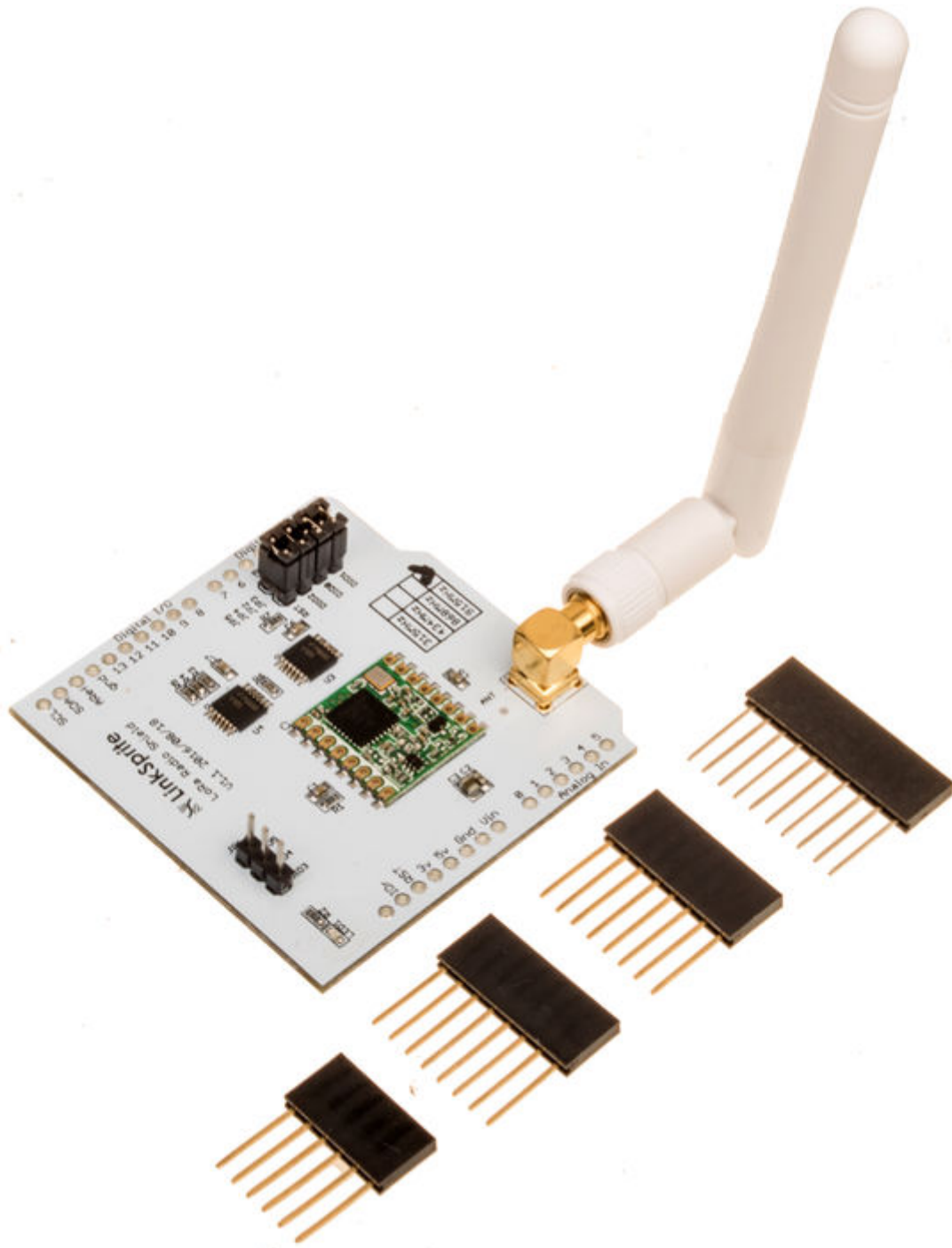
Steps

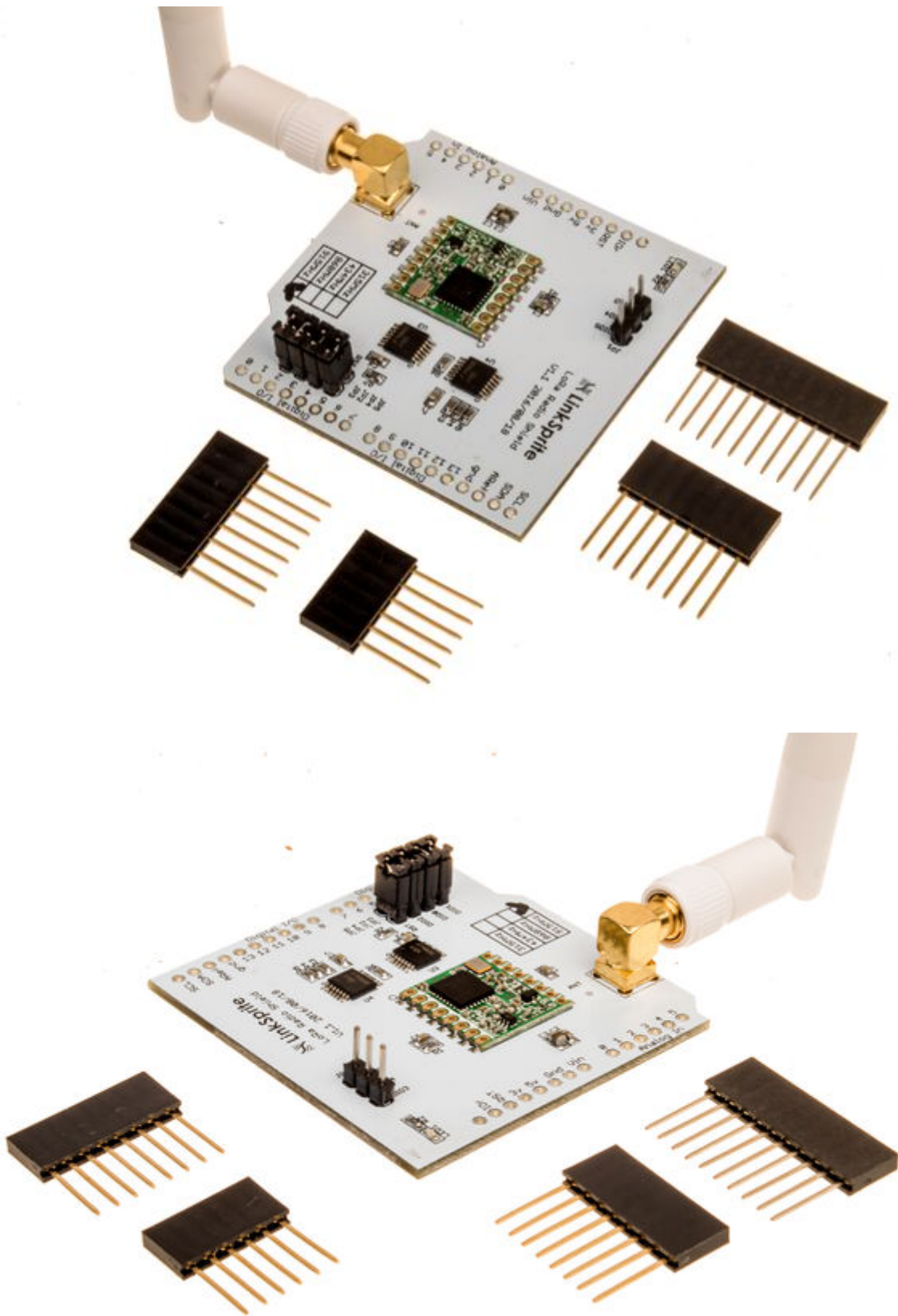
1. Assemble the hardware
2. Program Arduino Uno
3. Program on the pcDuino8 Uno
4. Test

Document

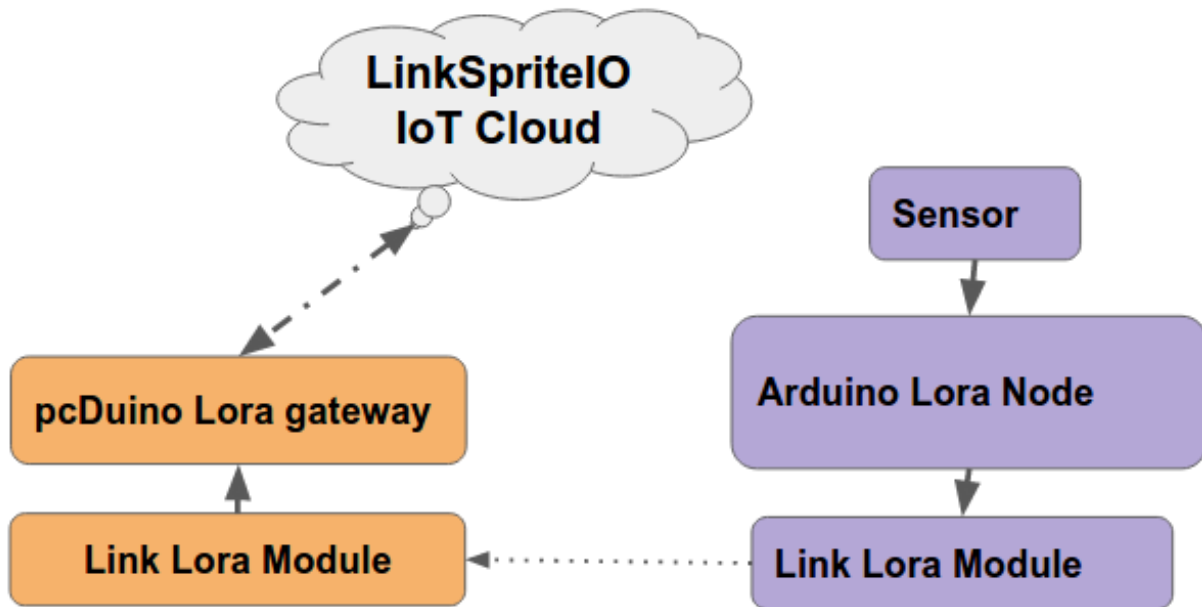
Introduction







Thanks to this [open source project \(http://cpham.perso.univ-pau.fr/LORA/RPIgateway.html\)](http://cpham.perso.univ-pau.fr/LORA/RPIgateway.html) which uses the Raspberry Pi as Lora gateway and Arduino as Lora node device. We port this project to pcDuino and send the received data from Lora node to [LinkSpriteIO \(https://github.com/delongqilinksprite/peduino-lora-AP/blob/master/www.linksprite.io\)](https://github.com/delongqilinksprite/peduino-lora-AP/blob/master/www.linksprite.io). The basic architecture is shown as the following diagram.



This Lora Radio Shield is base on Low-cost LoRa Module, learn more about [Low-cost LoRa Module \(http://www.linksprite.com/wiki/index.php5?title=Low-cost_LoRa_Module\)](http://www.linksprite.com/wiki/index.php5?title=Low-cost_LoRa_Module)

There is an Arduino Lora node to read the sensor's data and send it to pcDuino Lora gateway. After pcDuino Lora gateway receive the data and it will send it to LinkSpriteIO which is our IoT cloud.

The following content, I will introduce the details of how to DIY a low-cost Lora gateway with pcDuino and Arduino.

Required

pcDuino lora gateway

- pcDuino8 Uno (http://linksprite.com/wiki/index.php5?title=PcDuino8_Uno) x 1
- LoRa Radio Shield x 1

Arduino lora node

- Arduino Uno x 1
- LoRa Radio Shield x 1

Steps

1. Assemble the hardware

- According to the following pins map table to connect the lora module and pcDuino or Arduino

Lora module pins	Arduino Pins	pcDuino Pins
SCK	13	13
MISO	12	12
MOSI	11	11
NSS	10	10
RST	5	5
VCC	3.3V	3.3V
GND	GND	GND

Arduino Lora node

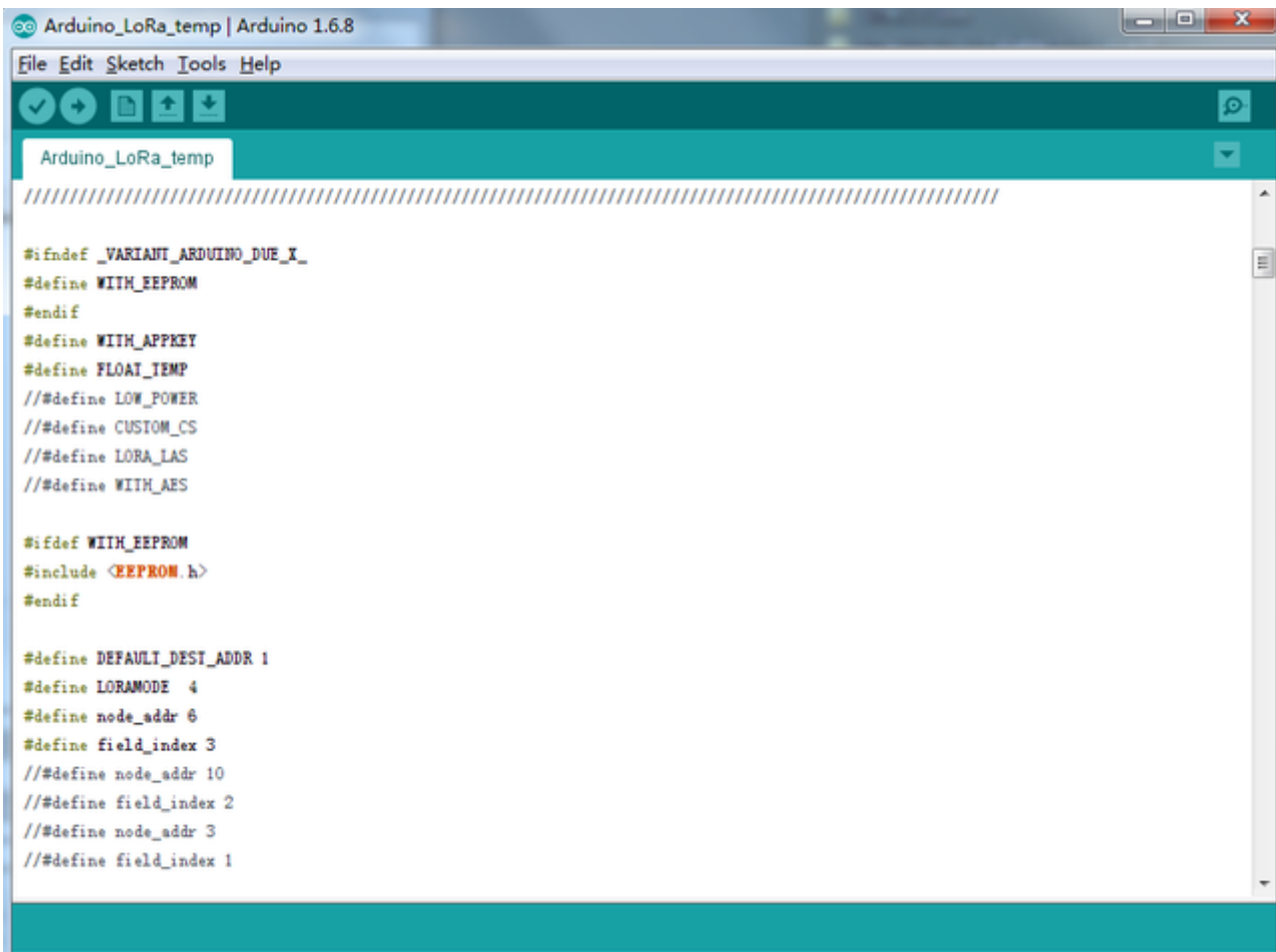


pcDuino Lora Gateway



2. Program Arduino Uno

- Download the Arduino program from [github \(https://github.com/YaoQ/pcduino-lora-AP/tree/master/arduino-node\)](https://github.com/YaoQ/pcduino-lora-AP/tree/master/arduino-node)
- Use Arduino IDE to open the Arduino_LoRa_node project in examples folder
- Upload this program to Arduino Uno
- Open Serial Monitor to check the message



```

Arduino_LoRa_temp | Arduino 1.6.8
File Edit Sketch Tools Help

Arduino_LoRa_temp

////////////////////////////////////

#ifndef _VARIANT_ARDUINO_DUE_X_
#define WITH_EEPROM
#endif

#define WITH_APPKEY
#define FLOAT_TEMP
// #define LOW_POWER
// #define CUSTOM_CS
// #define LORA_LAS
// #define WITH_AES

#ifdef WITH_EEPROM
#include <EEPROM.h>
#endif

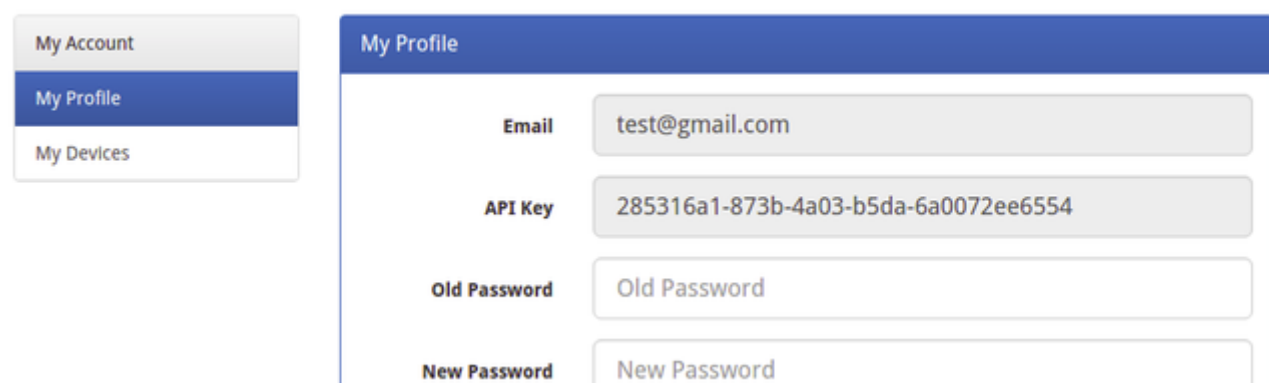
#define DEFAULT_DEST_ADDR 1
#define LORAMODE 4
#define node_addr 6
#define field_index 3
// #define node_addr 10
// #define field_index 2
// #define node_addr 3
// #define field_index 1

```

3. Program on the pcDuino8 Uno

Create device on [LinkSpriteIO \(http://www.linksprite.io/\)](http://www.linksprite.io/)

- Go to www.linksprite.io and sign up
- Enter your Email and password to create a new account
- Go to My Account to get your own API Key.



My Profile	
Email	test@gmail.com
API Key	285316a1-873b-4a03-b5da-6a0072ee6554
Old Password	Old Password
New Password	New Password

- Click My Device, and choose Create DIY Device.

Create New Device

Device Name

Device Type

Group Name

- Click the created device icon and get the DeviceID **.

temp

Device ID

Device Name

Group Name

Download the source code

- Access to the pcDuino8 Uno Ubuntu system Note: user name and password are all: linaro:

```
git clone https://github.com/YaoQ/pcduino-lora-AP
cd pcduino-lora-AP/pcduino-gateway
make
```

- Use your own deviceID and apikey to update the line 27 and 28 in LinkSpriteIO_Post.py

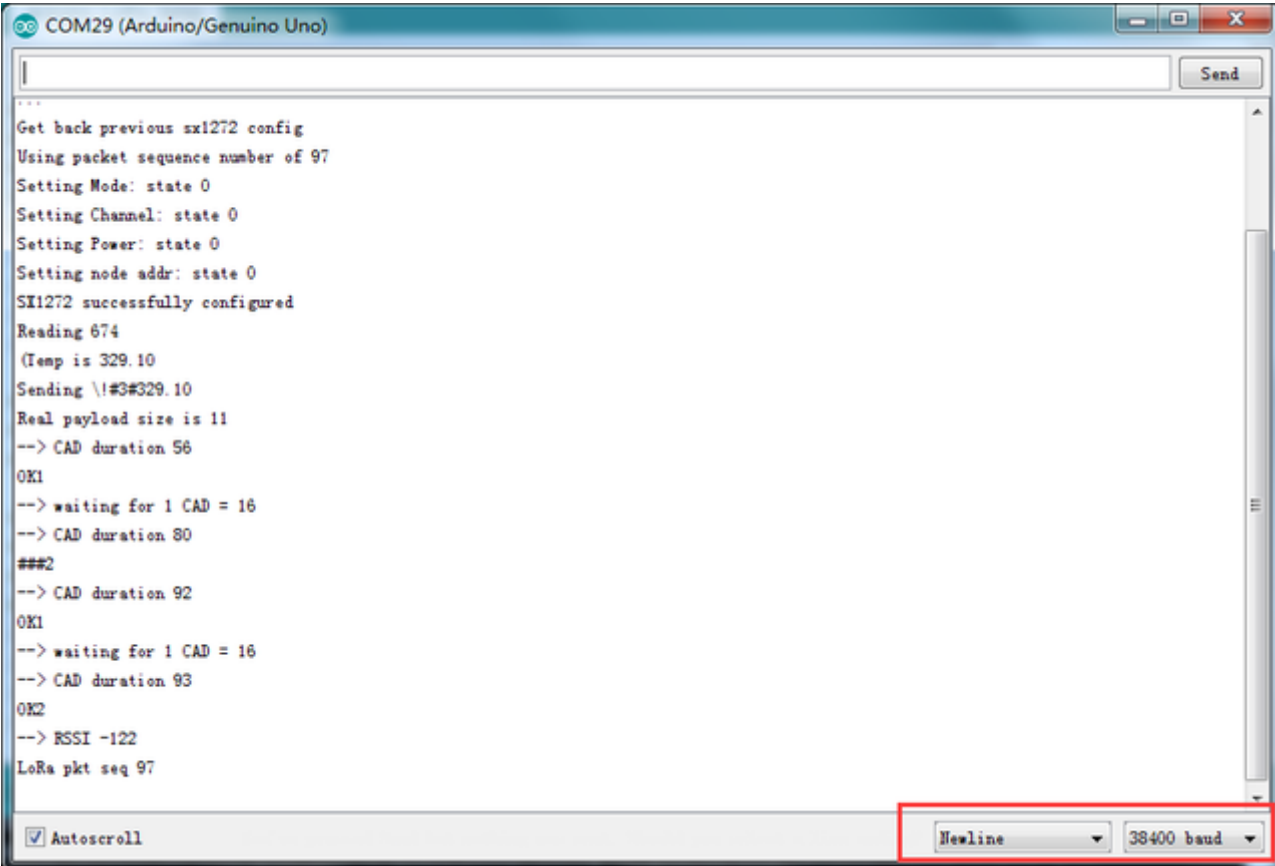
```
26 # Use your own deviceID and apikey
27 deviceID="xxxxxxxxx"
28 apikey = "xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx"
```

4. Test

To make it simple, we just use random number to simulate the temperature value and send it periodically to pcDuino lora gateway. Then pcDuino Lora gateway will post data to LinkSpriteIO.

```
./pcduino-gateway | python LinkSpriteIO_Post.py
```

On Arduino side:



```
...  
Get back previous sx1272 config  
Using packet sequence number of 97  
Setting Mode: state 0  
Setting Channel: state 0  
Setting Power: state 0  
Setting node addr: state 0  
SX1272 successfully configured  
Reading 674  
(Temp is 329.10  
Sending \!#329.10  
Real payload size is 11  
--> CAD duration 56  
OK1  
--> waiting for 1 CAD = 16  
--> CAD duration 80  
###2  
--> CAD duration 92  
OK1  
--> waiting for 1 CAD = 16  
--> CAD duration 93  
OK2  
--> RSSI -122  
LoRa pkt seq 97
```

Autoscroll Newline 38400 baud

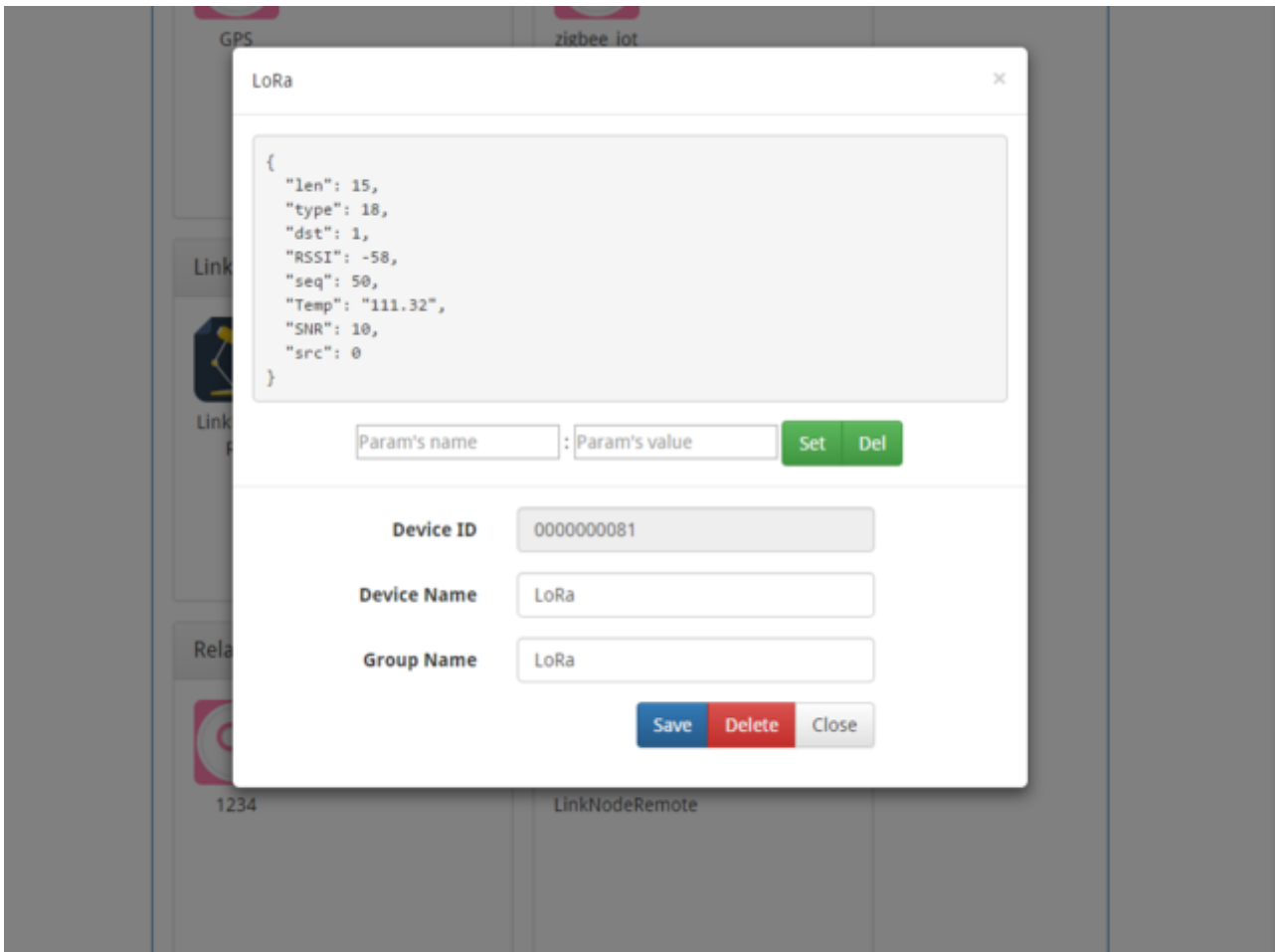
On pcDuino side:

```

root@linaro-alip:/derron/pduino-gateway#
root@linaro-alip:/derron/pduino-gateway# vi LinkSpriteIO_Post.py
root@linaro-alip:/derron/pduino-gateway# ./pduino-gateway | python LinkSpriteIO_Post.py
SX1276 detected, starting.
SX1276 LF/HF calibration
...
*****Power ON: state 0
Default sync word: 0x12
LoRa mode 4
Setting mode: state 0
Channel CH_10_868: state 0
Set LoRa Power to M
Power: state 0
Get Preamble Length: state 0
Preamble Length: 8
LoRa addr 1: state 0
SX1272/76 configured as LR-BS. Waiting RF input for transparent RF-serial bridge
--- rxlor. dst=1 type=0x12 src=6 seq=44 len=14 SNR=9 RSSIpkt=-58 BW=500 CR=4/5 SF=12
1970-01-01T05:56:54.062438
splitted in: [1, 18, 6, 44, 14, 9, -58]
(dst=1 type=0x12(DATA WAPPKEY) src=0 seq=44 len=14 SNR=9 RSSI=-58)
1970-01-01T05:56:54.059
yþ
-----
recieve value is:124.2
-----
--- rxlor. dst=1 type=0x12 src=6 seq=45 len=15 SNR=9 RSSIpkt=-59 BW=500 CR=4/5 SF=12
1970-01-01T05:57:01.213453
splitted in: [1, 18, 6, 45, 15, 9, -59]
(dst=1 type=0x12(DATA WAPPKEY) src=0 seq=45 len=15 SNR=9 RSSI=-59)
1970-01-01T05:57:01.212
yþ
-----
recieve value is:111.32
-----
--- rxlor. dst=1 type=0x12 src=6 seq=46 len=14 SNR=9 RSSIpkt=-58 BW=500 CR=4/5 SF=12
1970-01-01T05:57:07.419656
splitted in: [1, 18, 6, 46, 14, 9, -58]
(dst=1 type=0x12(DATA WAPPKEY) src=0 seq=46 len=14 SNR=9 RSSI=-58)
1970-01-01T05:57:07.418
yþ
-----
recieve value is:121.9
-----
--- rxlor. dst=1 type=0x12 src=6 seq=48 len=14 SNR=10 RSSIpkt=-58 BW=500 CR=4/5 SF=12
1970-01-01T05:57:18.565191
splitted in: [1, 18, 6, 48, 14, 10, -58]
(dst=1 type=0x12(DATA WAPPKEY) src=0 seq=48 len=14 SNR=10 RSSI=-58)
1970-01-01T05:57:18.564
yþ
-----
recieve value is:124.2
-----

```

On LinkSpriteIO side:



Document

LoRa Radio Shield V1.0.pdf (<https://s3.amazonaws.com/cutedigi/LoRa++Radio+Shield+V1.0/LoRa++Radio+Shield+V1.0.pdf>)

Retrieved from "http://linksprite.com/wiki/index.php?title=Lora_Radio_Shield&oldid=11985"

This page was last edited on 2 March 2017, at 06:28.