

Simple Prototype of LoRa Communications

As defined by Semtech, LoRa® is a *wireless technology developed to create the low-power, wide-area networks (LPWANs) required for machine-to-machine (M2M) and Internet of Things (IoT) applications. The technology offers a very compelling mix of long range, low power consumption and secure data transmission and is gaining significant traction in IoT networks being deployed by wireless network operators.*

In the following, we present a prototype of LoRa communication between two wireless modules. This enables to get basic insights on the implementation of the technology and prepares some advanced prototypes and experimentations.

-. Hardware and Cost

- Arduino Uno or Mega (x2) for [10 USD](#).
- Dragino shields (x2) for [20 USD](#).
- 868 MHz Antenna (x2) for [5 USD](#).

-. Software

- RadioHead: The Packet Radio library for embedded microprocessors can be downloaded from <http://www.airspayce.com/mikem/arduino/RadioHead/> or from this [direct link](#).
- Arduino IDE: Specific OS versions can be downloaded from <https://www.arduino.cc/en/Main/Software>.

-. Compatibility

This tutorial is written for :

- Arduino IDE 1.6.9.
- RadioHead 1.61.
- Dragino hardware release v1.3.
- Mac OS 10.11.5.

Beware of the necessary modifications, if any of the previous specifications is not supported in your case.

-. Installation

Start by plugging the Dragino shields on the Arduino devices and mounting the antennas as shown in Fig. 1.



Figure 1. Arduino with LoRa Dragino shield.

Connect the two Arduino devices to USB ports on your computer. If this is the first time you use Arduino IDE, make sure to install the necessary USB drivers by selecting **Tools > Boards Manager** and installing Arduino AVR boards.

Now, you have to select the corresponding Board and Port in the **Tools** menu to program your Arduino. For example, you can use the following:

- For Arduino Uno: Board = Arduino/Genuino Uno Mega, Port = /dev/cu.wchusbserial1420.
- For Arduino Mega: Board = Arduino/Genuino Mega 2560, Port = /dev/cu.usbmodem141511.

Unzip the RadioHead library and copy it to your sketchbook library folder as detailed in <https://www.arduino.cc/en/Guide/Libraries>.

- Running Basic Sketches

Start by setting the central frequency of the LoRa modules. For this, open the `RH_RF95.cpp` file locate in the RadioHead folder and change the frequency to 868.1 Mhz:

[RH_RF95.cpp](#)

```
setFrequency(868.1);
```

[Download the](#)

basic sketches

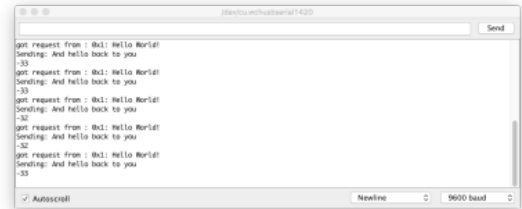
that implement a reliable LoRa communication between the two modules. Open the sketches with Arduino IDE, compile and upload on the two arduino modules, respectively. On the serial interfaces, you should obtain similar results as in Fig. 2.



```

Sending to rf95_reliable_datagram_server
got reply from : @x2: And hello back to you
-21
Sending to rf95_reliable_datagram_server
got reply from : @x2: And hello back to you
-22
Sending to rf95_reliable_datagram_server
got reply from : @x2: And hello back to you
-22
Sending to rf95_reliable_datagram_server
got reply from : @x2: And hello back to you
-22
Sending to rf95_reliable_datagram_server
got reply from : @x2: And hello back to you
-22

```



```

got request from : @x1: hello World!
Sending: And hello back to you
-33
got request from : @x1: hello World!
Sending: And hello back to you
-33
got request from : @x1: hello World!
Sending: And hello back to you
-32
got request from : @x1: hello World!
Sending: And hello back to you
-32
got request from : @x1: hello World!
Sending: And hello back to you
-33

```

-. Modifying the Radio Parameters

Bw125Cr48Sf4096 Bw = 125 kHz, Cr = 4/8, Sf = 4096chips/symbol, CRC on. Slow+long range

[RH_RF95.cpp](#)

```

//setModemConfig(Bw125Cr45Sf128); // Radio default
setModemConfig(Bw125Cr48Sf4096); // slow and reliable?

```

[RHReliableDatagram.h](#)

```

#define RH_LONG_TIMEOUT 8000

```

[RHReliableDatagram.cpp](#)

```

RHReliableDatagram::RHReliableDatagram(RHGenericDriver& driver, uint8_t
thisAddress)
    : RHDatagram(driver, thisAddress)
{
    _retransmissions = 0;
    _lastSequenceNumber = 0;
    //_timeout = RH_DEFAULT_TIMEOUT;
    _timeout = RH_LONG_TIMEOUT;
    _retries = RH_DEFAULT_RETRIES;
}

```

From:
<http://wiki.lahoud.fr/> - **wikiroute**

Permanent link:
http://wiki.lahoud.fr/doku.php?id=reliable_lora_communication&rev=1475224472

Last update: **2016/09/30 10:34**

