# LoRa Arduino library short user guide

Library is composed by 2 class:

1. LORA class : contains function to define and use simple LoRa communication and network
2. SX (external name) class. The SX1278 class contains several basic functions to manage radio module. This class is used by major class LORA, but can be used autonomously.

Actually another class REMOTEC is included. But this class uses LoRa module as simple amplitude modulation (OOK) for remote controlled sockets based on HX22xx (or similar) (like Avidsen or Velleman). Two examples : RemPwr and RemPwrScanner are also included in examples folder for these purposes.

**Use LORA**

**Definitions:**

1. Include library: #include <LoRa.h>
2. Class instance ex.: LORA LR;
3. Initialization with a number for cryptography ex.: LR.begin(4769);
4. Addresses definition:
    1. Network structure; that is organization of devices addresses space and consequent network addresses space. Ex.: LR.defDevRange(6); (see help for details)
    2. Network address ex.: LR.defNetAddress(733); (see help for details)

If you want change transmission parameters use:

LR.setConfig(SF,BW,CR); where:

SF: Spreading factor code (default 12)
BW: band width code (default 7)
CR: correction code (default 4)

**Sending (example):**

device number 1 sends a message “mess” (string) to device number 12 of network 733 (implicit)

LR.sendNetMess(12,1,”mess”,4); (where 4 is the length of “mess”)

**Receiving (example):**

Before receiving radio module has to set in receiving mode:

LR.receiveMessMode();

Now can be tested messages incoming in a loop:

if (LR.receiveNetMess(1,12, buffer, bufferlen)>0) break;

If received (i.e. break executed) then message can be read:

LR.getMessage(); (returns string)

If you want receive messages from any device, you can uses 0 as from address:

if (LR.receiveNetMess(1,0, buffer, bufferlen)>0) break;

And sender can be returned by:

LR.getNetSender(); (returns number)