# Demo Steps by Steps

- How to connect a LoRa end-node with a LoRa Gateway
- Go to website https://github.com/CongducPham/LowCostLoRaGw
- Click « Clone or download »
- Unzip the downloaded file

- Install Arduino IDE : https://www.arduino.cc/en/Main/Software
- Copy all library in «  \LowCostLoRaGw-master\Arduino\libraries » to your Arduino library core directory
- Open Arduino and Arduino_LoRa_Gateway_1_4.ino project

# Demo Steps by Steps

- In Arduino_LoRa_Gateway_1_4.ino code, uncomment the #define PABOOST line

```
// IMPORTANT when using an Arduino only. For a Raspberry-based gateway the distribution uses a radio.makefile file
//////////////////////////////////////////////////////////////////////////////////////////////////////////////////
// please uncomment only 1 choice
//
// uncomment if your radio is an HopeRF RFM92W, HopeRF RFM95W, Modtronix inAir9B, NiceRF1276
// or you known from the circuit diagram that output use the PABOOST line instead of the RFO line
#define PABOOST
//////////////////////////////////////////////////////////////////////////////////////////////////////////////////
```

- Several parameters can be changed, but the LoRA modes, SF and frequency band must be the same between the GW and the node

- It is time to connect your board

# Demo Steps by Steps

- Connection between the USB-UART programmer to the Arduino mini pro
  - Vcc – 3.3V
  - Rx -> Tx
  - Tx -> Rx
  - Gnd -> Gnd
  - GRN-> DTR-RST
- Select the right port in Arduino IDE
- Compile and upload the code on your board (the UART-USB should blink during the operation)
- Open your serial monitor and use 38400 baud

# Demo Steps by Steps

- You should have this screen :

```
401 bytes of free memory.
SX1272 detected, starting
...
^$**********Power ON: state 0
^$Default sync word: 0x12
^$LoRa mode 1
^$Setting mode: state 0
^$Channel CH_10_868: state 0
^$Set LoRa power dBm to 14
^$Power: state 0
^$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
```

- Now the gateway is ready to receive data

# Demo Steps by Steps

- Now open a second Arduino IDE program (not a new window, really a second application)
- Open the Arduino_LoRa_Ping_Pong.ino sketch
- In the code, uncomment the #define PABOOST line

```
// IMPORTANT when using an Arduino only. For a Raspberry-based gateway the distribution uses a radio.makefile file
//////////////////////////////////////////////////////////////////////////////////////////////////////////////////
// please uncomment only 1 choice
//
// uncomment if your radio is an HopeRF RFM92W, HopeRF RFM95W, Modtronix inAir9B, NiceRF1276
// or you known from the circuit diagram that output use the PABOOST line instead of the RFO line
#define PABOOST
//////////////////////////////////////////////////////////////////////////////////////////////////////////////////
```

- Select the right port
- Check that the gateway and node configuration are the same (frequency band, mode, etc ...)
- Upload your code on the board

# Demo Steps by Steps

End-node side

```
Simple LoRa ping-pong with the gateway
Arduino Pro Mini detected
ATmega328P detected
SX1276 detected, starting
SX1276 LF/HF calibration
...
Setting Mode: state 0
Setting Channel: state 0
Setting Power: state 0
Setting node addr: state 0
SX1272 successfully configured
--> CAD duration 547
OK1
--> waiting for 1 CAD = 62
--> CAD duration 549
OK2
--> RSSI -124
Sending Ping
wait for ACK

Starting 'getACK'
## ACK received:
Destination: 8
Source: 1
ACK number: 0
ACK length: 2
ACK payload: 0
ACK SNR of rcv pkt at gw: -3
##

Packet sent, state 0
Pong received from gateway!
```

Gateway side side

```
401 bytes of free memory.
SX1272 detected, starting
...
^$**********Power ON: state 0
^$Default sync word: 0x12
^$LoRa mode 1
^$Setting mode: state 0
^$Channel CH_10_868: state 0
^$Set LoRa power dBm to 14
^$Power: state 0
^$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
## ACK set and written in FIFO ##
## ACK to send:
Destination: 8
Source: 1
ACK number: 7
ACK length: 2
ACK payload: 0
ACK SNR last rcv pkt: 6
##

^$ACK requested by 8
--- rxlora. dst=1 type=0x18 src=8 seq=7 len=4 SNR=6 RSSIpkt=-42 BW=125 CR=4/5 SF=12
^p1,24,8,7,4,6,-42
^r125,5,12
ÿþPing
```

# Demo Steps by Steps

- With the board very close, your RSSI should be around -45
- If not, try to upload the code again

- Add a battery to the node
- And run !

- Then you can add sensor, actuator …
- Let’s start your project