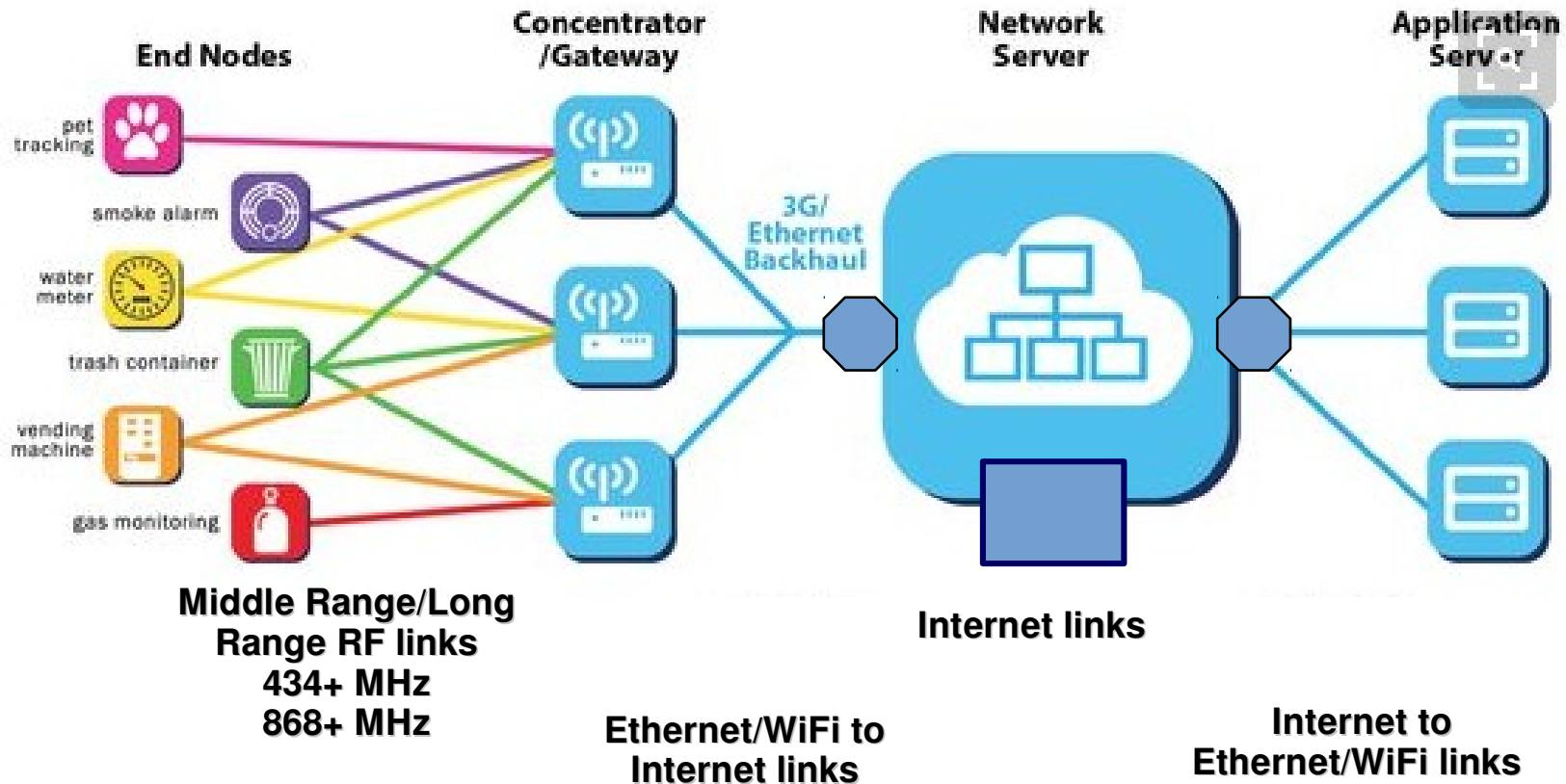






# Internet of Things

Connecting Thing to the Internet





# Internet of Things

## IoT servers – ThingSpeak.com



Report



Internet



Temperature Sensor

**The open IoT platform with MATLAB analytics**



# Internet of Things

IoT servers – ThingSpeak.fr

RIO/SMTR ThingSpeak Channels Support ▾ Blog Sign In Sign Up

## Billions and Billions.

The open data platform for the Internet of Things.

Get Started Now

Documentation Support GitHub

SmartComputerLab

**The open IoT platform for RIO/SMTR**





# Internet of Things

## IoT servers – ThingSpeak.com

ThingSpeak™

Channels ▾




Apps

Community

Support ▾

## My Channels

New Channel

Name	Created
 bakobox <a href="#">Private</a> <a href="#">Public</a> <a href="#">Settings</a> <a href="#">API Keys</a> <a href="#">Data Import / Export</a>	2016-03-05
 opiz <a href="#">Private</a> <a href="#">Public</a> <a href="#">Settings</a> <a href="#">API Keys</a> <a href="#">Data Import / Export</a>	2016-06-21
 wemos <a href="#">Private</a> <a href="#">Public</a> <a href="#">Settings</a> <a href="#">API Keys</a> <a href="#">Data Import / Export</a>	2016-11-03





# Internet of Things

bakobox

Channel ID: 96085

DHT11

Author: bakobox

Access: Public

[Private View](#)

[Public View](#)

[Channel Settings](#)

[API Keys](#)

[Data Import / E](#)

## Write API Key

Key

P18HVHSXTYAARD83

[Generate New Write API Key](#)

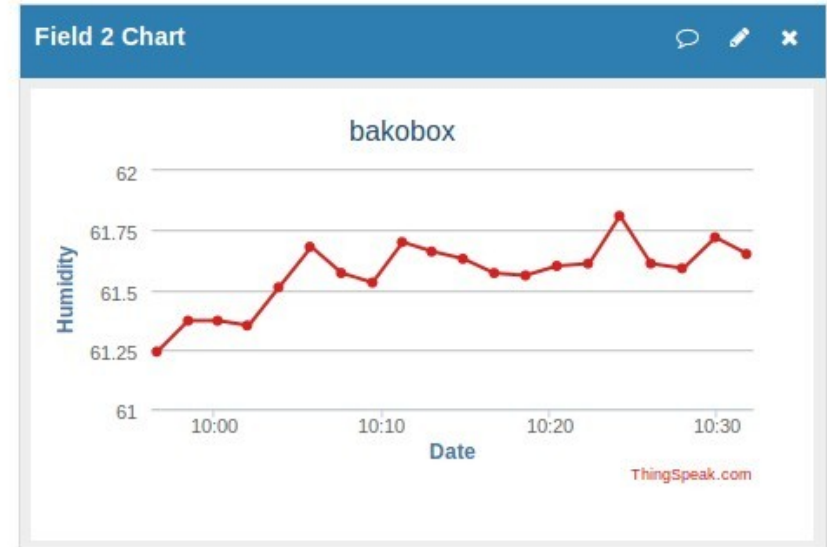
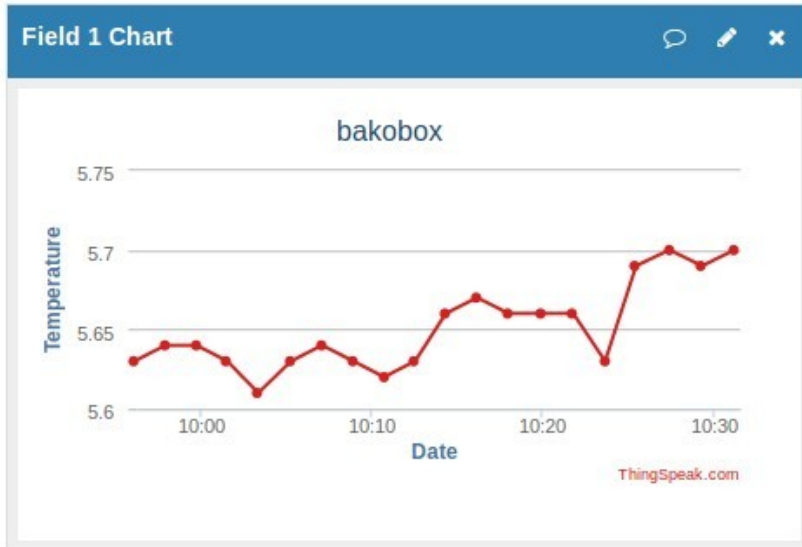
## Read API Keys

Key

9DYDN6XM1G5V87BS



# Internet of Things







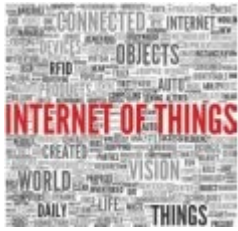


# Internet of Things

## Displaying the data on OLED screen

```
void setup() {  
  Serial.begin(9600);  
  u8x8.begin(); // initialize OLED  
  u8x8.setFont(u8x8_font_chroma48medium8_r);  
}
```

```
void loop()  
{  
  d1++;d2+=2;d3+=4 ;  
  dispData()  
}
```



# Internet of Things

## Sensor data capture

```
#include <Wire.h> // driver I2C
#include "SparkFunHTU21D.h"
HTU21D myTempHumi;

void setup()
{
  Serial.begin(9600);
  Serial.println("HTU21D Example!");
  myTempHumi.begin();
}
```





# Internet of Things

## Sensor data capture

```
void loop()  
{  
  float humd = myTempHumi.readHumidity();  
  float temp = myTempHumi.readTemperature();  
  Serial.print(" Temperature:");  
  Serial.print(temp, 1);  
  Serial.print("C");  
  Serial.print(" Humidity:");  
  Serial.print(humd, 1);  
  Serial.print("%");  
  Serial.println();  
  delay(1000);  
}
```



# Internet of Things

## WiFi modes and functions

### WiFi networks **scanning**

```
#include "WiFi.h"
#include <U8x8lib.h>
U8X8_SSD1306_128X64_NONAME_SW_I2C u8x8(15, 4,
16);

void setup()
{
  WiFi.mode(WIFI_STA);
  WiFi.disconnect();
  delay(100);
  u8x8.begin();
  u8x8.setFont(u8x8_font_chroma48medium8_r);
}
```



# Internet of Things

## WiFi modes and functions

```
void loop()
```

```
{
```

```
  int n = WiFi.scanNetworks();
```

```
  if(n==0) u8x8.drawString(0,0, "Searching !");
```

```
  else {
```

```
    u8x8.drawString(0, 0, "Networks found: ");
```

```
    u8x8.clear();
```

```
    for(int i=0;i<n;++i)
```

```
      {
```

```
        char currentSSID[16];
```

```
        memset(currentSSID,0x00,16);
```

```
        WiFi.SSID(i).toCharArray(currentSSID,16);
```

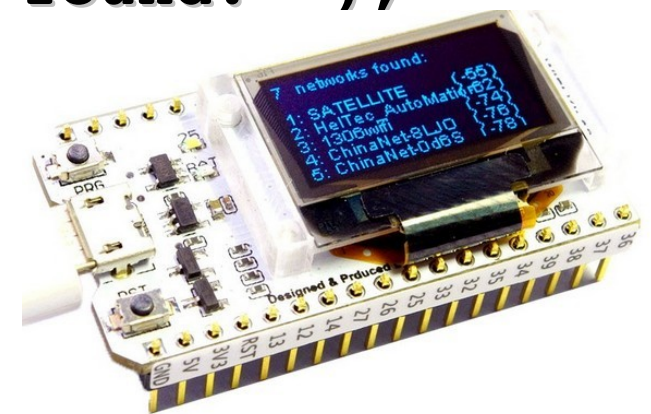
```
        u8x8.drawString(0,i+1,currentSSID);
```

```
      }
```

```
    }
```

```
  }
```

WiFi networks **scanning**





# Internet of Things



## ThingSpeak.h

```
//#define THINGSPEAK_URL "86.217.8.171:3000"  
#define THINGSPEAK_URL "http://90.49.255.63:443"  
  
//#define THINGSPEAK_IPADDRESS IPAddress(82,217,11,191)  
#define THINGSPEAK_IPADDRESS IPAddress(90,49,255,63)  
  
//#define THINGSPEAK_PORT_NUMBER 80  
#define THINGSPEAK_PORT_NUMBER 443
```

Choice between:  
// ThingSpeak.com and  
ThingSpeak.fr



# Internet of Things

## WiFi - **STA mode**

```
#include <WiFi.h>
#include "ThingSpeak.h"

char ssid[] = "YotaPhoneAP"; // your network SSID
char pass[] = "..."; // your network passw

unsigned long myChannelNumber = 1;
const char * myWriteAPIKey="MEH7A0FHAMNWJE8P" ;

WiFiClient client;
```



# Internet of Things

## Connecting to WiFi AP

```
void setup() {  
  Serial.begin(9600);  
  WiFi.disconnect(true); // remove l'EEPROM WiFi credentials  
  delay(1000);  
  WiFi.begin(ssid, pass);  
  delay(1000);  
  while (WiFi.status() != WL_CONNECTED) {  
    delay(500);  
    Serial.print(".");  
  }  
  IPAddress ip = WiFi.localIP();  
  Serial.print("IP Address: ");  
  Serial.println(ip);  
  Serial.println("WiFi setup ok");  
  delay(1000);  
  ThingSpeak.begin(client); // TCP connection to server  
  delay(1000);  
}
```



# Internet of Things

## Sending data to ThingSpeak.fr

```
int tout=10000; // in milliseconds
float luminosity=100.0, temperature=10.0;

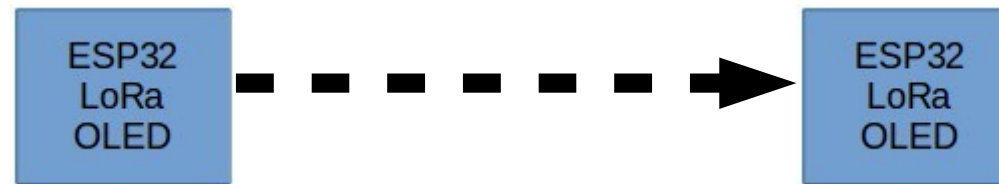
void loop()
{
while (WiFi.status() != WL_CONNECTED) {
    delay(500);Serial.print(".");
}
ThingSpeak.setField(1,luminosity); // preparing field1
ThingSpeak.setField(2,temperature); // preparing field2
ThingSpeak.writeFields(myChannelNumber, myWriteAPIKey);
luminosity++;
temperature++;
delay(tout);
}
```





# Internet of Things

## LoRa a Long Range radio link



## LoRa modem connection – SPI with sx127x

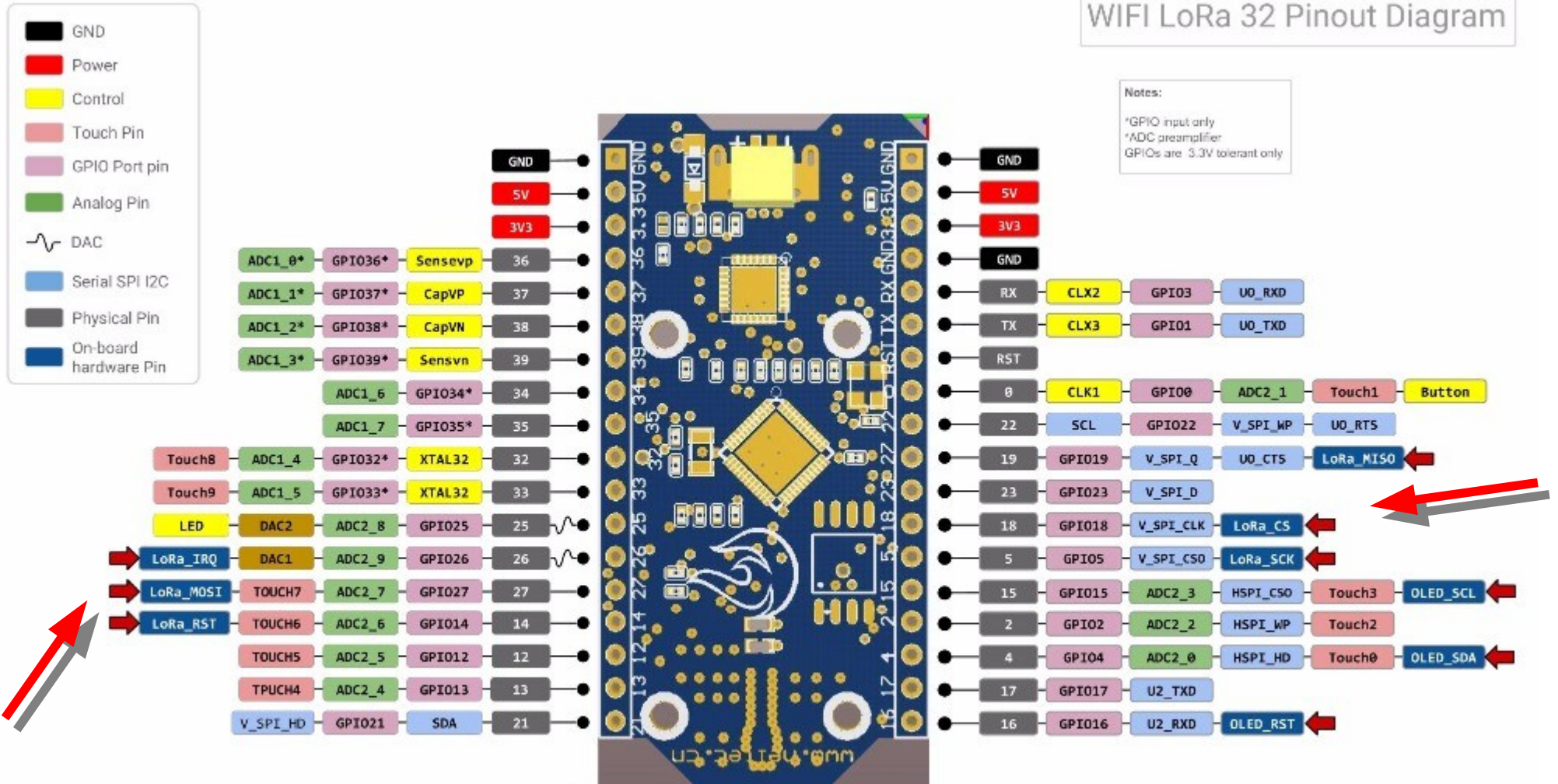
```
#include <SPI.h> // include libraries
#include <LoRa.h>
#define SCK      5 // GPIO5 -- SX127x's SCK
#define MISO     19 // GPIO19 -- SX127x's MISO
#define MOSI     27 // GPIO27 -- SX127x's MOSI
#define SS       18 // GPIO18 -- SX127x's CS
#define RST      14 // GPIO14 -- SX127x's RESET
#define DI0      26 // GPIO26 -- SX127x's IRQ
```



# Internet of Things

## LoRa – SPI connection lines

WiFi LoRa 32 Pinout Diagram



→ Pins with this arrow are used by on-board OLED or LoRa, they must not be used for other purpose unless you know what you are doing!

# Internet of Things

## LoRa parameters and packets



```
#define freq      868E6
#define sf 8
#define sb 125E3
```

parameters

**union** pack

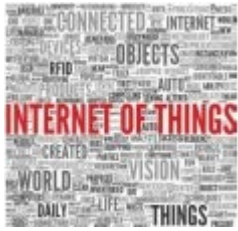
```
{
    uint8_t frame[16]; // frame with bytes
    float data[4]; // 4 floating point values
} sdp ; // packet to send
```

simple packet



# Internet of Things

## Complete LoRaTS packet format



```
union tspack
{
    uint8_t frame[128];
    struct packet
    {
        uint8_t head[4];
        uint16_t num;
        uint16_t tout;
        union {
            float sensor[8];
            char text[32];
        } tsdata;
    } pack;
} sdf,sbf,rdf,rbf; // data or beacon frame
```

# Internet of Things



## setup function

```
void setup() {  
  Serial.begin(9600);  
  pinMode(DIO, INPUT); // signal interrupt  
  SPI.begin(SCK, MISO, MOSI, SS);  
  LoRa.setPins(SS, RST, DIO);  
  if (!LoRa.begin(freq)) {  
    Serial.println("Starting LoRa failed!");  
    while (1);  
  }  
  LoRa.setSpreadingFactor(sf);  
  LoRa.setSignalBandwidth(sb);  
}
```

# Internet of Things

sending function – sender board



```
float d1=12.0, d2=321.54 ;
```

```
void loop() // la boucle de l'emetteur
{
  Serial.println("New Packet") ;
  LoRa.beginPacket(); // start packet
  sdp.data[0]=d1;
  sdp.data[1]=d2;
  LoRa.write(sdp.frame,16);
  LoRa.endPacket();
  d1++; d2+=2;
  delay(2000);
}
```

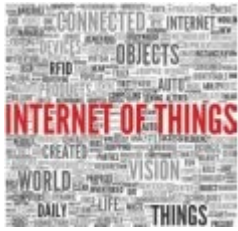
# Internet of Things



receiving function – receiver board

```
float d1,d2; int rssi,i=0;

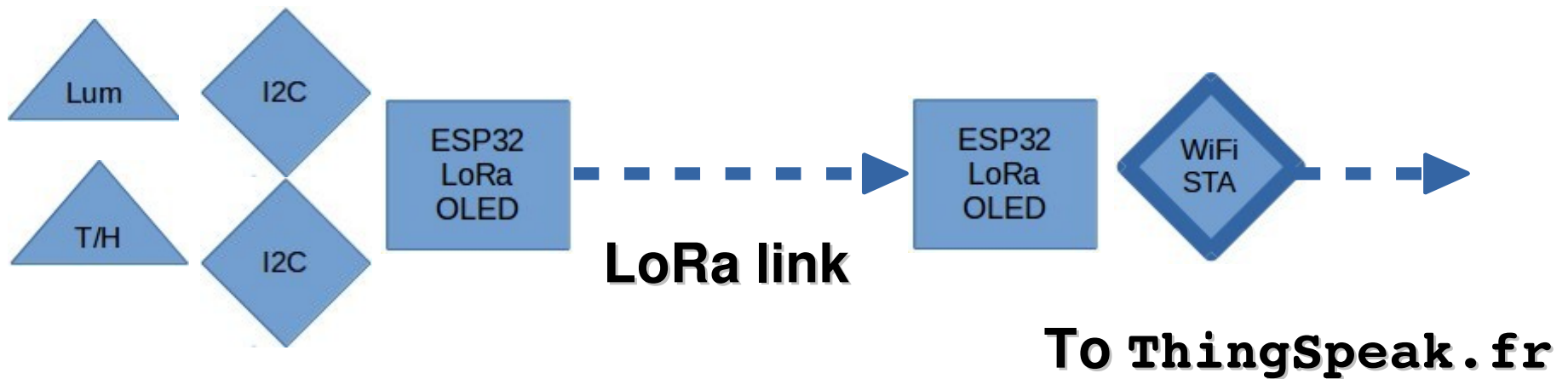
void loop()
{
int packetLen;
packetLen=LoRa.parsePacket();
if(packetLen==16)
{ i=0;
while (LoRa.available()) {
rdp.frame[i]=LoRa.read();i++;
}
d1=rdp.data[0];
d2=rdp.data[1];
rssi=LoRa.packetRssi();
}
}
```



# Internet of Things

## What next ?

In next step we are going to build a complete IoT architecture like this one :





# Internet of Things

And what next, next ?

**In the following labs we are going to study and experiment :**

**With freeRTOS applied to IoT**

**And develop several IoT architectures to be selected according to the application domain such as :**

- **Human/Interface communication**
- **Security,**
- **Environment,**
- **Detection,**
- **Commerce**
- **And Very Long Range,..**