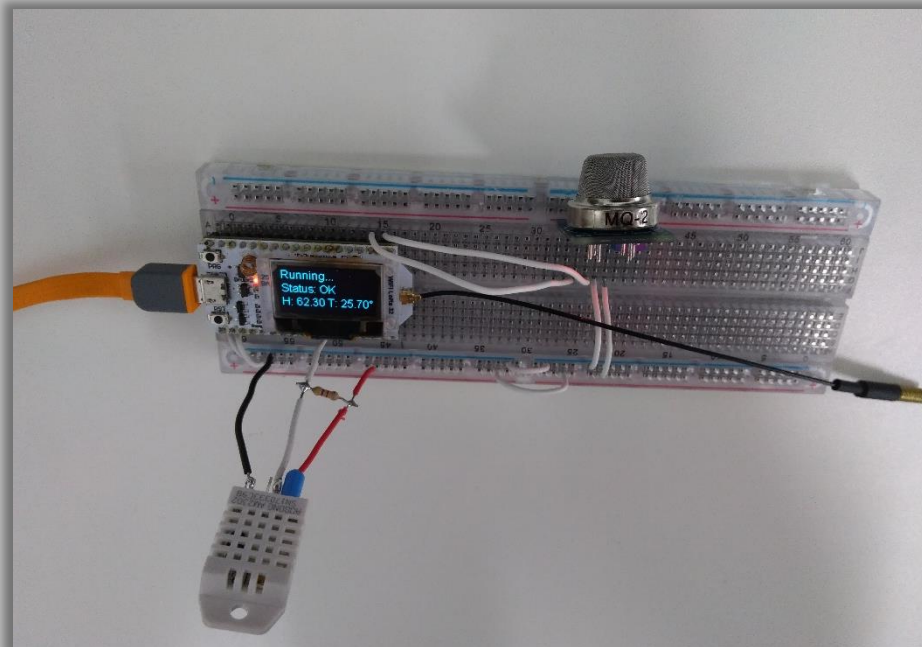
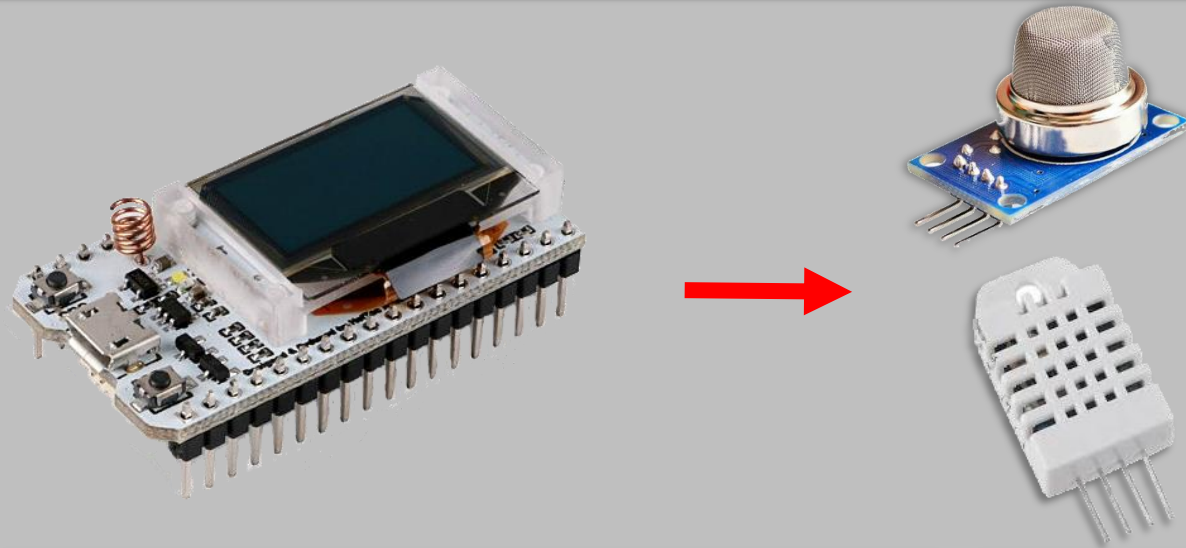
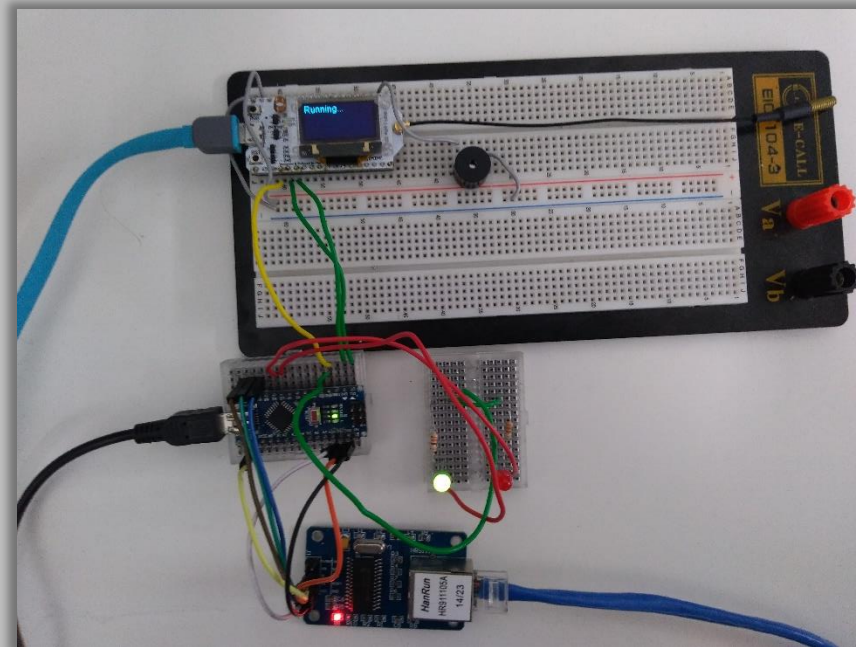


# ESP32 Lora: Dispositivos de Segurança



Sender



Receiver



Por Fernando Koyanagi

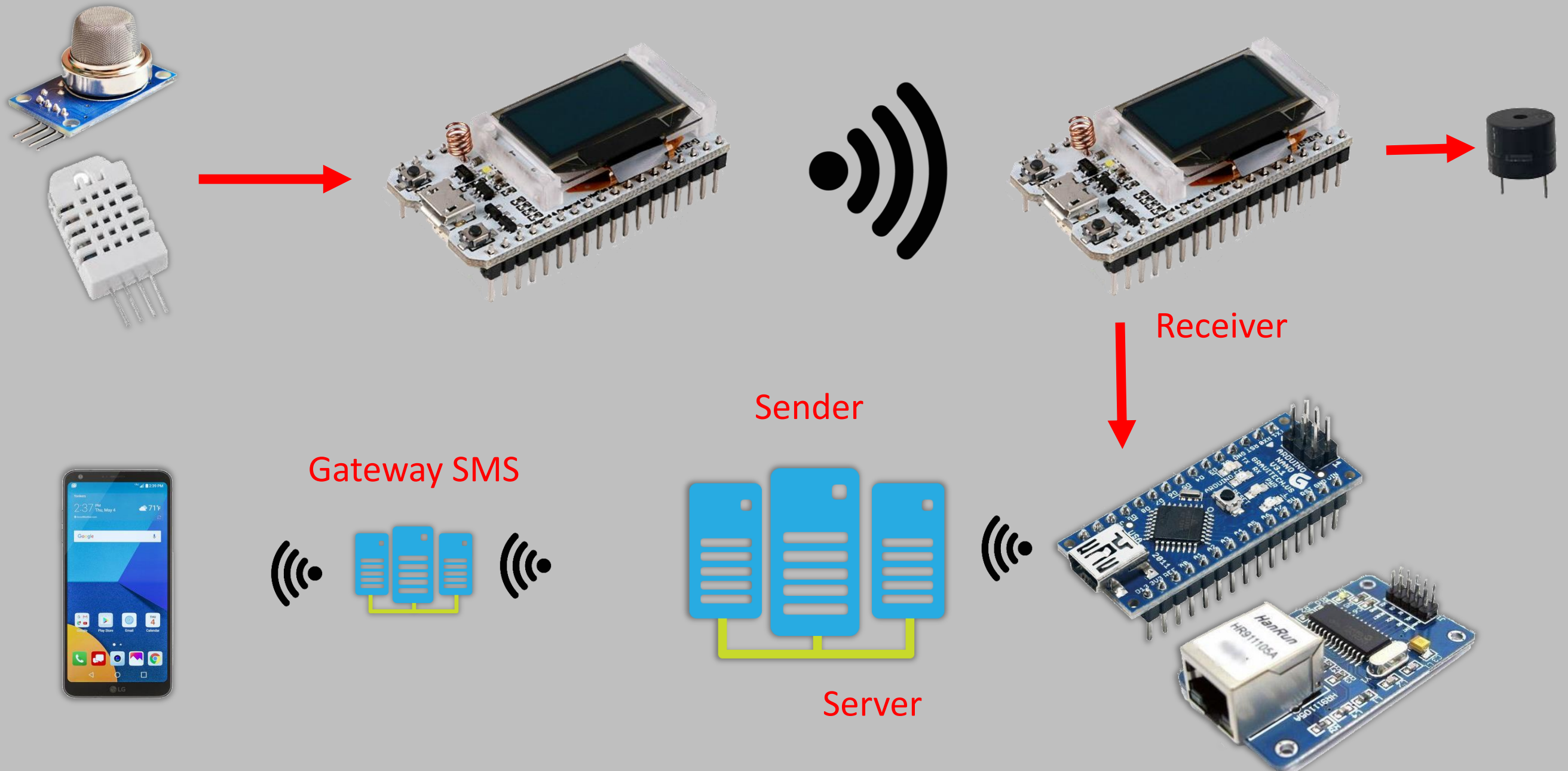
# Intenção dessa aula (Continuação...)

1. Revisar o funcionamento do protótipo com ESP32 Lora
2. Entender o código .INO do Receiver e do Arduino Nano.





# Funcionamento (Transmissão de dados)





Em [www.fernandok.com](http://www.fernandok.com)

Seu e-mail



PRINCIPAL SOBRE FERNANDO K ARDUINO ESP8266 ESP32 LORAWAN MOTOR DISPLAY MATERIAIS DOWNLOAD

Receba o meu conteúdo  
GRATUITAMENTE

Insira aqui seu melhor email...

QUERO RECEBER GRÁTIS



## Motor de Passo Nema 23 com Driver TB6600 e Arduino Due

by Fernando K Tecnologia - 2:44 PM

Hoje vamos voltar a falar de Motor de Passo. Vamos utilizar um Nema 23 que será controlado por um Driver TB6600 e um Arduino Due. É p...

Leia mais



## ESP32 Longa Distância - LoRaWan

by Fernando K Tecnologia - 9:46 AM

Neste artigo vamos tratar da LoRaWAN, uma rede que vai longe gastando pouca energia. Mas, o quanto "longe"? Com o chip que uso no vídeo...

Leia mais



## Motor de HD com Arduino

by Fernando K Tecnologia - 2:00 PM

### QUAL ASSUNTO VOCÊ TEM

- ☐ Arduino
- ☐ ESP8266
- ☐ ESP32
- ☐ Motor
- ☐ Display
- ☐ Sensor

You may select multiple answers.

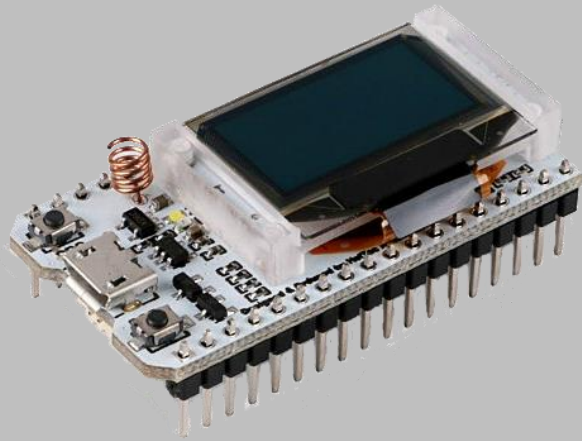
Votar Exibir resultados

Votos até o momento: 32

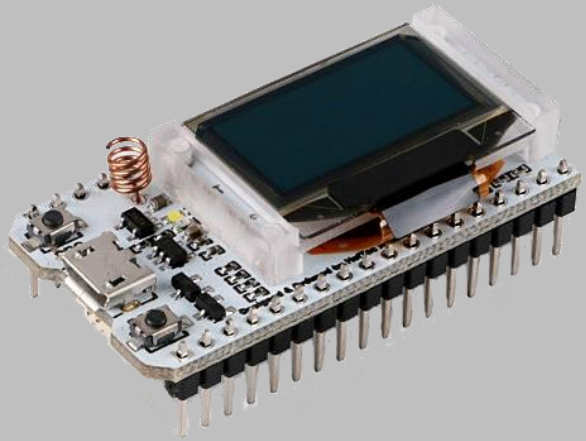
Dias restantes para votar: 49

### FACEBOOK





# Código Receiver



# Receiver – Organização do código

Setup

Função responsável por ler os bytes do pacote recebido e atribuir para uma String

Loop

readPacket

showDisplay

Função responsável por **exibir uma mensagem no display** com as fontes e alinhamentos já configurados.

alarm

Função responsável por **ativar o buzzer** durante 1 segundo

## Código Receiver [Includes e Defines]

```
#include <SPI.h> //responsável pela comunicação serial
#include <LoRa.h> //responsável pela comunicação com o WIFI Lora
#include <Wire.h> //responsável pela comunicação i2c
#include "SSD1306.h" //responsável pela comunicação com o display

// Definição dos pinos
#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 DI00 26 // GPIO26 -- SX127x's IRQ(Interrupt Request)

#define BAND 433E6 //Frequencia do radio - podemos utilizar ainda :
433E6, 868E6, 915E6

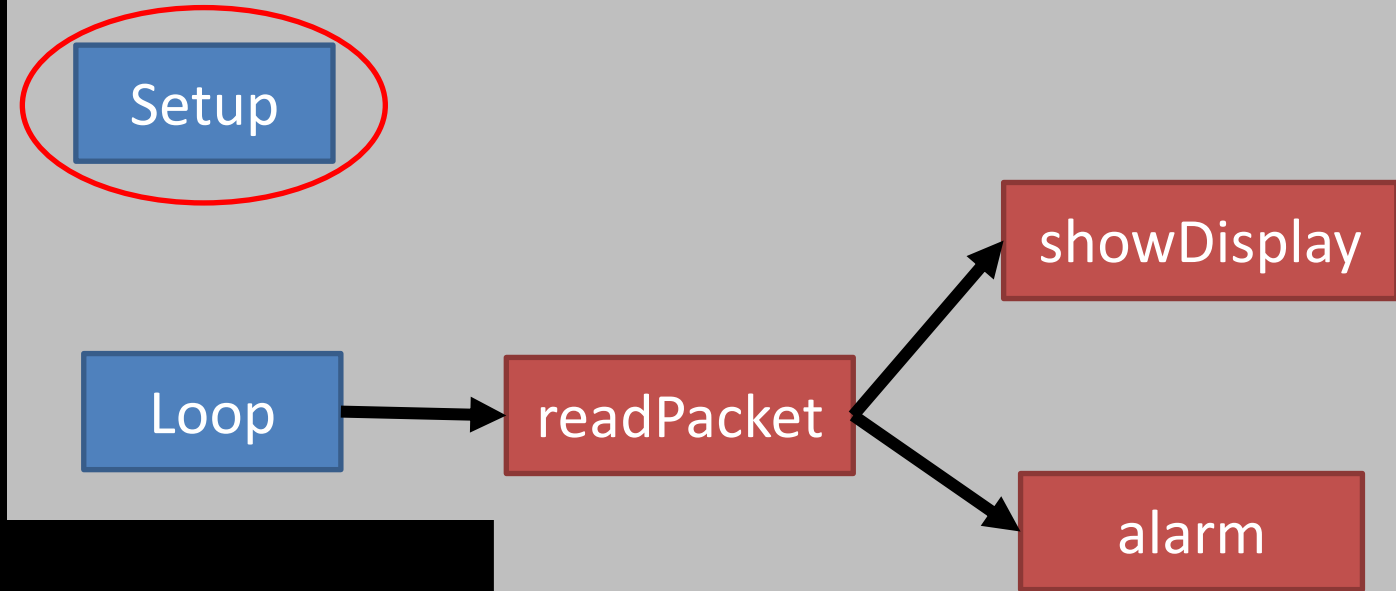
//parametros: address,SDA,SCL
SSD1306 display(0x3c, 4, 15); //construtor do objeto que controlaremos o
display

String packSize; //tamanho do pacote convertido em String
String packet; //pacote recebido
const int buzzerpin = 13; //buzzer
```



## Código Receiver [Setup]

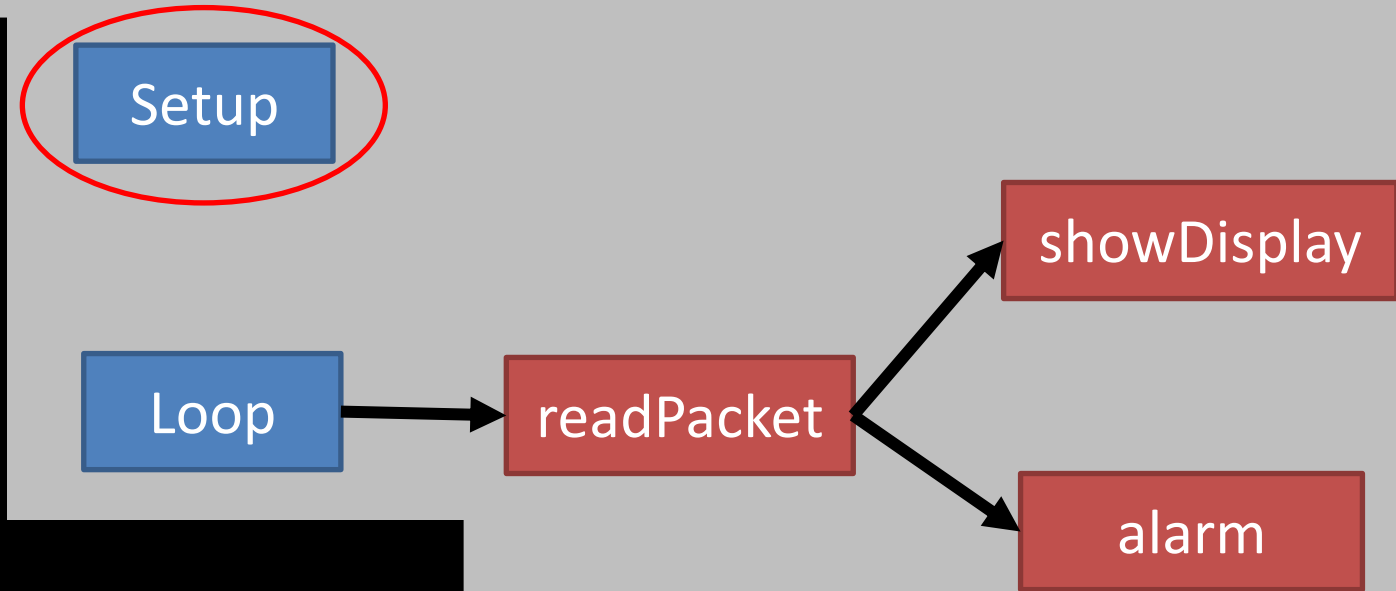
```
void setup()  
{  
  //inicia serial com 9600b p/s  
  Serial.begin(9600);  
  //configura buzzer como saída  
  pinMode(buzzerpin, OUTPUT);  
  //configura RST do oled como saída  
  pinMode(16, OUTPUT);  
  //reseta o OLED  
  digitalWrite(16, LOW);  
  //aguarda 50ms  
  delay(50);  
  //enquanto o OLED estiver ligado, GPIO16 deve estar HIGH  
  digitalWrite(16, HIGH);  
  //inicializa display  
  display.init();  
  //inverte a tela verticalmente (de ponta cabeça)  
  display.flipScreenVertically();  
  //configura a fonte do display  
  display.setFont(ArialMT_Plain_10);  
  //aguarda 1500ms  
  delay(1500);  
}
```





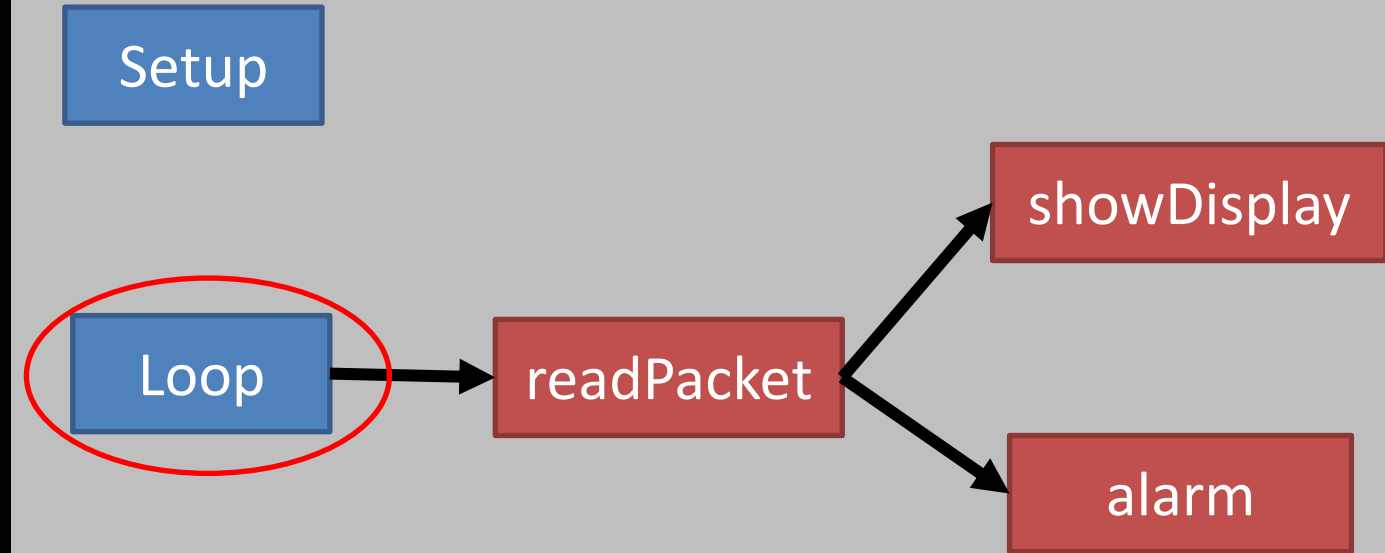
# Código Receiver [Setup]

```
//limpa display
display.clear();
//inicia a com serial com o Lora
SPI.begin(SCK,MISO,MOSI,SS);
//conf. os pinos que serão utilizados
pela biblioteca
(deve ser chamado antes do LoRa.begin)
LoRa.setPins(SS,RST,DI00);
//inicializa o Lora com a frequência
específica.
if (!LoRa.begin(BAND))
{
  //escreve na pos 0,0 a mensagem em aspas
  display.drawString(0, 0, "Starting LoRa failed!");
  //exibe no display
  display.display();
  //deixa em loop infinito até que o ESP seja reiniciado
  while (1);
}
//habilita o Lora para receber dados
LoRa.receive();
}
```



## Código Receiver [Loop]

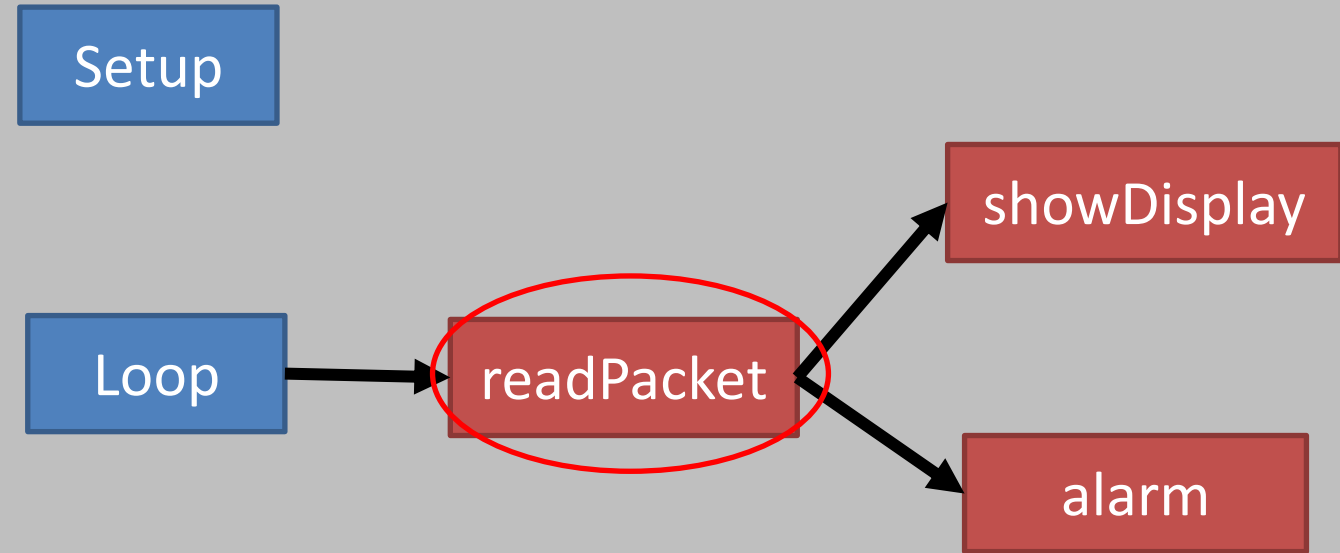
```
void loop()
{
  //limpa display
  display.clear();
  //configura alinhamento de texto à esquerda
  display.setTextAlignment(TEXT_ALIGN_LEFT);
  //configura fonte do texto
  display.setFont(ArialMT_Plain_16);
  //escreve na pos 0,10 a mensagem entre aspas
  display.drawString(0, 0, "Running...");
  //exibe no display
  display.display();
  //parsePacket: checa se um pacote foi recebido
  //retorno: tamanho do pacote em bytes. Se retornar 0 (ZERO) nenhum pacote foi recebido
  int packetSize = LoRa.parsePacket();
  //caso tenha recebido pacote
  if (packetSize)
  {
    //chama a função responsável por recuperar o conteúdo do pacote recebido
    readPacket(packetSize);
  }
  delay(10); //aguarda 10ms
}
```



## Código Receiver [readPacket]

```
//função responsável por recuperar o conteúdo
do pacote recebido
//parametro: tamanho do pacote (bytes)
void readPacket(int packetSize)
{
  //deixa string com valor vazio
  packet = "";

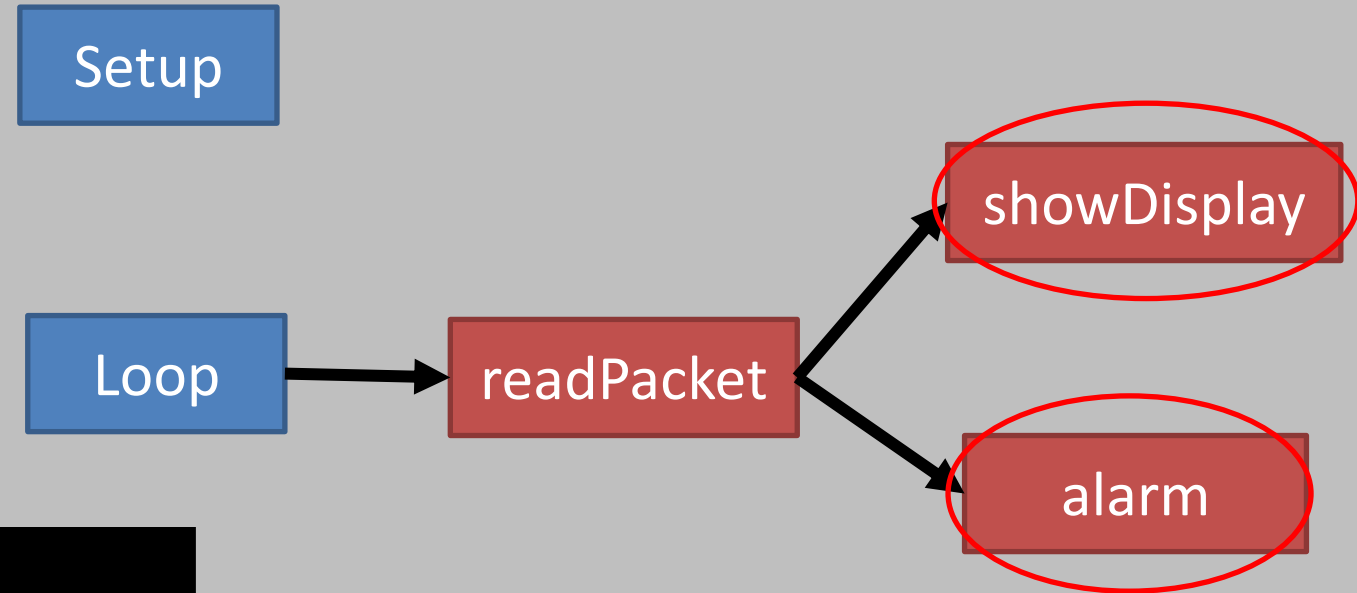
  //lê byte a byte do pacote e atribui à string
  for (int i = 0; i < packetSize; i++)
  {
    //recupera o dado recebido e concatena na variável "packet"
    packet += (char) LoRa.read();
  }
  //se a mensagem recebida contém "alarm", então soa o buzzer
  if(strstr(packet.c_str(),"ALARM")!=NULL)
  {
    Serial.println(packet); //exibe pacote
    showDisplay("Gas Detected!"); //exibe no display
    alarm(); //soa o buzzer
  }
}
```



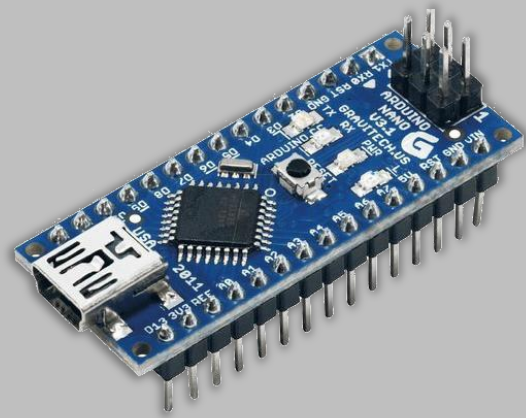
## Código Receiver [alarm e showDisplay]

```
void alarm()
{
  //ativa buzzer
  digitalWrite(buzzerpin, HIGH);
  //aguarda 1 segundo
  delay(1000);
  //desativa buzzer
  digitalWrite(buzzerpin, LOW);
}

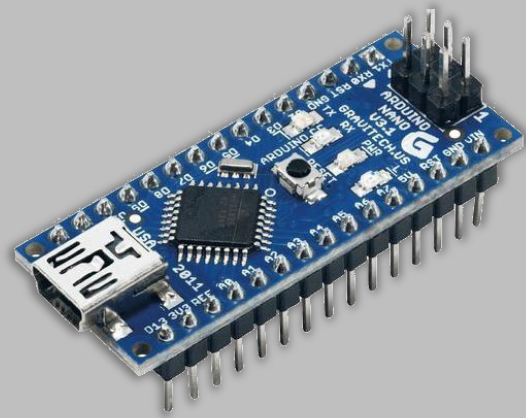
//exibe mensagem recebida como parâmetro
void showDisplay(String msg)
{
  display.clear(); //limpa o display
  //configura alinhamento de texto à esquerda
  display.setTextAlignment(TEXT_ALIGN_LEFT);
  //configura fonte do texto
  display.setFont(ArialMT_Plain_16);
  //escreve na pos 0,25 a mensagem
  display.drawString(0 , 25 , msg);
  //exibe no display
  display.display();
}
```







# Código Arduino Nano



# Arduino – Organização do código

Setup



startEthernet

Função responsável por iniciar a conexão do **client ethernet**

Loop



splitString

Função responsável por “retirar” os **valores de umidade e temperatura** da string recebida pela serial.

## Código Arduino [Includes e Defines]

```
#include <SPI.h> //lib para a Serial Peripheral Interface (SPI)
#include <UIPEthernet.h> //lib Ethernet para o módulo ENC28J60

#define GREENLED 7 //pino led verde
#define REDLED 8 //pino led vermelho

//endereço mac (1,2,3,4,5) para a configuração ethernet
uint8_t mac[6] = {0x00,0x01,0x02,0x03,0x04,0x05};
//gateway de envio de SMS
char server[] = "211.180.xxx.xxx";
//ip da rede destinado ao módulo ethernet (conectado por cabo)
IPAddress ip(198,168,xxx,xxx);
//objeto Ethernet para conexões de clientes
EthernetClient client;
//Umidade (Humidity) e Temperatura
String H,T;

//Flag que permite apenas um envio de mensagem
//O envio fica dentro do "loop()" , assim que o arduino recebe dados na porta serial
ele envia a mensagem
//Caso esta flag não exista, um número descontrolado de mensagens será enviado
bool messageSent = false;
```

# Código Arduino [Setup]

```
void setup()
{
  //inicializa Serial com 9600 bits por segundo
  Serial.begin(9600);

  //enquanto a serial não for inicializada, aguarda
  while (!Serial);

  //exibe mensagem
  Serial.println("Serial ok");
  //define pino do led verde como saída
  pinMode(GREENLED,OUTPUT);
  //define pino do led vermelho como saída
  pinMode(REDLED,OUTPUT);

  //inicializa e configura o client ethernet
  if(startEthernet())
  {
    //se a ethernet foi iniciada com sucesso
    //acende o led verde
    digitalWrite(GREENLED,HIGH);
  }
}
```

Setup

startEthernet

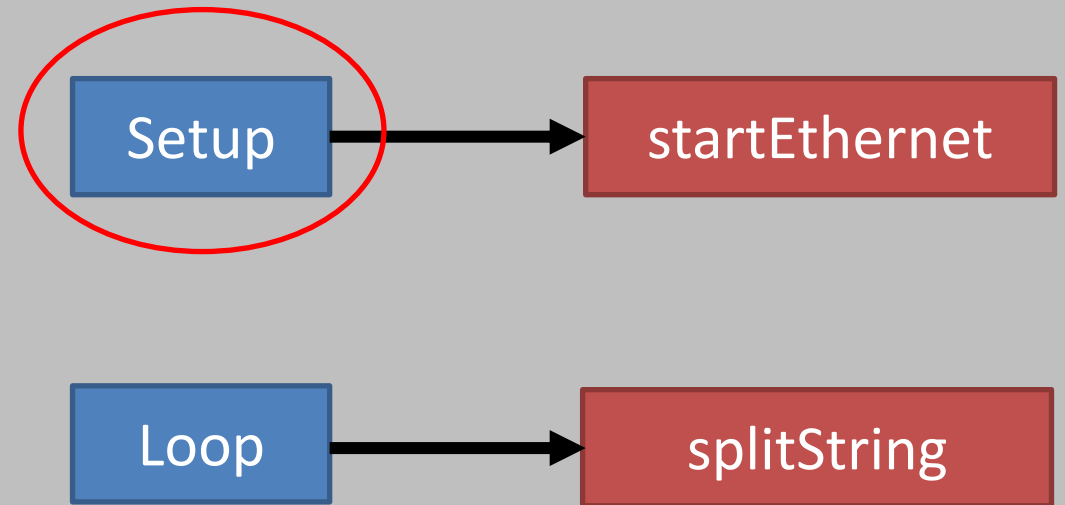
Loop

splitString



## Código Arduino [Setup]

```
else
{
  //se a ethernet não for inicializada
  //pisca led vermelho
  digitalWrite(GREENLED,HIGH);
  delay(500);
  digitalWrite(GREENLED,LOW);
  delay(500);
  digitalWrite(GREENLED,HIGH);
  delay(500);
  digitalWrite(GREENLED,LOW);
}
}
```

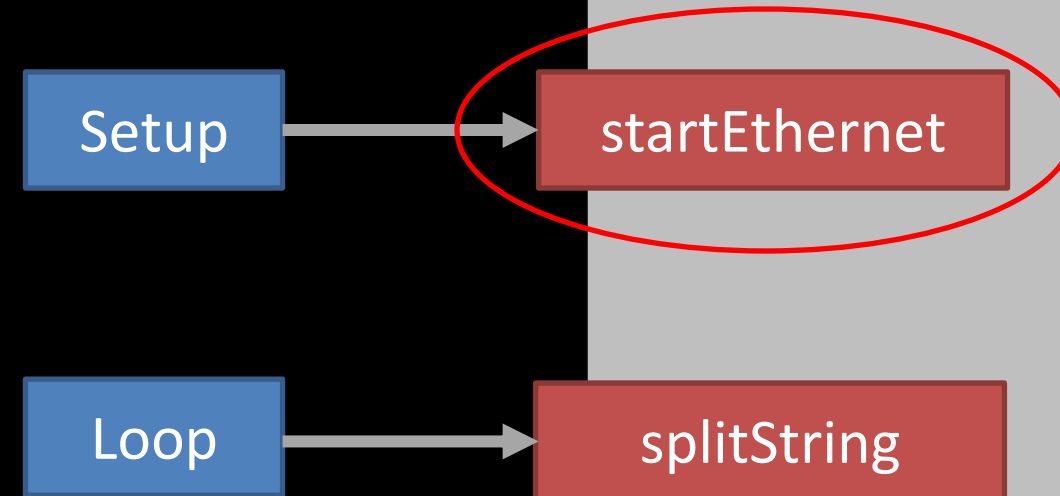


# Código Arduino[startEthernet]

//Obs: As exibições na serial (Serial.println) são apenas para verificações, ao serem retiradas do código não mudarão o funcionamento do protótipo

//função responsável por inicializar e conectar o client ethernet

```
bool startEthernet()
{
  //configura dados de rede
  Ethernet.begin(mac, ip); // mac/ip da rede interna
  //dá ao Ethernet Shield um segundo para inicializar
  delay(1000);
  //exibe mensagem
  Serial.println("connecting...");
  //se o cliente conectou
  if (client.connect(server, 80))
  {
    //exibe mensagem
    Serial.println("connected");
    //retorna verdadeiro
    return true;
  }
  //se não...
  //exibe mensagem
  Serial.println("connection failed");
  //retorna falso
  return false;
}
```



# Código Arduino [Loop]

```
void loop()
{
  //se existem dados enviados pela serial
  if(Serial.available())
  {
    //se o cliente não estiver conectado
    if(!client.connected())
    {
      //exibe mensagens
      Serial.println("Erro de envio");
      Serial.println("Desconectando.");
      //desconecta cliente
      client.stop();
      //encerra comunicação serial
      Serial.end();
      //evita que o código abaixo seja executado
      return;
    }
    //se o cliente estiver conectado...
```

Setup

startEthernet

Loop

splitString

# Código Arduino [Loop]

```
//lê mensagem enviada pela serial
String msg = Serial.readString();
//se na mensagem existir a palavra "ALARM" e o SMS ainda
não estiver sido enviado
if(strstr(msg.c_str(),"ALARM")!=NULL && !messageSent)
{
  //"desmembra" mensagem e atribui para as variáveis os
valores de Umidade e Temperatura
  splitString(msg);
  Serial.println("Packet: "+msg);
  //envia sms
  //aqui vai o seu link para envio do sms
  client.println("GET /script/sms/meu_script_sms.php?umidade="+H+"&temperatura="+T);
  //fecha conexão client
  client.stop();
  //exibe mensagem enviada
  Serial.println("Message sent!");
  //sinaliza que enviou acendendo o led vermelho
  digitalWrite(REDLED,HIGH);
  //atribui true para a flag, evitando múltiplos envios
  messageSent = true;
}
```

Setup

startEthernet

Loop

splitString



# Código Arduino [Loop]

```
//continuação da função loop...

//se o client está desconectado
if (!client.connected())
{
  //exibe mensagem
  Serial.println("disconnecting.");
  //desconecta client
  client.stop();
  //sinaliza a desconexão apagando o led verde
  digitalWrite(GREENLED, LOW);
  //não faz nada até que o Arduino seja reiniciado
  while (true);
}
}
```

Setup

startEthernet

Loop

splitString

# Código Arduino[splitString]

```
//Exemplo de mensagem: ALARM|30.12|27.82°C  
//"desmembra" mensagem atribuindo os valores às variáveis
```

```
void splitString(String msg)
```

```
{
```

```
  H = T = ""; //inicia variáveis com vazio
```

```
  int i;
```

```
  //percorre string até encontrar o delimitador pipe "|"
```

```
  for(i=0; i< msg.length() && msg.charAt(i)!='|'; i++);
```

```
  i++;
```

```
  //percorre string até o próximo delimitador atribuindo valor de umidade
```

```
  while(i< msg.length() && msg.charAt(i)!='|')
```

```
  {
```

```
    H+=msg.charAt(i);
```

```
    i++;
```

```
  }
```

```
  i++;
```

```
  //percorre string até o símbolo "°" atribuindo valor de temperatura
```

```
  while(i< msg.length() && msg.charAt(i)!='°')
```

```
  {
```

```
    T+=msg.charAt(i);
```

```
    i++;
```

```
  }
```

```
}
```

Setup

startEthernet

Loop

splitString

# Script de envio de Mensagens SMS

```
<?php
$umidade = $_GET['umidade'];
$temperatura = $_GET['temperatura'];
header('Content-type: text/html; charset=UTF-8');
function post_to_url($url, $data)
{
    $fields = http_build_query($data);
    $post = curl_init();
    $url = $url.'?'.$fields;
    curl_setopt($post, CURLOPT_URL, $url);
    curl_setopt($post, CURLOPT_POST, 1);
    curl_setopt($post, CURLOPT_POSTFIELDS, $fields);
    $result = curl_exec($post);
    if($result == false)
    {
        die('Curl error: ' . curl_error($post));
    }
    curl_close($post);
}
$url = "https://GateWay-SMS";
$data = array
(
    'content' => 'Gás detectado!!!Umidade do ar:'.$umidade.' -Temperatura:'.$temperatura.' C',
    'sender' => 'ID', 'receivers' => '18900000000'
);

    post_to_url($url, $data);
?>
```

\*É preciso um gateway de SMS  
para que o script funcione!

https://sms.comtele.com.br/

←

→

↻

🏠

Seguro | https://sms.comtele.com.br

🔍

Apps

🔥

Fórmula de Lançamei

🔍

Google

📍

Google Maps

🔗

DownSub.com | Dow

🍏

(3) Seus alunos | Khar

📺

YouTube

📄

New Tab

»

📁

Outros favorit

SMS

Início

# Gateway SMS

Gerenciamento e envio de SMS.

Solicite Seu Teste Grátis

COMTELE  
A MESSAGING COMPANY

Usuário ou E-mail

Digite seu usuário.

Senha

Digite sua senha.

➡ Conectar



Em [www.fernandok.com](http://www.fernandok.com)

Download arquivos PDF, **INO e Headers** do código fonte

