



# Projeto 07

## Coisas Remotas – Teoria

Jan K. S. – [janks@puc-rio.br](mailto:janks@puc-rio.br)

ENG4051 – Projeto Internet das Coisas

# Internet

6  
0  
2  
1  
3  
4  
5  
7  
8  
9

Bases Numéricas

# Arquivo Texto

# Gamified Education

A website to help gamified courses, built with Django (Python).

```


```

## ## Motivation

Life in school is very boring. Teachers usually keep talking on and on while students passively watch them. The main goal is usually getting good grades in tests, instead of practicing the concepts.

But it doesn't have to be like that. There are modern methodologies like [Active Learning]([https://en.wikipedia.org/wiki/Active\\_learning](https://en.wikipedia.org/wiki/Active_learning)), [Flipped Classroom]([https://en.wikipedia.org/wiki/Flipped\\_classroom](https://en.wikipedia.org/wiki/Flipped_classroom)) and [Gamification in Learning]([https://en.wikipedia.org/wiki/Gamification\\_in\\_education](https://en.wikipedia.org/wiki/Gamification_in_education)). These concepts can make a huge motivation and performance. I recommend watching [this video](<https://www.youtube.com/watch?v=MuDLw1zIc94>) and [Salman Khan]([https://www.ted.com/talks/sal\\_khan\\_lets\\_use\\_video\\_to\\_reinvent\\_education](https://www.ted.com/talks/sal_khan_lets_use_video_to_reinvent_education)).

This app targets the Gamification part of the equation.

Não é fácil de entender  
dados binários...



É fácil de entender dados  
na forma de texto.



# Arquivo Binário

The image shows a Mac OS X file browser window titled "imagem.png". The file is identified as a "Image" type. The main pane displays a large amount of corrupted data, appearing as a dense, illegible sequence of characters and symbols, likely representing the raw bytes of the image file.

# Problema para Exibir Dados Binários na Forma de Texto

Base 10 (Decimal)	Base 2 (Binário)	Base 16 (Hexadec.)
0	0	0
1	1	1
2	10	2
3	11	3
4	100	4
5	101	5
6	110	6
7	111	7
8	1000	8
9	1001	9
10	1010	A
11	1011	B
12	1100	C
13	1101	D
14	1110	E
15	1111	F
16	10000	10
17	10001	11
18	10010	12
19	10011	13

Base 10 (Decimal)	Base 2 (Binário)	Base 16 (Hexadec.)
20	10100	14
21	10101	15
22	10110	16
23	10111	17
24	11000	18
25	11001	19
26	11010	1A
27	11011	1B
28	11100	1C
29	11101	1D
30	11110	1E
31	11111	1F
32	100000	20
33	100001	21
34	100010	22
35	100011	23
36	100100	24
37	100101	25
...	...	...
255	11111111	FF

Exemplo de Bases Numéricas

Base 10 (Decimal)	Base 64	Base 10 (Decimal)	Base 2 (Binário)	Base 10 (Decimal)	Base 2 (Binário)	Base 10 (Decimal)	Base 2 (Binário)
0	A	20	U	40	o	60	8
1	B	21	W	41	p	61	9
2	C	22	V	42	q	62	+
3	D	23	X	43	r	63	/
4	E	24	Y	44	s	64	BA
5	F	25	Z	45	t	65	BB
6	G	26	a	46	u	66	BC
7	H	27	b	47	w	67	BD
8	I	28	c	48	v	68	BE
9	J	29	d	49	x	69	BF
10	K	30	e	50	y	70	BG
11	L	31	f	51	z	71	BH
12	M	32	g	52	0	72	BI
13	N	33	h	53	1	73	BJ
14	O	34	i	54	2	74	BK
15	P	35	j	55	3	75	BL
16	Q	36	k	56	4	76	BM
17	R	37	l	57	5	77	BN
18	S	38	m	58	6	78	BO
19	T	39	n	59	7	79	BP

10110100 00010100 01110000 10101000 00101111 01010011

Base 10      180      020      112      168      047      083

18 caracteres

10110100 00010100 01110000 10101000 00101111 01010011

Base 16      B4      14      70      A8      2F      53

12 caracteres

101101 000001 010001 110000 101010 000010 111101 010011

Base 64      t      B      R      w      q      C      9      T

8 caracteres

10110100 00010100 01110000 10101000 00101111 01010011

101101 000001 010001 110000 101010 000010 111101 010011

t B R w q C 9 T

10110100 00010100 01110000 10101000 00101111

101101 000001 010001 110000 101010 000010 111100

t B R w q C 8=

10110100 00010100 01110000 10101000

101101 000001 010001 110000 101010 000000

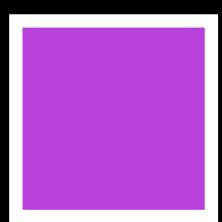
t B R w q A==

10110100 00010100 01110000

101101 000001 010001 110000

t B R w

# Base Hexadecimal



Cor  
#AB4AD6



IPv6

9d05:e2ce:d16f:6321:523e:538b:d443:65e1



RFID

A0 F3 77 8E

Endereços de Memória

0x7FFF1200

0x7FFF1201

0x7FFF1202

0x7FFF1203

## Base 64

### Certificado HTTPS

-----BEGIN CERTIFICATE-----

MIIIfzCCA10gAwIBAgIRAIQz7DSQ0NZRGPGu20CiwAwDQYJKoZIhvcNAQELBQAw  
TzELMAkGA1UEBhMCVVMxKTAnBgNVBAotIEludGVybmV0IFNlY3VyaXR5IFJlc2Vh

...

Imagen no HTML

```

```

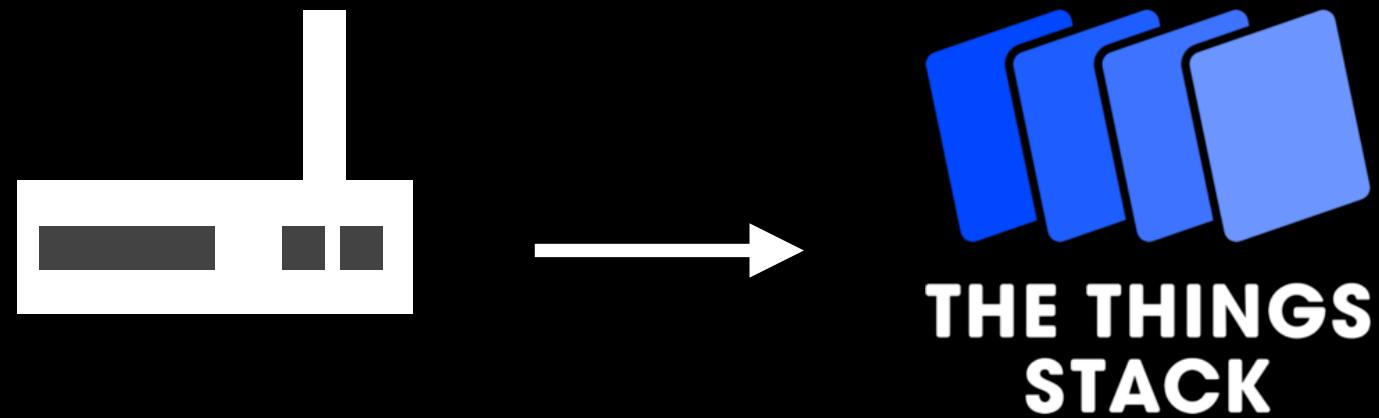


```
CayenneLPP dados(50); // limite de 50 bytes  
// em alguma função do código
```

```
dados.addTemperature(1, 27.5); // temperatura 1 (canal 1)  
dados.addTemperature(2, 25.5); // temperatura 2 (canal 2)  
dados.addPresence(1, true);  
dados.addBarometricPressure(1, 1035);  
dados.addRelativeHumidity(1, 76);  
dados.addAnalogInput(1, 34.55);
```

```
uint8_t* buffer = dados.getBuffer();  
String mensagem = "";  
for (int i = 0; i < dados.getSize(); i++) {  
    if (buffer[i] < 16) {  
        mensagem += "0";  
    }  
    mensagem += String(buffer[i], HEX);  
}  
mensagem.toUpperCase();  
Serial.println(mensagem);  
  
dados.reset();
```

Codificação Cayenne



A screenshot of a GitHub repository page for "TheThingsNetwork/lorawan-stack". The page title is "The Things Stack, an open source LoRaWAN Network Server". The repository description states: "The Things Stack is an open source LoRaWAN network stack suitable for large, global and geo-distributed public and private networks as well as smaller networks. The architecture follows the LoRaWAN Network Reference Model for standards compliance and interoperability. This project is actively maintained by [The Things Industries](#)." The "Features" section lists: • LoRaWAN Network Server

- Supports LoRaWAN 1.0
- Supports LoRaWAN 1.0.1
- Supports LoRaWAN 1.0.2
- Supports LoRaWAN 1.0.3
- Supports LoRaWAN 1.0.4

Network Server: The Things Stack Community Edition



# THE THINGS STACK

Screenshot of the The Things Stack Application Overview page for the "Laboratório de Internet das Coisas" application.

The URL in the browser is [ns.cetuc.puc-rio.br/console/applications/lab-iot](https://ns.cetuc.puc-rio.br/console/applications/lab-iot).

**Left Sidebar:**

- Home
- Applications** (selected)
- Gateways

Search bar: Search, ⌘K

Laboratório de Internet das Coisas

- Application overview (selected)
- End devices
- Live data
- Payload formatters
- Integrations
- Collaborators
- API keys
- General settings

Top end devices: +

Resources: ⓘ Resources

v3.32.3.223021ce8

**Application Overview:**

**Laboratório de Internet das Coisas** (ID: lab-iot)

Last activity 9 hours ago • 5 End devices

**End devices:**

NAME	LAST ACTIVITY
44102	10 hr. ago •
44091	10 hr. ago •
44112	10 hr. ago •
44092	10 hr. ago •
44111	9 hr. ago •

**Network activity:**

- Packets per data rate
- Packets per channel (24 days)

**Unlock the Network Operations Center**  
Quickly troubleshoot issues and get detailed network statistics

**Upgrade now**

Página Principal da The Things Stack



# THE THINGS STACK

 Applications > Laboratório de IoT > End devices > 44102 > **Device overview**

# Default uplink payload formatter

## Setup

**Formatter type \***

CayenneLPP

Custom Javascript formatter

GRPC service

CayenneLPP

None

## Connection information

**MQTT server host**

Public address: endereco.mqtt.com:1883

Public TLS address: endereco.mqtt.com:8883

**Connection credentials**

Username: Meu-usuário

Password: [Generate new API key](#) [Go to API keys](#)

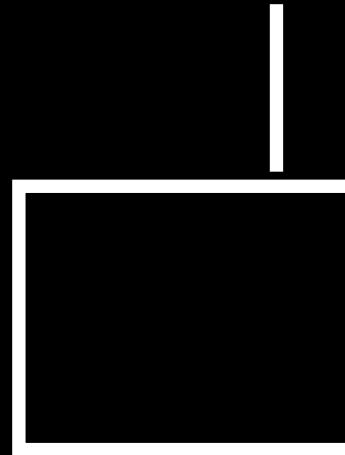




**THE THINGS  
STACK**

Memoriza  
essas chaves!

OK!



## chaves OTAA (Over the Air Activation)

AppEUI: 01 2F 5E 25 90 73 A2 AE "endereço"  
(JoinEUI)

DevEUI: 48 DF D7 81 37 59 E4 23 "usuário"

AppKey: 95 F3 77 2C ... "senha"



THE THINGS  
STACK

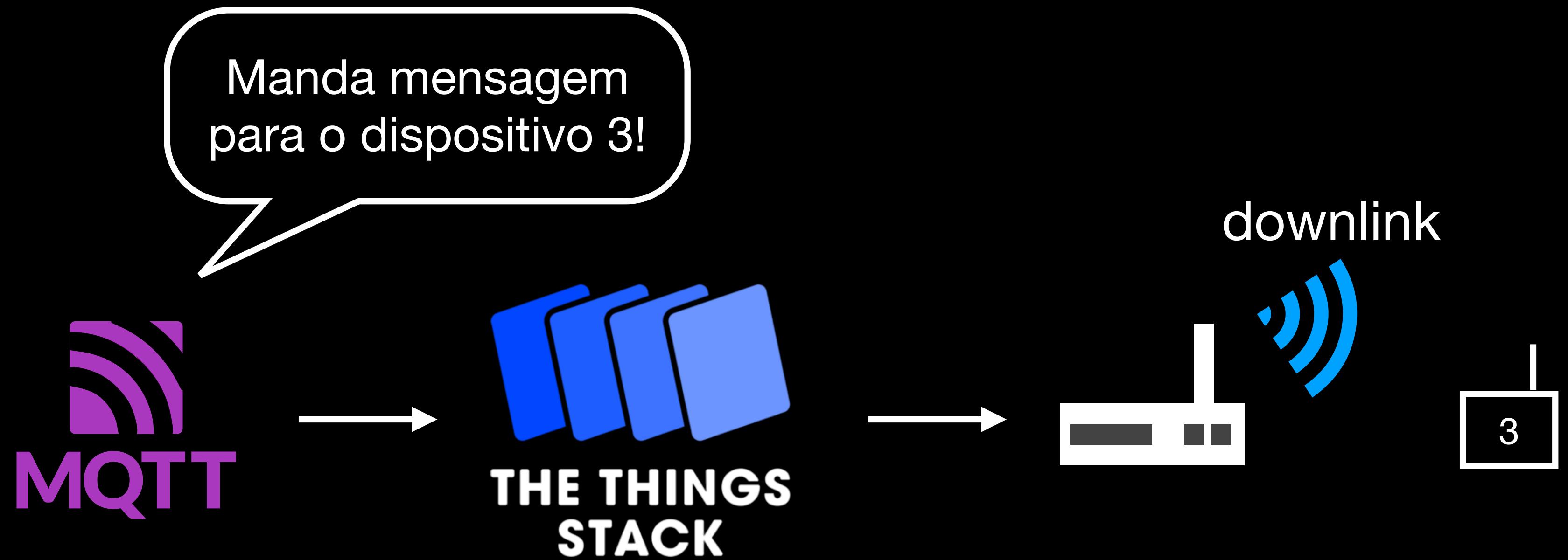


MQTT

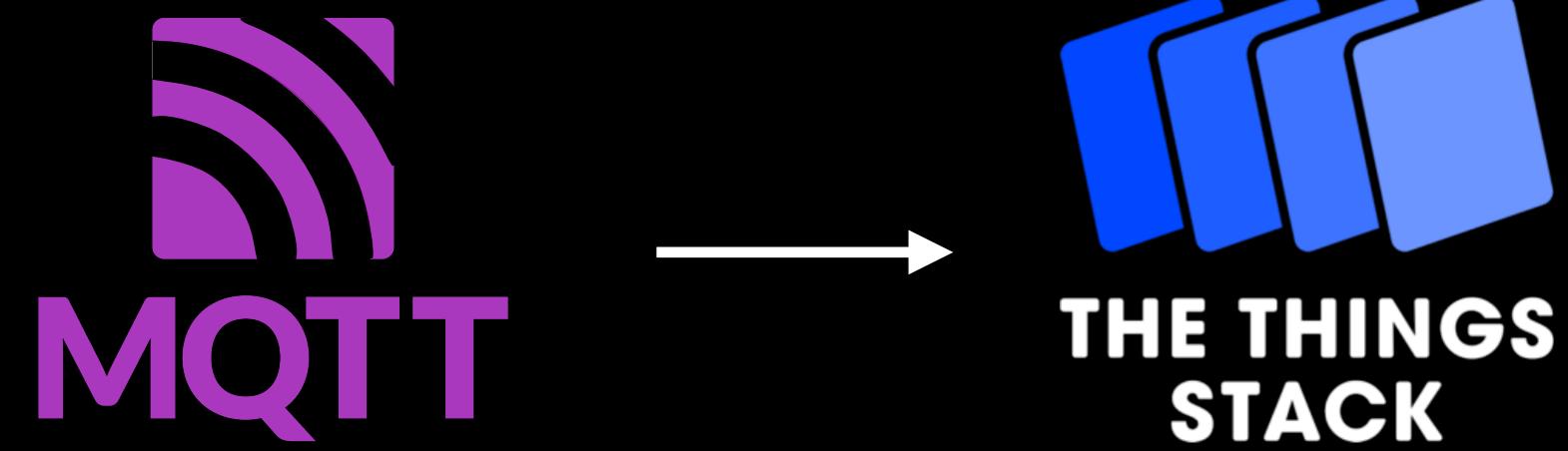
v3/**ID\_DO\_APPLICATIVO**/devices/**ID\_DO\_DISPOSITIVO**/up

```
{  
    "end_device_ids": { "device_id": "ID_DO_DISPOSITIVO", ... },  
    "correlation_ids": [ ... ]  
    "received_at": "2024-10-04T00:14:25.304980849Z",  
    "uplink_message": {  
        "session_key_id": "0ZJeyvBtQqwF7uhARuSkCA==", ← base 64  
        "f_port": 1,  
        "f_cnt": 71,  
        "frm_payload": "Awf/1w==", ← base 64  
        "decoded_payload": { "temperature_1": -4.1 },  
        "rx_metadata": [ { ... } ],  
        "settings": { ... },  
        "received_at": "2024-10-04T00:14:25.096716981Z",  
        ...  
    }  
}
```

Encaminhamento das Mensagens e Eventos para o MQTT



Arquitetura Geral do LoRaWAN para Downlink



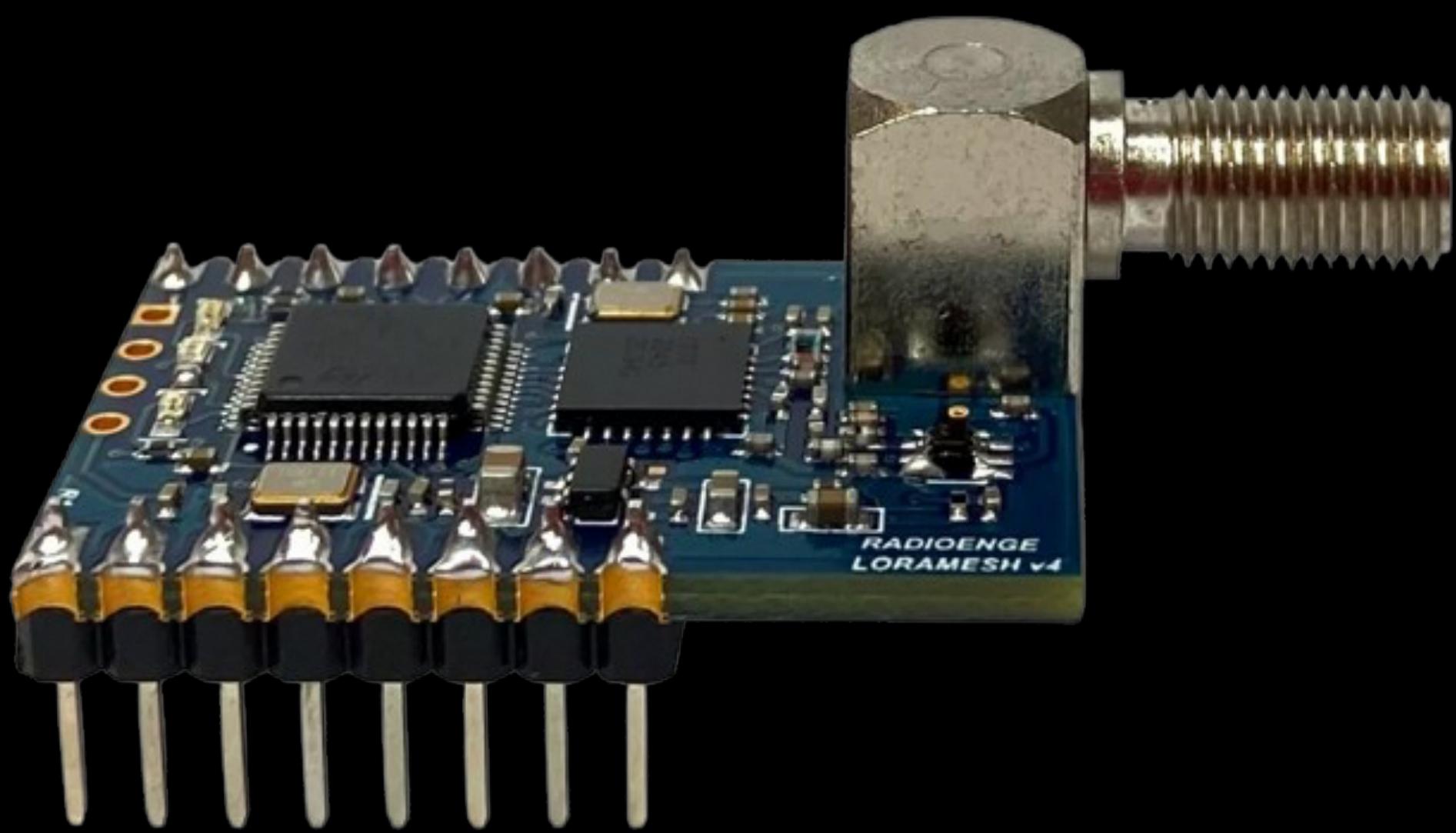
v3/**ID\_APPLICATIVO**/devices/**ID\_DO\_DISPOSITIVO**/down/push

```
{  
  "downlinks": [  
    {"f_port": 5,  
     "frm_payload": "SmFuIEsuIFMu",  
     "priority": "HIGH"  
    }]  
}
```

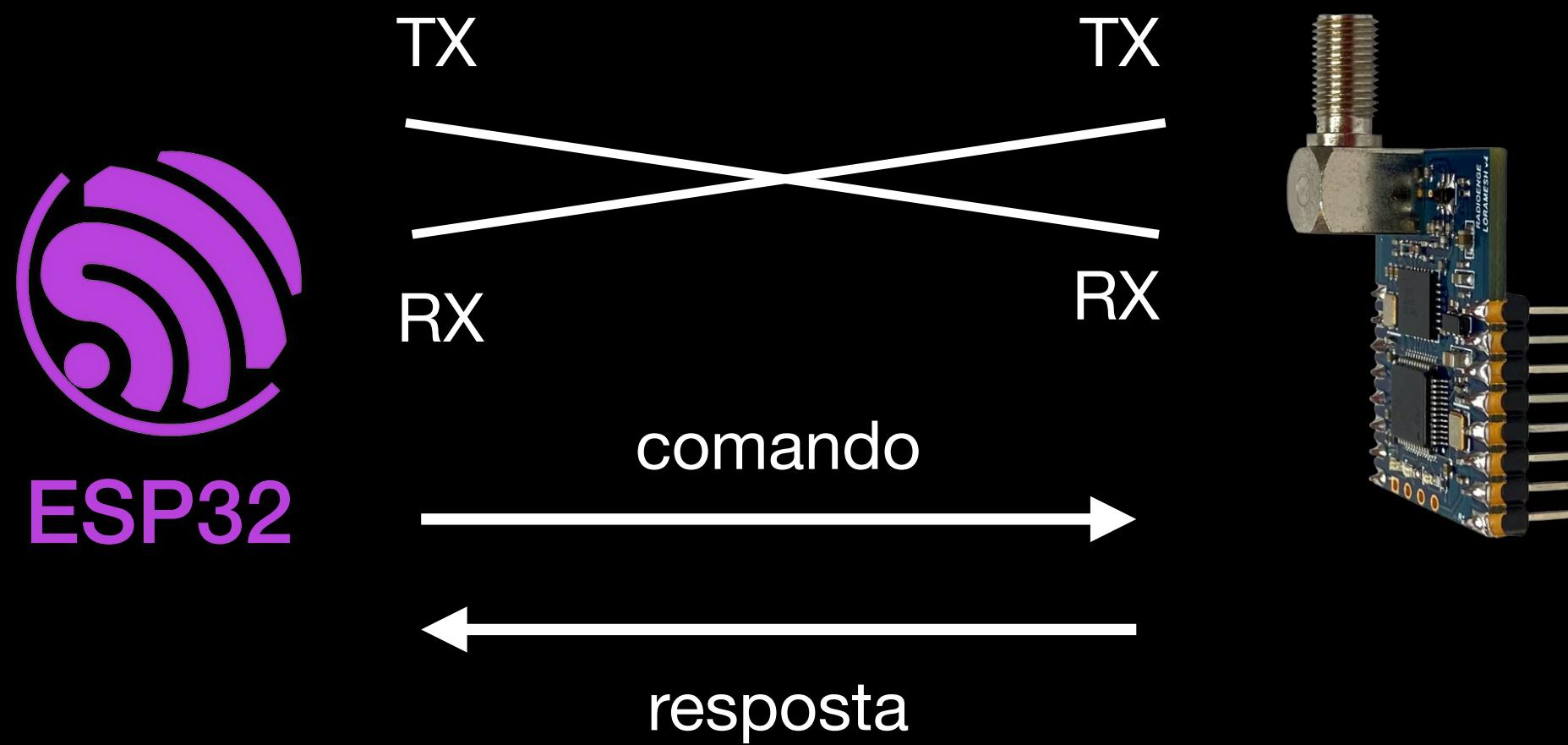
← base 64

A large white arrow points from the bottom-left towards the JSON object, indicating the direction of data flow.

Coisas



Módulo LoRaWAN Radioenge



Objetivo	Formato do Comando
Testar comunicação com modem	AT
Executar comando	AT+COMANDO
Verificar valor de uma propriedade	AT+PROPRIEDADE=?
Definir valor de uma propriedade	AT+PROPRIEDADE=VALOR

Comunicação Serial via Comandos AT

Objetivo	Comando	Respostas
Testar comunicação com módulo	AT	AT_OK
Ler Classe do Dispositivo	AT+CLASS=?	A C
Ajustar Classe do Dispositivo	AT+CLASS=A AT+CLASS=C	AT_OK
Ler Método de Ativação	AT+NJM=?	0 (ABP) 1 (OTAA)
Ler AppKey	AT+APPKEY=?	chave hexadecimal
Ler AppEUI	AT+APPEUI=?	chave hexadecimal
Ler DevEUI	AT+DEUI=?	chave hexadecimal
Conectar na rede	AT+JOIN	AT_JOIN_OK AT_JOIN_ERROR AT_ALREADY_JOINED
Status da conexão	AT+NJS=?	0 (ainda não conectou) 1 (conectado)
Envio de Texto (Uplink na Porta 1)	AT+SEND=1:teste...	AT_TX_OK AT_TX_LENGTH_ERROR
Envio de Hexa (Uplink na porta 1)	AT+SENDB=1:74657374652E2E2E	AT_NO_NETWORK JOINED
Recebimento de Dados (Downlink)	--	RX:4a616e:2:-70:28

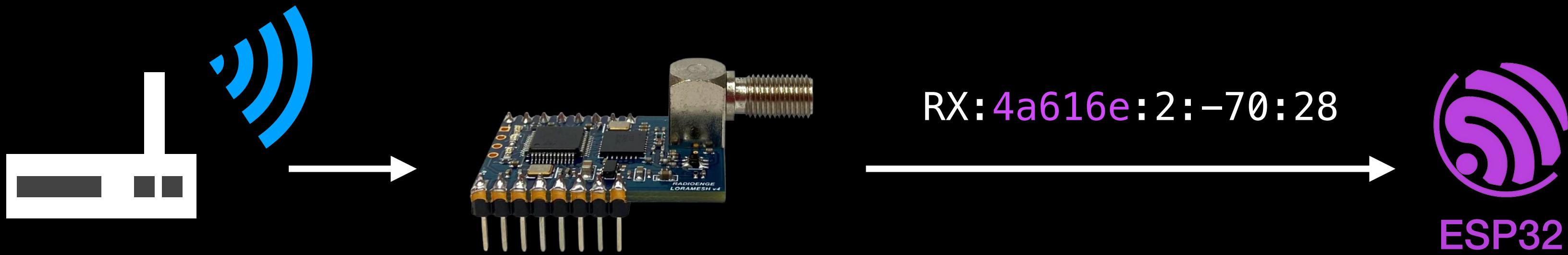
## Exemplos de Comandos AT do Modem

```
void setup() {  
    Serial1.begin(9600, SERIAL_8N1, 47, 48); // pinos 47 e 48  
    Serial.begin(115200); delay(500);  
  
    Serial1.println("AT+JOIN");  
}  
}
```

```
void loop() {  
    if (Serial1.available() > 0) {  
        String texto = Serial1.readStringUntil('\n');  
        texto.trim();  
        Serial.println("Resposta do módulo LoRaWAN: " + texto);  
    }  
}
```

```
AT_OK  
AT_JOIN_OK  
AT_TX_OK  
AT_TX_OK  
RX:4a616e:2:-70:28
```

```
// em alguma função do código...  
  
Serial1.println("AT+SEND=1:0i!"); // porta : texto  
Serial1.println("AT+SENDB=1:A0FF4D"); // porta : dado hexadecimal
```



```

String hexadecimalParaTexto(String textoHexadecimal) {
    String resultado = "";

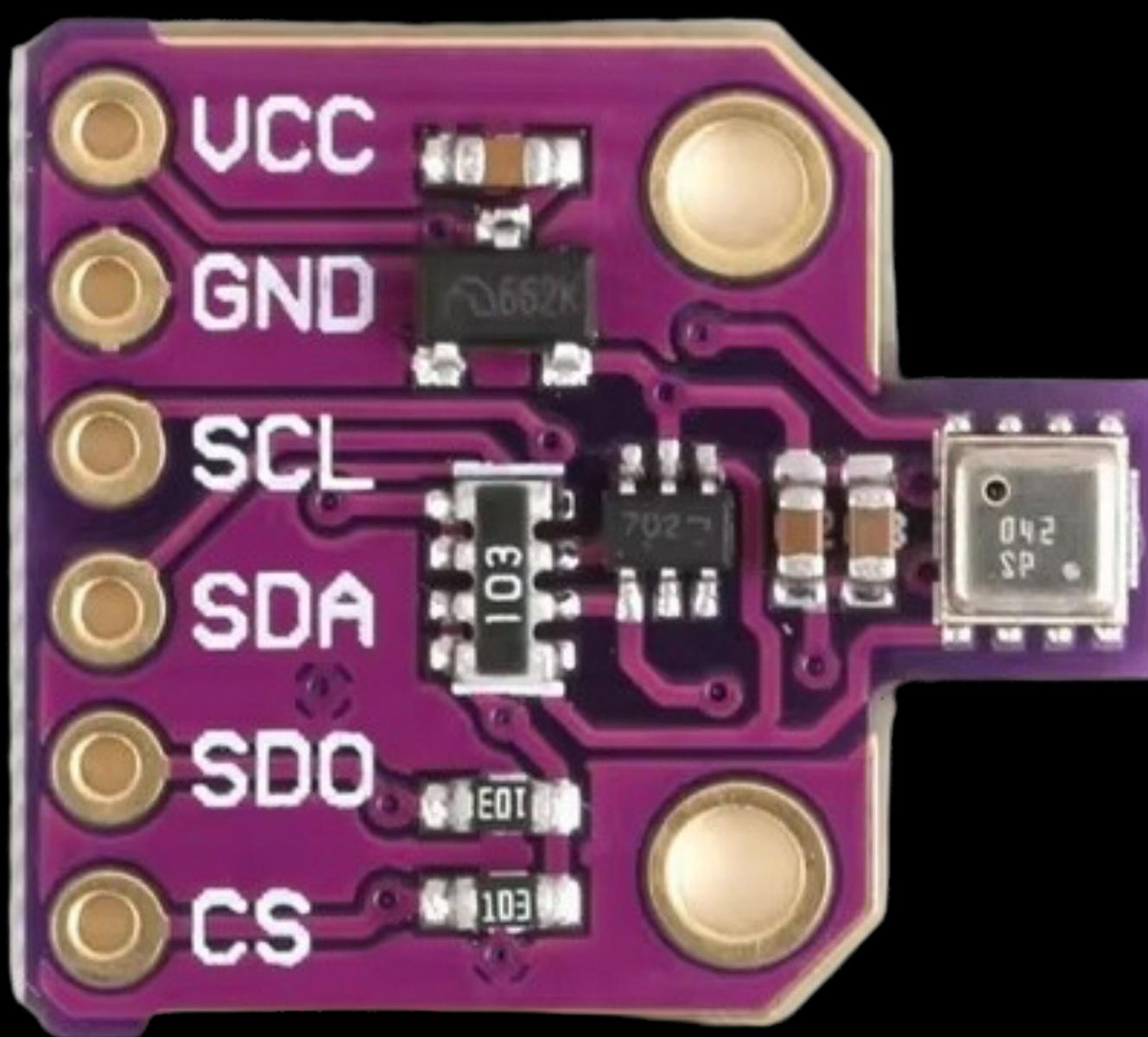
    textoHexadecimal.replace(" ", "");

    // Converte cada par de caracteres hex em um caractere ASCII
    for (int i = 0; i < textoHexadecimal.length(); i += 2) {
        String par = textoHexadecimal.substring(i, i + 2);
        char caractere = (char)strtol(par.c_str(), NULL, 16);
        resultado += caractere;
    }

    return resultado;
}

```

Função de Conversão de Texto Hexadecimal para Texto ASCII



Módulo BME650

```
#include <Adafruit_BME680.h>

Adafruit_BME680 sensorBME;

void setup() {
    Serial.begin(115200); delay(500);

    if (!sensorBME.begin()) {
        Serial.println("Erro no sensor BME");
        while (true);
    }

    // aumenta amostragem dos sensores (1X, 2X, 4X, 8X, 16X ou NONE)
    sensorBME.setTemperatureOversampling(BME680_OS_8X);
    sensorBME.setHumidityOversampling(BME680_OS_2X);
    sensorBME.setPressureOversampling(BME680_OS_4X);

    sensorBME.setIIRFilterSize(BME680_FILTER_SIZE_3);
    sensorBME.setGasHeater(320, 150); // °C e ms, (0, 0) para desativar
}

                                // código continua a seguir...
```

Exemplo de Leitura do BME680

```
// continuação do código...
// em alguma função do código...
sensorBME.performReading();

float temperatura = sensorBME.temperature; // °C
float pressao = sensorBME.pressure / 100.0; // hPa
float altitude = sensorBME.readAltitude(1013.25); // m
float umidade = sensorBME.humidity; // %
float resistencia_gas = sensorBME.gas_resistance / 1000.0; // kΩ
```



ZZZ...

Modo Hibernação do ESP32

# Modo Ativo

50mA - 250mA



# Modo Sono Profundo

0.01mA



Consumo de Energia em Diferentes Modos

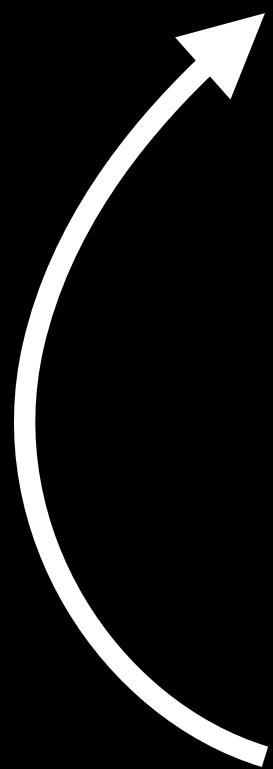
```
void setup() {  
    // ...  
}
```



envia dados



ZZZ...



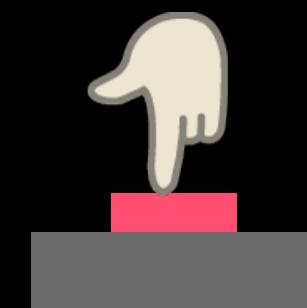
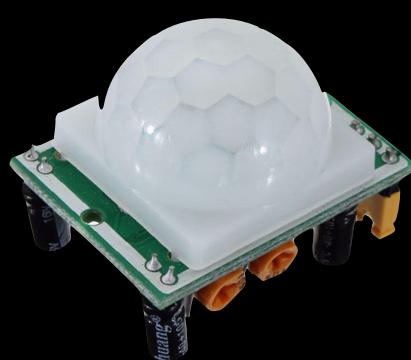
!

acorda com timer

ou

!

acorda com  
interrupção num pino



Eventos para Acordar o ESP32

```

int pinoParaAcordar = 4;

int contador1 = 0; // volta para 0 após dormir
RTC_DATA_ATTR int contador2 = 0; // mantém valor mesmo após dormir

void setup() {
    Serial.begin(115200); delay(500);

    // agenda para acordar depois de 10000000 µs (10 segundos)
    esp_sleep_enable_timer_wakeup(10e6);

    // ou... acorda quando tiver HIGH no pino desejado
    pinMode(pinoParaAcordar, INPUT);
    esp_sleep_enable_ext0_wakeup((gpio_num_t) pinoParaAcordar, HIGH);

    Serial.printf("Contador 1: %d\n", contador1); contador1++;
    Serial.printf("Contador 2: %d\n", contador2); contador2++;

}

        // em alguma função do código...
esp_deep_sleep_start(); // durma agora

```

```

Contador 1: 0
Contador 2: 0
Contador 1: 0
Contador 2: 1
Contador 1: 0
Contador 2: 2

```