



# Radioenge

## Módulo LoRaMESH

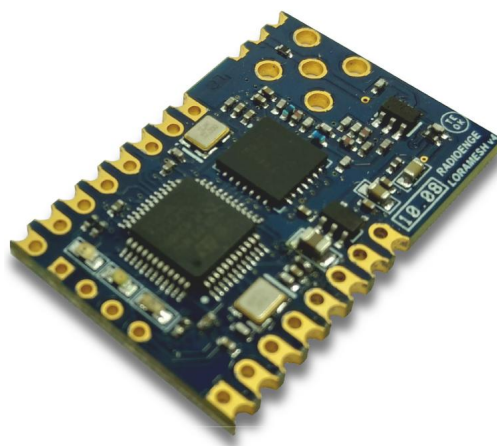
### Manual de Utilização

Revisão - Março de 2024

# Módulo LoRaMESH EndDevice

---

O módulo LoRaMESH EndDevice Radioenge é um transceiver que alia a tecnologia de modulação LoRa®, que proporciona baixo consumo e longo alcance, com uma topologia MESH, que permite a criação de uma ampla área de cobertura com uma rede altamente escalável e de baixo custo. O dispositivo possui duas interfaces seriais UART, sendo uma destinada a configuração e outra para comunicação de dados de forma transparente.



## Recursos

- ▶ Tecnologia LoRaMESH Radioenge;
- ▶ Configurável como mestre ou escravo;
- ▶ 2 interfaces UART (comandos e transparente);
- ▶ Configuração via interface UART ou Rádio;
- ▶ Fixação acastelada ou por barra de pinos;
- ▶ Atualização de firmware via UART ou Rádio;
- ▶ 8 GPIOs, sendo 2 configuráveis como entradas analógicas;
- ▶ 3 LEDs para sinalização de operação;
- ▶ Diagnóstico remoto do dispositivo.

## Características

- ▶ Alimentação de 4 a 12 Vcc (+3,3 Vcc sem regulador);
- ▶ Dimensões: 33 mm x 22 mm x 3 mm;
- ▶ Temperatura de operação: -5°C a +55°C;
- ▶ Baixo consumo;
- ▶  $\mu$ Processador integrado: ARM Cortex-M0+ 32-bits;
- ▶ Taxa de dados máxima: 21,9 kbps (LoRa), 250 kbps (FSK);
- ▶ Topologia Mesh.

## Características de RF

- ▶ Operação na Banda ISM de 915 MHz;
- ▶ Sensibilidade de recepção: -137 dBm (LoRa), -92 dBm (FSK);
- ▶ Potência de transmissão máxima: +20 dBm;
- ▶ Modulação LoRa ou FSK;
- ▶ Homologação ANATEL: 02021-18-07215.

## Aplicações

- ▶ Internet of Things (IoT);
- ▶ Automação doméstica e comercial;
- ▶ Sistemas de segurança e monitoramento;
- ▶ Aquisição e envio de dados;
- ▶ Leitura remota de sensores.

## Sumário

<b>1</b>	<b>Tecnologia LoRaMESH</b>	<b>3</b>
<b>2</b>	<b>Pinagem</b>	<b>4</b>
<b>3</b>	<b>Especificações</b>	<b>5</b>
3.1	Especificações de operação . . . . .	5
3.2	Valores máximos absolutos . . . . .	5
3.3	Parâmetros de operação . . . . .	6
3.4	Taxas de dados – LoRa . . . . .	7
3.5	Payload máximo – LoRa . . . . .	8
3.6	Payload máximo – FSK . . . . .	8
<b>4</b>	<b>Exemplo de Aplicação</b>	<b>9</b>
<b>5</b>	<b>Modos de Operação</b>	<b>10</b>
5.1	Definição de Classe e Janela . . . . .	10
<b>6</b>	<b>Descrição das Interfaces</b>	<b>11</b>
6.1	Interface dos comandos . . . . .	11
6.2	Interface transparente . . . . .	11
<b>7</b>	<b>Descrição do Hardware</b>	<b>12</b>
7.1	GIPOs . . . . .	12
7.2	Saída de antena . . . . .	12
7.3	<i>Bypass</i> do regulador . . . . .	12
7.4	LEDs de sinalização . . . . .	13
<b>8</b>	<b>Descrição do Protocolo de Comunicação</b>	<b>14</b>
<b>9</b>	<b>Lista de comandos detalhada</b>	<b>16</b>
9.1	Leitura e escrita dos parâmetros de rádio (0xD6) . . . . .	16
9.1.1	Leitura (subcomando 0x00) . . . . .	16
9.1.2	Escrita (subcomando 0x01) . . . . .	16
9.2	Leitura local (0xE2) . . . . .	17
9.2.1	Comandos de leitura e configuração de parâmetros . . . . .	17
9.2.1.1	Leitura local (0xE2) . . . . .	17
9.3	Leitura remota (0xD4) . . . . .	17
9.3.0.1	Leitura remota (0xD4) . . . . .	17
9.4	Comando de configuração de rádio (0xCA) . . . . .	17
9.4.0.1	Comando de configuração de rádio (0xCA) . . . . .	17
9.5	Resposta aos comandos de Leitura local, Leitura remota e Comando de configuração de rádio . . . . .	18
9.5.0.1	Resposta aos comandos de Leitura local, Leitura remota e Comando de configuração de rádio . . . . .	18
9.6	Configuração de GPIO (0xC2) . . . . .	20
9.6.1	Configuração (subcomando 0x02) . . . . .	20
9.6.2	Leitura (subcomando 0x00) . . . . .	21
9.6.3	Escrita (subcomando 0x01) . . . . .	21
9.7	Diagnóstico (0xE7) . . . . .	22
9.8	Ruído (0xD8) . . . . .	22
9.9	RSSI (0xD5) . . . . .	23
9.10	Traça rota (0xD2) . . . . .	24
9.11	Teste periódico (0x01) . . . . .	24
9.12	Configuração de tempo periódico (0xCC) . . . . .	25
9.12.1	Leitura (subcomando 0x02) . . . . .	25
9.12.2	Escrita (subcomando 0x01) . . . . .	25
9.13	Configuração de modo de Classe e interface transparente (0xC1) . . . . .	26

9.13.0.1 Modo de operação (subcomando 0x00) . . . . .	26
9.13.1 Comando para interface transparente (subcomando 0x01) . . . . .	26
9.14 Repasse de comando para outro rádio (0xEF) . . . . .	27
9.15 Carrega FW (Loader) e Apaga banco (0xEB) . . . . .	28
9.15.1 Carrega FW (subcomando 0x00) . . . . .	28
9.15.2 Apaga banco (subcomando 0x01) . . . . .	28
9.16 Ativa Banco (0xEA) . . . . .	29
9.17 Apaga Banco (0xE9) . . . . .	30
9.18 Canal Serial (0xD9) . . . . .	31
9.19 Ruído (0xD8) . . . . .	32
9.20 Rota (0xD3) . . . . .	33
9.21 Grava senha (0xCD) . . . . .	34
9.22 ECO (0xA0) . . . . .	35
9.23 Variáveis dinâmicas (0x88) . . . . .	36
9.24 LEDs (0x83) . . . . .	37
9.25 Comando de aplicação (<0x80) . . . . .	38
9.25.1 Configuração de SENHA (0xCD) . . . . .	38
9.25.2 Comando de aplicação (<0x80) . . . . .	39
<b>10 Características Físicas</b> . . . . .	<b>40</b>
10.1 Dimensões . . . . .	40
10.2 Footprint recomendado . . . . .	41
10.2.1 SMD . . . . .	41
10.2.2 PTH . . . . .	42
<b>11 CRC 16</b> . . . . .	<b>43</b>
11.1 Linguagem C . . . . .	43
11.2 Python . . . . .	44
11.3 Java / Android . . . . .	45
<b>12 Contato</b> . . . . .	<b>46</b>

## 1 Tecnologia LoRaMESH

O LoRaMESH é um sistema de comunicação MESH próprio da Radioenge que utiliza a tecnologia LoRa® na camada física. Ele alia o longo alcance na comunicação e o baixo consumo característicos do LoRa®, com a versatilidade e robustez da rede MESH. Em uma rede MESH, cada dispositivo, além de trabalhar como um usuário enviando ou recebendo informações, também funciona como um roteador, encaminhando os pacotes de outros rádios. Este roteamento acontece de forma automática (auto-rota) e, caso um dispositivo utilizado seja retirado, o próprio algoritmo MESH encontrará uma nova rota (auto-cura) restabelecendo a comunicação.

Uma rede LoRaMESH Radioenge é formada por dois tipos distintos de dispositivos: o mestre (único para toda a rede) e os escravos. Um mestre pode enviar comandos ou pacotes com informações para qualquer escravo em uma rede, enquanto um escravo envia informações apenas para o mestre. A única interação entre escravos é para fazer o roteamento dos pacotes. No mestre, normalmente, é feito o escoamento dos dados da rede para uma aplicação. Existem duas identificações para um EndDevice LoRaMESH Radioenge. Primeiramente, existe o Unique ID/Número de Série (4 bytes), que é único para cada rádio produzido. Além desta, existe o ID (2 bytes) de um rádio, o qual é único em uma rede. O ID configurado individualmente pelo próprio usuário.

Desta forma, a rede LoRaMESH Radioenge é uma solução perfeita para projetos de IoT, pois, uma vez estabelecida a rede com o mestre, a auto-cura e a auto-rota da rede MESH possibilitam uma solução robusta a falhas de comunicação e também altamente escalável, uma vez que novos dispositivos são facilmente incorporados à rede.

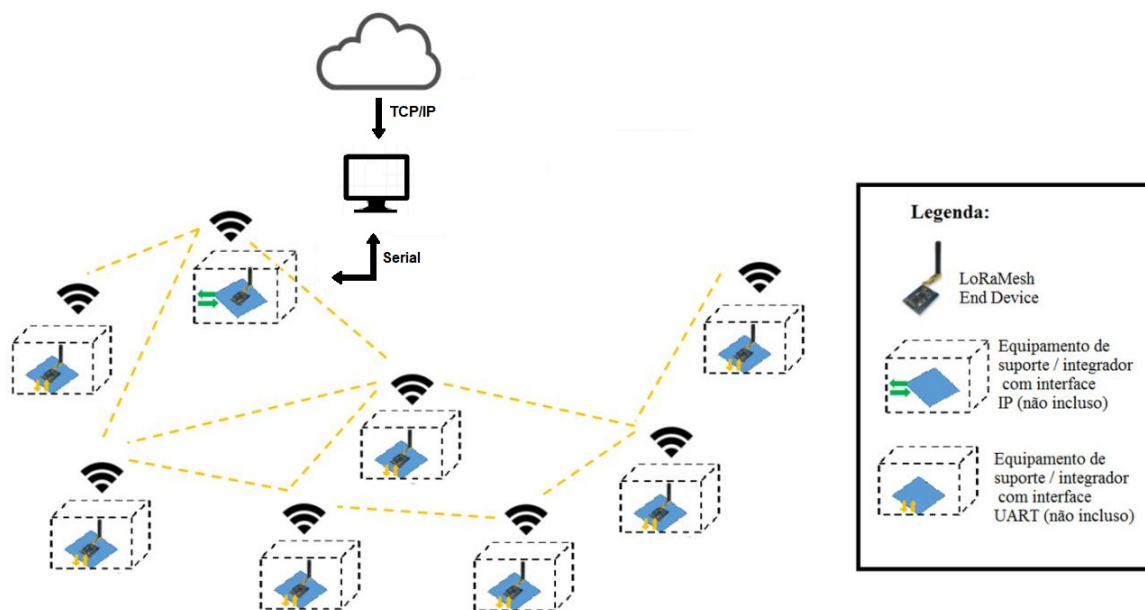


Figura 1: Diagrama de uma rede Mesh

Uma rede é definida por sua SENHA (4 bytes) e pelos parâmetros LoRa, sendo estes: Spreading Factor (SF), Bandwidth (BW) e Coding Rate (CR). Todos estes parâmetros devem ser configurados em cada rádio individualmente, juntamente com o ID. Ao configurar uma SENHA, automaticamente o rádio também definirá um canal de operação entre os 67 disponíveis. No caso de duas ou mais redes trabalhando no mesmo canal, não há problema de um rádio de uma rede receber um comando ou pacote de outra, desde que as senhas sejam diferentes. A SENHA é utilizada para criptografar os pacotes de rádio (AES128).

## 2 Pinagem

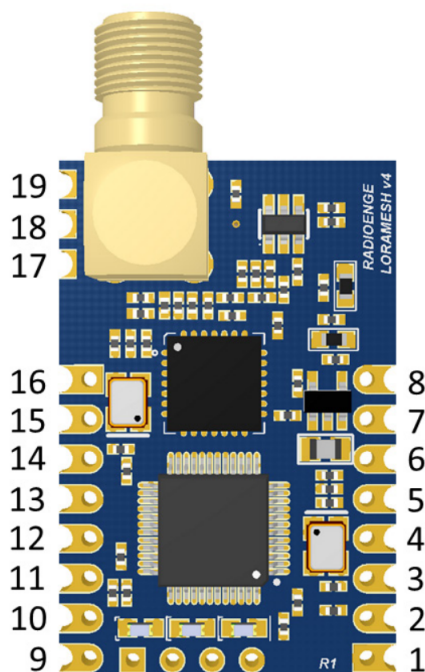


Figura 2: Numeração dos pinos

Tabela 1: Pinagem

Pino	Nome	Tipo	Descrição
1	GND	Alimentação	Conectado ao ground
2	RX_1	Entrada	RX da interface UART de comando
3	TX_1	Saída	TX da interface UART de comando
4	VCC	Alimentação	Conectado à alimentação
5	VCC	Alimentação	Conectado à alimentação
6	TX_2	Saída	TX da interface UART transparente
7	RX_2	Entrada	RX da interface UART transparente
8	GND	Alimentação	Conectado ao ground
9	GPIO0	I/O	Pino de uso geral
10	GPIO1	I/O	Pino de uso geral
11	GPIO2	I/O	Pino de uso geral
12	GPIO3	I/O	Pino de uso geral
13	GPIO4	I/O	Pino de uso geral
14	GPIO5	I/O ou Analógico	Pino de uso geral ou entrada analógica
15	GPIO6	I/O ou Analógico	Pino de uso geral ou entrada analógica
16	GPIO7	I/O	Pino de uso geral
17	GND	Saída RF	Conectado ao ground
18	ANT	Saída RF	Saída de RF para antena externa
19	GND	Saída RF	Conectado ao ground

### 3 Especificações

#### 3.1 Especificações de operação

Tabela 2: Especificação de operação

Especificação	Descrição
Modulação	LoRa® ou FSK
Faixas de frequência	902,5 – 907 MHz e 915 – 928 MHz
Largura de banda	125 kHz / 250 kHz / 500 kHz (LoRa); 250 kHz (FSK)
Spreading Factor (LoRa)	7, 8, 9, 10, 11 e 12
Coding Rate (LoRa)	4/5, 4/6, 4/7 e 4/8
Estabilidade de frequência	±5,0 ppm
Protocolo	LoRaMESH Radioenge
Sensibilidade	-137 dBm (LoRa); -92 dBm (FSK)
Potência de transmissão	Máx. +20 dBm / 100 mW
Conexão com antena	Pad acastelado (pino 18) ou conector SMA-M
Interface de comunicação	UART

#### 3.2 Valores máximos absolutos

Tabela 3: Valores máximo absolutos

Parâmetro	Mínimo	Máximo	Un.
Tensão entre VCC e GND (com regulador)	-0,3	16,0	V
Tensão entre VCC e GND (sem regulador)	-0,3	4,0	V
Tensão nos pinos de UART (classe A)	-0,3	3,6	V
Tensão nos pinos de UART (classe C)	-0,3	5,0	V
Tensão nos pinos GPIO e Analógicos	-0,3	4,0	V
Corrente máxima drenada por uma GPIO	-	16,0	mA
Corrente máxima fornecida por uma GPIO	-	16,0	mA
Corrente máxima drenada por todas as GPIOs	-	90	mA
Temperatura de armazenamento	-10	+70	°C
Temperatura de operação	-5	+55	°C
Potência máxima na entrada RF	-	0	dBm

### 3.3 Parâmetros de operação

Tabela 4: Parâmetros de operação

Parâmetro	Mínimo	Típico	Máximo	Un.
Tensão de Alimentação (com regulador)	1,8	-	12	V
Tensão de Alimentação (sem regulador)	1,8	3,3	3,6	V
Consumo de corrente durante transmissão (Vcc = 3,3V)	-	111	-	mA
Consumo de corrente durante recepção (Vcc = 3,3V)	-	20	-	mA
Tensão de saída em nível baixo (GPIO, I <sub>o</sub> = 8 mA)	-	-	0,4	V
Tensão de saída em nível alto (GPIO, I <sub>o</sub> = 8 mA)	2,9	-	-	V
Limiar de tensão de entrada em nível baixo (GPIO)	-	-	1,0	V
Limiar de tensão de entrada em nível alto (GPIO)	2,3	-	-	V
Limiar de tensão de entrada em nível baixo (USART)	-	-	1,0	V
Limiar de tensão de entrada em nível alto (USART)	2,3	-	-	V
Corrente drenada/fornecida por uma entrada	-	-	50	nA
Faixa de leitura das entradas analógicas	0,0	-	3,3	V
Impedância da entrada analógica	-	-	50	kΩ
Resolução do ADC	8	-	12	bits
Baudrate das interfaces UART	9600	-	57600	bps
Taxa de dados – LoRa	180	-	21900	bps
Taxa de dados – FSK	-	250	-	kbps



### 3.4 Taxas de dados – LoRa

Tabela 5: Taxa de dados LoRa - Coding Rate 4/8

Taxa de dados (bps)		Spreading Factor					
		7	8	9	10	11	12
Largura de Banda (kHz)	125	3418	1953	1098	610	336	183
	250	6836	3906	2197	1220	671	366
	500	13672	7813	4395	2441	1343	732
<b>Coding Rate 4/8</b>							

Tabela 6: Taxa de dados LoRa - Coding Rate 4/7

Taxa de dados (bps)		Spreading Factor					
		7	8	9	10	11	12
Largura de Banda (kHz)	125	3906	2232	1256	698	384	209
	250	7813	4464	2511	1395	767	419
	500	15625	8929	5022	2790	1535	837
<b>Coding Rate 4/7</b>							

Tabela 7: Taxa de dados LoRa - Coding Rate 4/6

Taxa de dados (bps)		Spreading Factor					
		7	8	9	10	11	12
Largura de Banda (kHz)	125	4557	2604	1465	814	448	244
	250	9115	5208	2930	1628	895	488
	500	18229	10417	5859	3255	1790	977
<b>Coding Rate 4/6</b>							

Tabela 8: Taxa de dados LoRa - Coding Rate 4/5

Taxa de dados (bps)		Spreading Factor					
		7	8	9	10	11	12
Largura de Banda (kHz)	125	5469	3125	1758	977	537	293
	250	10938	6250	3516	1953	1074	586
	500	21875	12500	7031	3906	2148	1172
<b>Coding Rate 4/5</b>							

### 3.5 Payload máximo – LoRa

Tabela 9: Payload máximo LoRa - Coding Rate 4/8

Payload máximo (bytes)		Spreading Factor					
		7	8	9	10	11	12
Largura de Banda (kHz)	125	237	237	174	81	18	11
	250	237	237	237	196	92	24
	500	237	237	237	237	218	102
<b>Coding Rate 4/8</b>							

Tabela 10: Payload máximo LoRa - Coding Rate 4/7

Payload máximo (bytes)		Spreading Factor					
		7	8	9	10	11	12
Largura de Banda (kHz)	125	237	237	201	96	24	11
	250	237	237	237	226	108	32
	500	237	237	237	237	237	120
<b>Coding Rate 4/7</b>							

Tabela 11: Payload máximo LoRa - Coding Rate 4/6

Payload máximo (bytes)		Spreading Factor					
		7	8	9	10	11	12
Largura de Banda (kHz)	125	237	237	237	116	31	11
	250	237	237	237	237	130	37
	500	237	237	237	237	237	144
<b>Coding Rate 4/6</b>							

Tabela 12: Payload máximo LoRa - Coding Rate 4/5

Payload máximo (bytes)		Spreading Factor					
		7	8	9	10	11	12
Largura de Banda (kHz)	125	237	237	237	141	45	11
	250	237	237	237	237	158	52
	500	237	237	237	237	237	174
<b>Coding Rate 4/5</b>							

### 3.6 Payload máximo – FSK

Tabela 13: Payload máximo FSK

<b>Payload máximo (bytes)</b>	217
-------------------------------	-----

## 4 Exemplo de Aplicação

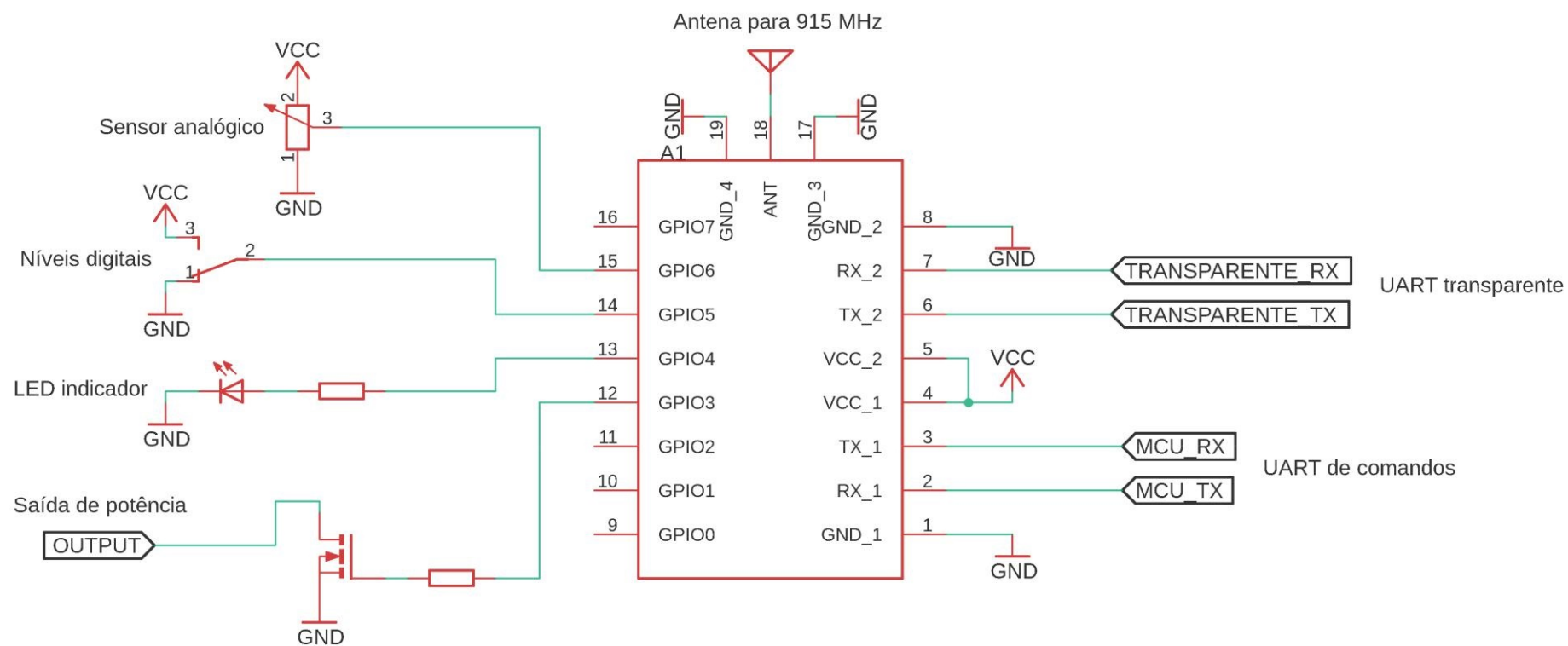


Figura 3: Exemplo de aplicação do EndDevice Radioenge

## 5 Modos de Operação

O EndDevice LoRaMESH Radioenge possui dois tipos de configuração para operar relacionadas com o consumo do rádio no modo de recepção: Classe A e Classe C. A configuração dos modos de operação é feita através do comando 0xC1.

### • Classe A

Configuração válida apenas para o rádio escravo. Quando configurado como Classe A, o dispositivo não entra em modo de recepção; ao invés disso, ele permanece em modo de baixo consumo, não fazendo mais roteamento de pacotes na rede MESH. Neste estado, o escravo apenas envia comandos de aplicação para o mestre ou responde aos comandos localmente.

Sempre que recebe uma mensagem na sua serial, o rádio liga a recepção de acordo com a configuração da JANELA de recepção. Neste momento, ele pode receber as respostas do rádio mestre. As configurações possíveis de janela de recepção são: 5, 10 ou 15s. Este modo é recomendado para aplicação com restrição de consumo como, por exemplo, situações onde o rádio é alimentado por uma bateria.

**Quando configurado em classe A, observar a tensão máxima do barramento serial. Não pode ultrapassar 3,6V, conforme indicado na Tabela 3.**

### • Classe C

Neste modo, o rádio permanece sempre com a recepção da interface de rádio ligada. Quando configurado desta forma, o rádio, além de poder enviar e receber pacotes de rádio a qualquer momento, também faz o roteamento de pacotes de outros escravos. Este modo é recomendado para situações onde não há restrição de consumo.

### 5.1 Definição de Classe e Janela

#### • CLASSE

A classe é o modo de economia de energia do módulo.

- **Classe A:** o módulo opera com baixo consumo de energia. Nesta configuração o módulo não participa da MESH (não repete pacotes) porque se mantém em estado de baixo consumo “deep sleep”. Para iniciar uma transmissão é necessário receber qualquer comando em uma de suas seriais USART.
- **Classe C:** o módulo não economiza energia e participa da rede MESH ativamente.

#### • JANELA

Se módulo estiver configurado como Classe A, a Janela é o intervalo de tempo que o módulo permanecerá com o receptor ligado após uma transmissão. Se estiver configurado como Classe C este parâmetro não terá função.

## 6 Descrição das Interfaces

### 6.1 Interface dos comandos

O módulo LoRaMESH Radioenge possui uma interface serial para configuração dos parâmetros de operação, envio de comandos de rádio e pacotes de informações. Através do comando de Configuração de rádio (0xCA), é possível configurar o baudrate como **9600**, **38400**, **57600** ou **115200** bps; sendo o padrão de fábrica **9600** bps. A Tabela 14 apresenta as configurações da interface serial.

Esta interface implementa o protocolo serial Radioenge, baseado no padrão MODBUS. Este protocolo é descrito em detalhes na Seção 8.

Tabela 14: Parâmetros da interface serial

Parâmetro	Valor
Baudrate	9600, 38400, 57600 e 115200 bps
Pacote	8 bit
Paridade	Não
Stop bit	1 bit
Controle de fluxo	Não

### 6.2 Interface transparente

Além da interface de configuração, o EndDevice LoRaMESH Radioenge possui uma interface serial que pode ser usada para o envio e recebimento de pacotes de dados de forma transparente ao dispositivo. O comando para uso da interface transparente é configurado através do comando 0xC1.

A operação desta interface difere entre o mestre e o escravo. Para o rádio mestre, é preciso indicar para qual rádio a mensagem se destina. Da mesma forma, as mensagens que chegam ao rádio mestre possuem a identificação do rádio de origem. Esta identificação é enviada/recebida juntamente com os dados, conforme a tabela abaixo.

Byte 0	Byte 1	N Bytes
ID (LSB)	ID (MSB)	Payload

No caso de um escravo, apenas o campo *payload* é enviado ou recebido.

Os parâmetros desta interface são os mesmo da interface de comandos. Inclusive, caso seja alterado o baudrate da interface de comandos, também será alterado na transparente.

## 7 Descrição do Hardware

O objetivo desta seção é detalhar algumas configurações possíveis para a utilização do hardware do EndDevice LoRaMESH da Radioenge.

### 7.1 GIPOs

Este dispositivo possui 8 pinos de uso geral, sendo possível configurar dois (GPIO5 e GPIO6) como entradas analógicas (12 bits). As funções de GPIO, tanto de entrada quanto de saída, podem ser utilizadas pela interface serial de comandos ou por rádio através do mestre da rede.

### 7.2 Saída de antena

O dispositivo conta com duas possibilidades de conexão de antena (50  $\Omega$ ):

- ▶ Conector SMA-M, para a conexão direta de uma antena ao módulo;
- ▶ Pad acastelado (Pino 18) para conexão com uma antena integrada de PCI.



*Apenas uma das saídas de antena deve ser utilizada por vez. Caso use o conector SMA, deve-se deixar o pino 18 desconectado. Em caso de usar o pino 18, deve-se remover o conector SMA.*

### 7.3 Bypass do regulador

Em algumas soluções, principalmente com restrição de consumo, é necessário diminuir o consumo do rádio. Além de configurar o rádio para desligar a recepção (Classe A), é possível diminuir ainda mais o consumo fazendo o *bypass* do regulador. Para tal, deve-se **retirar o componente A** e **soldar um resistor 0R 0402 na posição B**, conforme indicado na Figura 4.

Para tais casos, é importante observar os limites da alimentação, conforme indicado na Tabela 4.

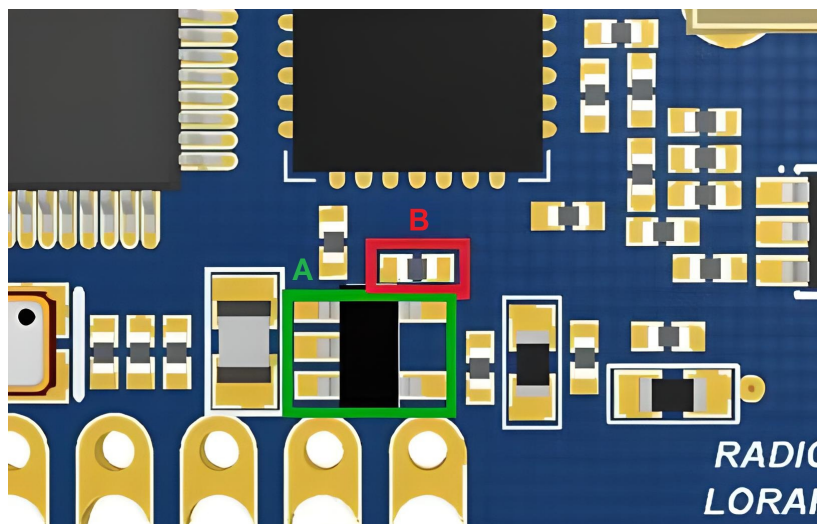


Figura 4: Realização do bypass do regulador

## 7.4 LEDs de sinalização

O módulo possui três LEDs para a sinalização de operação. Cada LED indica uma operação distinta:

- ▶ Vermelho: pisca quando ocorre alguma transmissão via rádio.
- ▶ Verde: pisca quando ocorre alguma recepção via rádio.
- ▶ Amarelo: pisca a cada 1 segundo para indicar que o rádio está com a recepção ativa.

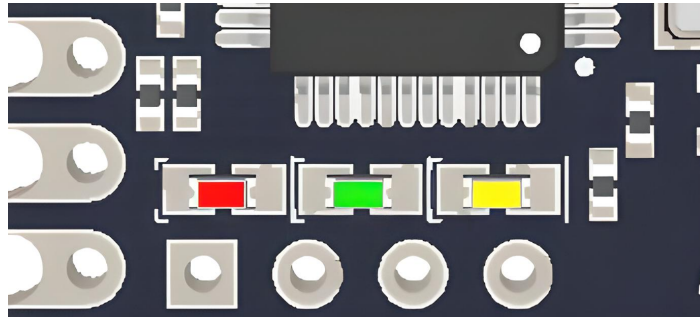


Figura 5: LEDs de sinalização

## 8 Descrição do Protocolo de Comunicação

Os comandos seriais reconhecidos pelo EndDevice LoRaMESH Radioenge seguem o padrão mostrado a seguir:

N Bytes					
ID (LSB)	ID (MSB)	CMD	Payload (n bytes)	CRC (LSB)	CRC (MSB)

- **ID:** indica o identificador único do dispositivo dentro da rede. Existem 4 situações sobre o uso do ID no pacote de comando:
  - 1) Utilizando a serial do rádio mestre:
    - 1.1) Para interrogar o próprio mestre, deve-se colocar a identificação do próprio mestre, ou seja, o ID como 0;
    - 1.2) Para enviar um comando do mestre para o escravo, utilizar o ID do escravo;
  - 2) Utilizando a serial do rádio escravo:
    - 2.1) Para interrogar o próprio escravo, utilizar o ID do próprio escravo;
    - 2.2) Para enviar uma mensagem a partir do escravo para o mestre, utilizar o ID do próprio escravo. Esta mensagem, quando recebida na serial do mestre, conterà nos campos de ID a identificação do rádio que enviou o pacote.
- **CMD:** especifica o comando enviado ao EndDevice, conforme a Tabela 15.
- **Payload:** representam os parâmetros a serem enviados ou dados de aplicação.
- **CRC:** é utilizado para a verificação da integridade do pacote. Caso a cálculo do CRC indique alguma falha no pacote, o rádio apenas descartará a informação recebida.

**O ID 2047 é um endereço especial utilizado apenas no rádio mestre da rede MESH e que permite transmitir um determinado pacote para todos os rádios da rede (broadcast).** Por exemplo, pode-se enviar um comando para acionar uma GPIO de todos os módulos da rede endereçando o comando para o ID 2047.



**Tabela 15: Lista de comandos seriais**

<b>Comando (HEX)</b>	<b>Nome</b>	<b>Descrição</b>
0xD6	Leitura e escrita de parâmetros de rádio	Leitura e escrita dos parâmetros de rádio (LoRa)
0xE2	Leitura local	Leitura dos parâmetros do módulo (apenas pela serial local)
0xD4	Leitura remota	Leitura dos parâmetros de um rádio escravo. Realizado apenas através do mestre da rede
0xCA	Configuração de rádio	Escrita dos parâmetros de um rádio (ID e Máscara binária)
0xC2	Configuração de GPIO	Comando de configuração, leitura e escrita de GPIO
0xE7	Diagnóstico	Adquire informações de operação de um rádio (local ou remoto)
0xD8	Leitura ruído	Leitura do nível de ruído observado por um rádio (local ou remoto)
0xD5	Leitura RSSI	Retorna os níveis de potência de sinal observados (ida e volta) no enlace entre dois rádios
0xD2	Traça rota	Retorna a rota utilizada para se comunicar com um determinado rádio
0x01	Teste periódico	Teste periódico enviado dos escravos para o mestre. Configurável pelo comando 0xCA
0xCC	Configuração de tempo periódico	Configura ou lê o tempo periódico no rádio mestre
0xC1	Configuração de modo de Classe e interface transparente	Configura ou lê a classe do dispositivo e o comando da interface transparente
0xEF	Repasse de comando para outro rádio	Repasa um subcomando de um rádio para outro rádio no modo ponto a ponto
0xEB	Carrega FW e Apaga Banco	Carrega FW (loader) e apaga a memória do banco para atualização de FW
0xEA	Ativa Banco	Ativa o FW carregado
0xD9	Canal Serial	Seta canal de operação do rádio pela interface serial
0xD8	Ruído	Leitura do nível de ruído no canal atual
0xD3	Rota	Define a rota de comunicação estática com um determinado rádio
0xCD	Grava senha	Configura a senha do rádio selecionado
0xA0	Comando de ECO	Envia pacotes de dados para realizar testes na rede
0x88	Variáveis dinâmicas	Configura variáveis dinâmicas de funcionamento
0x83	LEDs	Configura o acendimento dos leds
0xCD	Configuração de senha	Configura a SENHA (serial apenas)
Demais comandos abaixo de 0x80	Comandos de aplicação	Qualquer comando neste intervalo é enviado integralmente à interface de comandos do destino

**Para os comandos de 0x02 até 0x0A, é possível configurar para que o mestre responda os escravos de forma automática. Esta configuração é feita através do comandos 0xCA.**

Também existe um comando reservado para a interface transparente. Este comando é configurado individualmente por rádio. Assim, quando um determinado rádio recebe este comando, além de enviar o pacote inteiro para a interface de comandos, também envia o *payload* para a outra interface. Da mesma forma, quando é utilizada a interface transparente de forma local, o dispositivo encapsula o *payload* recebido utilizando o comando configurado antes de enviar para a rede.

## 9 Lista de comandos detalhada

A lista a seguir apresenta o formato dos pacotes para cada comando. Campos não explicitados são dispostos na forma Little-Endian.

### 9.1 Leitura e escrita dos parâmetros de rádio (0xD6)

#### 9.1.1 Leitura (subcomando 0x00)

##### Envio

Byte	0	1	2	3	4	5	6	7
Dado	ID (LSB)	ID (MSB)	0xD6	0x00	0x01	0x00	CRC (LSB)	CRC (MSB)

##### Resposta

Byte	0	1	2	3	4	5	6	7	8	9
Dado	ID (LSB)	ID (MSB)	0xD6	0x00	Potência	BW	SF	CR	CRC (LSB)	CRC (MSB)

- **Potência:** nível da potência de saída em dBm, sendo 20 (0x14) o nível máximo e também a configuração padrão de fábrica;
- **BW:** Bandwidth da configuração LoRa, as configurações possíveis são 0, 1 e 2 conforme:
  - **0x00:** 125 kHz;
  - **0x01:** 250 kHz;
  - **0x02:** 500 kHz.
- **SF:** Spreading Factor da configuração LoRa. Deve ter um valor entre 7 e 12 ou zero conforme abaixo:
  - **0x07 a 0x0C:** indica operação com modulação LoRa;
  - **0x00:** indica operação em modo FSK.
- **CR:** Coding Rate da configuração LoRa. Deve ter um valor entre 1 e 4 conforme abaixo:
  - **0x01:** 4/5;
  - **0x02:** 4/6;
  - **0x03:** 4/7;
  - **0x04:** 4/8.

#### 9.1.2 Escrita (subcomando 0x01)

##### Envio

Byte	0	1	2	3	4	5	6	7	8	9
Dado	ID (LSB)	ID (MSB)	0xD6	0x01	Potência	BW	SF	CR	CRC (LSB)	CRC (MSB)

##### Resposta

Byte	0	1	2	3	4	5	6	7	8	9
Dado	ID (LSB)	ID (MSB)	0xD6	0x01	Potência	BW	SF	CR	CRC (LSB)	CRC (MSB)

## 9.2 Leitura local (0xE2)

### 9.2.1 Comandos de leitura e configuração de parâmetros

Os comandos de Leitura local (0xE2) e Leitura remota (0xD4) são utilizados para requisitar o ID (no caso da serial local) e o Unique ID/Número de série de um dispositivo na rede. Este último parâmetro é utilizado para a configuração do rádio através do comando 0xCA. As respostas estão descritas na seção 9.5.0.1, juntamente com as definições dos campos.

#### 9.2.1.1 Leitura local (0xE2)

Leitura dos parâmetros do rádio local (serial). Para requisitar as informações de um escravo através do mestre, utilize o comando 0xD4. Como este comando é utilizado para descobrir o ID de um rádio local, o dispositivo responderá independentemente do ID enviado.

##### Envio

Byte	0	1	2	3-5	6	7
Dado	ID (LSB)	ID (MSB)	0xE2	0x00	CRC (LSB)	CRC (MSB)

## 9.3 Leitura remota (0xD4)

### 9.3.0.1 Leitura remota (0xD4)

Comandos para requisitar os parâmetros de um rádio escravo através do mestre.

##### Envio

Byte	0	1	2	3-5	6	7
Dado	ID (LSB)	ID (MSB)	0xD4	0x00	CRC (LSB)	CRC (MSB)

## 9.4 Comando de configuração de rádio (0xCA)

### 9.4.0.1 Comando de configuração de rádio (0xCA)

##### Envio

Byte	0	1	2	3-4	5	6
Dado	ID (LSB)	ID (MSB)	0xCA	0x00	UID (MSB)	UID

7	8	9	10-11	12	13	14
UID_2	UID_2 (LSB)	Máscara de configuração	0x00	BAUDRATE serial	CRC (LSB)	CRC (MSB)

## 9.5 Resposta aos comandos de Leitura local, Leitura remota e Comando de configuração de rádio

### 9.5.0.1 Resposta aos comandos de Leitura local, Leitura remota e Comando de configuração de rádio

Os comandos de leitura local, leitura remota e configuração de rádio possuem o mesmo formato de resposta, sendo diferenciados apenas pelo byte de comando (CMD), conforme a tabela abaixo.

#### Resposta

Byte	0	1	2	3	4	5	6
Dado	ID (LSB)	ID (MSB)	CMD	RSV	RSV	UID (MSB)	UID
7	8	9	10	11	12	13	14
UID	UID (LSB)	RSV	RSV	Revisão de FW	Canal	RSV	RSV
15	16	17	18	19	20	21	22
RSV	BAUDRATE Serial	Versão de FW	Banco de memória	RSV	Teste Periódico (MSB)	Teste Periódico (LSB)	RSV
23	24	25	26	27	28	29	30
RSV	RSV	RSV	RSV	Máscara de configuração	0x01	CRC (LSB)	CRC (MSB)

- **CMD:** indica qual foi o comando que gerou esta resposta, podendo ser:
  - **0xE2:** Leitura local;
  - **0xD4:** Leitura remota;
  - **0xCA:** Comando de configuração de rádio.
- **UID:** é o Unique ID/Número de série do dispositivo, sendo único por rádio fabricado;
- **Versão e Revisão de Firmware:** mostra qual a versão de firmware está programada no dispositivo. Por exemplo, caso a versão seja a 3.18, os bytes versão e revisão de firmware indicarão 3 e 18 respectivamente;
- **Canal:** indica em qual canal de RF o rádio está operando. Pode assumir um valor de 0 a 66;
- **BAUDRATE Serial:** Baud rate utilizado na comunicação serial. Pode assumir os valores entre 0 e 3, conforme abaixo:
  - **0x00:** 9600 bps;
  - **0x01:** 38400 bps;
  - **0x02:** 57600 bps;
  - **0x03:** 115200 bps.
- **Banco de memória:** indica em qual banco de memória o firmware do rádio está gravado, sendo 0 o banco A e 1, o banco B;
- **BAUDRATE Serial:** indica o baudrate da serial, podendo ser:
  - 0:** 9600 bps;
  - 1:** 38400 bps;
  - 2:** 57600 bps;
  - 3:** 115200 bps;
- **Teste Periódico:** tempo do teste que é realizado periodicamente dos escravos para o mestre, em segundos;

- **Máscara de configuração:** máscara binária para configuração do rádio, válida unicamente no mestre:
  - **bit 0:** caso seja configurado como 1, todos os pacotes de teste periódico recebidos pelo mestre serão enviados para a serial de comandos;
  - **bit 1:** caso seja configurado como 1, o mestre responderá de forma automática os pacotes com os comandos de 0x02 até 0x0A;Demais bits são de uso reservado para versões futuras.

## 9.6 Configuração de GPIO (0xC2)

O comando 0xC2 permite configurar ou operar (leitura ou escrita) um pino específico de GPIO disponível, dependendo do subcomando utilizado. Todos os GPIOs disponíveis podem ser utilizados como entrada ou saída digital. No caso da entrada analógica (ADC de 12 bits), apenas os pinos GPIO5 e GPIO6 podem ser configurados como tal.

- **Pino:** valor do GPIO (0,1,2...) que será configurado ou operado. Pode assumir valores entre 0 e 7, relativos aos GPIO0 (0x00) a GPIO7 (0x07).
- **PULL:** indica o estado do resistor interno. Pode assumir os valores 0, 1 ou 2 conforme abaixo:
  - **0x00:** desabilitado;
  - **0x01:** pull up;
  - **0x02:** pull down.
- **INOUT:** indica o tipo do GPIO. Pode assumir os valores 0, 1 ou 3, conforme abaixo:
  - **0x00:** entrada digital;
  - **0x01:** saída digital;
  - **0x03:** entrada analógica (GPIO5 ou GPIO6 apenas).
- **NÍVEL:** indica o nível lógico de saída no comando de escrita. Pode assumir os valores 0 ou 1, conforme abaixo:
  - **0x00:** nível lógico baixo;
  - **0x01:** nível lógico alto.
- **Erro:** o campo erro na resposta indica se a operação foi bem sucedida (0x00) ou não (0x01). Caso não tenha sido, a resposta virá com as posições Pino, PULL e INOUT zeradas. Caso contrário, virá com as configurações que foram salvas.

### 9.6.1 Configuração (subcomando 0x02)

#### Envio

Byte	0	1	2	3	4	5	6	7	8
Dado	ID (LSB)	ID (MSB)	0xC2	0x02	Pino	PULL	INOUT	CRC (LSB)	CRC (MSB)

#### Resposta

Byte	0	1	2	3	4	5	6	7	8	9
Dado	ID (LSB)	ID (MSB)	0xC2	0x02	Erro	Pino	PULL	INOUT	CRC (LSB)	CRC (MSB)

### 9.6.2 Leitura (subcomando 0x00)

Realiza a leitura de um pino configurado tanto como analógico quanto digital. A diferenciação é indicada na resposta do comando.

#### Envio

Byte	0	1	2	3	4	5-6	7	8
Dado	ID (LSB)	ID (MSB)	0xC2	0x00	Pino	0x00	CRC (LSB)	CRC (MSB)

#### Resposta

Byte	0	1	2	3	4	5	6-7	8	9
Dado	ID (LSB)	ID (MSB)	0xC2	0x00	Erro	Pino	Payload	CRC (LSB)	CRC (MSB)

O campo **Payload** traz o resultado da leitura do pino especificado.

#### Payload

Bit	15	14-12	11-0
Dado	An/Dig.	RSV	Leitura

- **An/Dig:** indica se a leitura é digital (0x01) ou analógica (0x00).
- **Leitura:** caso a leitura seja do tipo analógica, ela variará entre 0 e 4095, onde o primeiro valor representa 0 V e o segundo, a tensão de alimentação do microcontrolador, a qual depende da configuração do regulador (com ou sem *bypass*). Para o caso de leitura digital, o campo leitura poderá assumir apenas dois valores: 0 para nível lógico baixo ou 1 para nível lógico alto.

### 9.6.3 Escrita (subcomando 0x01)

Envia o nível lógico desejado para um pino configurado como saída

#### Envio

Byte	0	1	2	3	4	5	6	7	8
Dado	ID (LSB)	ID (MSB)	0xC2	0x01	Pino	NÍVEL	0x00	CRC (LSB)	CRC (MSB)

#### Resposta

Byte	0	1	2	3	4	5	6	7	8
Dado	ID (LSB)	ID (MSB)	0xC2	0x01	Erro	0x00	NÍVEL	CRC (LSB)	CRC (MSB)

## 9.7 Diagnóstico (0xE7)

Envia requisição de informações de diagnóstico, como temperatura do *chip* e tensão de alimentação.

### Envio

Byte	0	1	2	3-5	6	7
Dado	ID (LSB)	ID (MSB)	0xE7	0x00	CRC (LSB)	CRC (MSB)

### Resposta

Byte	0	1	2	3	4	5	6
Dado	ID (LSB)	ID (MSB)	0xE7	Temp. mínima	Temp. atual	Temp. máxima	Tensão mínima

7	8	9-14	15	16-26	27	28
Tensão atual	Tensão máxima	RSD	0x02	RSD	CRC (LSB)	CRC (MSB)

- As temperaturas são dadas em °C com valores inteiros.
- As tensões são dadas em  $V_{cc} \times 10$ . Por exemplo, 3,3 V é representado pelo valor inteiro 33.

## 9.8 Ruído (0xD8)

Retorna o valor do ruído mínimo, máximo e médio observado pelo rádio no canal atual.

### Envio

Byte	0	1	2	3-5	6	7
Dado	ID (LSB)	ID (MSB)	0xD8	0x00	CRC (LSB)	CRC (MSB)

### Resposta

Byte	0	1	2	3	4	5	6	7
Dado	ID (LSB)	ID (MSB)	0xD8	Ruído mínimo	Ruído médio	Ruído máximo	CRC (LSB)	CRC (MSB)

O ruído é medido em dBm, sendo indicado apenas o valor em módulo, pois é uma variável sempre negativa. Assim, caso um dos campos de ruído retorne, por exemplo, 0x73, o valor do ruído será na realidade -115 dBm.



## 9.9 RSSI (0xD5)

Este comando permite verificar a qualidade do enlace entre dois rádios. Obviamente, por ser uma rede MESH, podem existir diversos saltos entre o mestre e o rádio destino. Este comando retorna os níveis de sinal recebidos (ida e volta) apenas do último salto entre o destino e o rádio que se comunica diretamente com ele, cujo ID está indicado no campo Gateway.

Novamente, o valor indicado para a potência de nível é em dBm com o valor em módulo, pois a medida real é sempre um valor negativo. Assim, caso um dos campos de RSSI seja 0x80, por exemplo, o valor real medido no receptor será de -128 dBm.

### Envio

Byte	0	1	2	3	4	5-7	8	9
Dado	ID (LSB)	ID (MSB)	0xD5	0x01	0x02	0x00	CRC (LSB)	CRC (MSB)

### Resposta

Byte	0	1	2	3	4	5
Dado	ID (LSB)	ID (MSB)	0xD5	GW (LSB)	GW (MSB)	RSSI IDA
6	7	8	9	10	11	12
RSSI VOLTA	SNR IDA	SNR VOLTA	RSV	RSV	CRC (LSB)	CRC (MSB)

- **GW:** ID do rádio que está se comunicando com o destino (Gateway);
- **RSSI IDA:** valor em módulo e em dBm da potência de sinal recebida no rádio destino;
- **RSSI VOLTA:** valor em módulo e em dBm da potência de sinal recebida no Gateway;
- **SNR IDA:** relação sinal-ruído medida no receptor do rádio destino. É medido em dB;
- **SNR VOLTA:** relação sinal-ruído medida no receptor do Gateway. É medido em dB.

### 9.10 Traça rota (0xD2)

Indica os IDs dos rádios que são utilizados na rota de comunicação do mestre com o rádio destino. O tamanho da resposta depende da quantidade de saltos necessários para alcançar o destino.

#### Envio

Byte	0	1	2	3-5	6	7
Dado	ID (LSB)	ID (MSB)	0xD2	0x00	CRC (LSB)	CRC (MSB)

#### Resposta

Byte	0	1	2	3	4	...	N+5	N+6
Dado	ID (LSB)	ID (MSB)	0xD2	ID1 (LSB)	ID1 (MSB)	...	CRC (LSB)	CRC (MSB)

### 9.11 Teste periódico (0x01)

#### Pacote recebido no mestre

Byte	0	1	2	3	4	5	6	7
Dado	ID (LSB)	ID (MSB)	0x01	TIPO	UID (MSB)	UID	UID	UID (LSB)

8	9	10	11	12	13
Tensão	Temp	RSV	SEQ	CRC (LSB)	CRC (MSB)

- **TIPO:** pode valer 0, 1 ou 2, conforme abaixo:
  - **0x00:** rádio resetou;
  - **0x01:** teste normal;
  - **0x02:** perdeu a comunicação.
- **UID:** Unique ID/NS do rádio escravo;
- **Tensão:** tensão de alimentação do rádio.

As informações recebidas são as do rádio escravo que enviou o pacote de teste periódico.

## 9.12 Configuração de tempo periódico (0xCC)

Comando de leitura ou gravação do tempo de teste periódico, sendo feito apenas no mestre. O tipo da operação é controlada pelo subcomando. O tempo de teste periódico é indicado em segundos.

### 9.12.1 Leitura (subcomando 0x02)

#### Envio

Byte	0	1	2	3	4-5	6	7
Dado	ID (LSB)	ID (MSB)	0xCC	0x02	0x00	CRC (LSB)	CRC (MSB)

#### Resposta

Byte	0	1	2	3	4	5	6	7
Dado	ID (LSB)	ID (MSB)	0xCC	0x02	Teste Periódico (LSB)	Teste Periódico (MSB)	CRC (LSB)	CRC (MSB)

### 9.12.2 Escrita (subcomando 0x01)

#### Envio

Byte	0	1	2	3	4	5	6	7
Dado	ID (LSB)	ID (MSB)	0xCC	0x01	Teste Periódico (LSB)	Teste Periódico (MSB)	CRC (LSB)	CRC (MSB)

#### Resposta

Byte	0	1	2	3	4	5	6	7
Dado	ID (LSB)	ID (MSB)	0xCC	0x01	Teste Periódico (LSB)	Teste Periódico (MSB)	CRC (LSB)	CRC (MSB)

### 9.13 Configuração de modo de Classe e interface transparente (0xC1)

Comando para configuração ou leitura do modo de operação do rádio e também do comando para uso da interface transparente. O tipo da operação é controlada pelo subcomando utilizado.

#### 9.13.0.1 Modo de operação (subcomando 0x00)

Comando de configuração:

##### Envio

Byte	0	1	2	3	4	5	6	7
Dado	ID (LSB)	ID (MSB)	0xC1	0x00	CLASSE	JANELA	CRC (LSB)	CRC (MSB)

##### Resposta

Byte	0	1	2	3	4	5	6	7	8
Dado	ID (LSB)	ID (MSB)	0xC1	0x00	ERRO	CLASSE	JANELA	CRC (LSB)	CRC (MSB)

- **ERRO:** indica se a configuração foi bem sucedida (0x00) ou não (0x01);
- **CLASSE:** indica a Classe a ser configurada. Pode assumir os valores 0 ou 2, conforme abaixo:
  - **0x00:** Classe A;
  - **0x02:** Classe C.
- **JANELA:** Indica o tempo em que o receptor ficará ligado após acordar o rádio (válido apenas para Classe A). Pode assumir os valores 0, 1 ou 2 conforme abaixo:
  - **0x00:** 5s;
  - **0x01:** 10s;
  - **0x02:** 15s.

A leitura pode ser feita colocando ambos, CLASSE e JANELA, como 0xFF.

#### 9.13.1 Comando para interface transparente (subcomando 0x01)

##### Envio

Byte	0	1	2	3	4	5	6
Dado	ID (LSB)	ID (MSB)	0xC1	0x01	TRANSP	CRC (LSB)	CRC (MSB)

##### Resposta

Byte	0	1	2	3	4	5	6	7
Dado	ID (LSB)	ID (MSB)	0xC1	0x01	ERRO	TRANSP	CRC (LSB)	CRC (MSB)

- **ERRO:** indica se a configuração foi bem sucedida (0x00) ou não (0x01);
- **TRANSP:** comando a ser interpretado como direcionado para a interface transparente. Deve assumir um valor entre 0x02 e 0x31, com estes inclusos.

A leitura pode ser feita colocando TRANSP como 0xFF.

### 9.14 Repasse de comando para outro rádio (0xEF)

Este comando repassa um subcomando de um rádio para outro no modo ponto a ponto. É útil quando queremos forçar a comunicação entre um escravo A e B.

#### Envio

Byte	0	1	2	3	4
Dado	ID (LSB)	ID (MSB)	0xEF	ID (LSB) do rádio intermediário	ID (MSB) do rádio intermediário
5	6	N+6	N+7	N+8	N+9
Comando a ser repassado	PAYLOAD	CRC (LSB) do comando repassado	CRC (MSB) do comando repassado	CRC (LSB)	CRC (MSB)

#### Resposta

Byte	0	1	2	3	4	5
Dado	ID (LSB)	ID (MSB)	0xEF	SUBCOMANDO	ERRO (gravação)	Endereço sendo gravado (LSB)
6	7	8	9	10	11	12
Endereço sendo gravado (MSB)	Banco Desativado	ERRO (CRC)	0x55	0x55	CRC (LSB)	CRC (MSB)

- **PAYLOAD:** N bytes do comando a ser repassado.

## 9.15 Carrega FW (Loader) e Apaga banco (0xEB)

### 9.15.1 Carrega FW (subcomando 0x00)

#### Envio

Byte	0	1	2	3	4	5	6
Dado	ID (LSB)	ID (MSB)	0xEB	0x00	TIPO	TAMANHO	END (LSB)
7	8	9-14	15	16	17	18	
END (MSB)	DADO #1	...	CRC (LSB) Firmware	CRC (MSB) Firmware	CRC	CRC	

#### Resposta

Byte	0	1	2	3	4	5	6
Dado	ID (LSB)	ID (MSB)	0xEB	0x00	ERRO (Gravação)	Endereço sendo gravado (LSB)	Endereço sendo gravado (MSB)
7	8	9	10	11	12		
Banco Desativado	ERRO (CRC)	0x55	0x55	CRC (LSB)	CRC (MSB)		

### 9.15.2 Apaga banco (subcomando 0x01)

#### Envio

Byte	0	1	2	3	4	5	6
Dado	ID (LSB)	ID (MSB)	0xEB	0x01	TIPO	TAMANHO	END (LSB)
7	8	9-14	15	16	17	18	
END (MSB)	DADO #1	...	CRC FW (LSB)	CRC FW (MSB)	CRC	CRC	

#### Resposta

Byte	0	1	2	3	4	5	6	7	8
Dado	ID (LSB)	ID (MSB)	0xEB	0x05	ERRO (Gravação)	RSV	RSV	CRC (LSB)	CRC (MSB)

- **TIPO:**

- **0x00:** normal;
- **0x01:** segmento.

- **ERRO (Gravação):** indica se a gravação foi bem sucedida (0x00) ou não (>0x01);

- **ERRO (CRC):** indica se houve erro de CRC (0x02) ou não (0x00).

### 9.16 Ativa Banco (0xEA)

Função que ativa a atualização do firmware.

#### Envio

Byte	0	1	2	3	4	5	6	7
Dado	ID (LSB)	ID (MSB)	0xEA	0x00	CRC FW (LSB)	CRC FW (MSB)	CRC (LSB)	CRC (MSB)

#### Resposta

Byte	0	1	2	3	4	5	6	7	8	9
Dado	ID (LSB)	ID (MSB)	0xEA	Banco Ativado	Banco Desativado	ERRO (CRC)	0x55	0x55	CRC (LSB)	CRC (LSB)

- **ERRO (CRC):** indica se houve erro de CRC (0x02) ou não (0x00).

### 9.17 Apaga Banco (0xE9)

#### Envio

Byte	0	1	2	8	9
Dado	ID (LSB)	ID (MSB)	0xE9	CRC(LSB)	CRC(LSB)

#### Resposta

Byte	0	1	2	3	4	5	6	7	8
Dado	ID (LSB)	ID (MSB)	0xE9	0x05	ERRO	0x00	0x00	CRC (LSB)	CRC (MSB)

- **ERRO:** indica se a configuração foi bem sucedida (0x00) ou não (>0x00);



### 9.18 Canal Serial (0xD9)

Seta canal de operação do rádio pela interface serial.

#### Envio

Byte	0	1	2	3	5	6
Dado	ID (LSB)	ID (MSB)	0xD9	CANAL	CRC (LSB)	CRC (MSB)

#### Resposta

Byte	0	1	2	3	5	6
Dado	ID (LSB)	ID (MSB)	0xD9	CANAL	CRC (LSB)	CRC (MSB)

### 9.19 Ruído (0xD8)

Retorna o valor do ruído ouvido pelo rádio selecionado.

#### Envio

Byte	0	1	2	3	5
Dado	ID (LSB)	ID (MSB)	0xD8	CRC (LSB)	CRC (MSB)

#### Resposta

Byte	0	1	2	3	4	5	6	7
Dado	ID (LSB)	ID (MSB)	0xD8	RSSI (Mín.)	RSSI (Med.)	RSSI (Máx.)	CRC (LSB)	CRC (MSB)

## 9.20 Rota (0xD3)

Define a rota de comunicação estática com um determinado rádio.

### Envio

Byte	0	1	2	3	4	5	6
Dado	ID (LSB)	ID (MSB)	0xD3	SUBCOMANDO	N	ID1 (LSB)	ID1 (MSB)
7	8	10	5+N*2	6+N*2			
ID2 (LSB)	ID2 (MSB)	...	CRC (LSB)	CRC (MSB)			

### Resposta

Byte	0	1	2	3	4	5	6
Dado	ID (LSB)	ID (MSB)	0xD3	SUBCOMANDO	N	ID1 (LSB)	ID1 (MSB)
7	8	10	5+N*2	6+N*2			
ID2 (LSB)	ID2 (MSB)	...	CRC (LSB)	CRC (MSB)			

- **ID (LSB) e ID (MSB):** ID do rádio que está sendo roteado. Sempre manda par ao rádio mestre.
- **SUBCOMANDO:**
  - **0x01:** escreve rota;
  - **0x02:** consulta.
- **N:** número de saltos.
- **ID1 (LSB):** primeiro rádio (após o mestre).
- **ID2 (LSB):** N-ésimo rádio.

### 9.21 Grava senha (0xCD)

Configura a senha do rádio selecionado.

#### Envio

Byte	0	1	2	3	4-7	8-11	12	13
Dado	ID (LSB)	ID (MSB)	0xCD	SUBCOMANDO	Senha nova	Senha antiga	CRC (LSB)	CRC (MSB)

• **SUBCOMANDO:**

- **0x01:** sem trava senha;
- **0x02:** com trava senha;
- **0x03:** teste de senha;
- **0x04:** grava via RF sem trava senha.

#### Resposta

Byte	0	1	2	3	4-6	7	8
Dado	ID (LSB)	ID (MSB)	0xCD	SUBCOMANDO	Trava senha	CRC (LSB)	CRC (MSB)

• **SUBCOMANDO:**

- **0x00:** senha ok, sem trava.

## 9.22 ECO (0xA0)

A função deste comando é efetuar testes de rede com pacotes de dados grandes.  
O escravo retorna o mesmo pacote recebido automaticamente.

### Envio

Byte	0	1	2	3	N+4	N+5
Dado	ID (LSB)	ID (MSB)	0xA0	N-BYTES	CRC (LSB)	CRC (MSB)

### Resposta

Byte	0	1	2	3	N+4	N+5
Dado	ID (LSB)	ID (MSB)	0xA0	N-BYTES	CRC (LSB)	CRC (MSB)

### 9.23 Variáveis dinâmicas (0x88)

Configura as variáveis dinâmicas de funcionamento.

#### Envio

Byte	0	1	2	3-5	6	7
Dado	ID (LSB)	ID (MSB)	0x88	SUBCOMANDO	CRC (LSB)	CRC (MSB)

#### Resposta

Byte	0	1	2	3-5	6	7
Dado	ID (LSB)	ID (MSB)	0x88	SUBCOMANDO	CRC (LSB)	CRC (MSB)

#### • SUBCOMANDO:

- **0x01**: zera tabela de repetição;
- **0x02**: desliga Spread Spectrum (só para o Mestre);
- **0x03**: religa o Spread Spectrum (só para o Mestre);
- **0x04**: reset do rádio;
- **0x05**: vai para o canal de guarda e força readquirir sincronismo;
- **0x06**: liga modo ponto a ponto (só para Mestre; desliga automaticamente no reset);
- **0x07**: liga modo ponto a ponto (só para Mestre).

## 9.24 LEDs (0x83)

A função deste comando é acender os leds para teste em produção, porém, pode ser usado em qualquer momento.

O estado dos leds é uma combinação binária, ou seja, é necessário configurar os bits para acender os leds.

### Envio

Byte	0	1	2	3	4	5
Dado	ID (LSB)	ID (MSB)	0x83	ESTADO LEDES	CRC (LSB)	CRC (MSB)

### Envio

Byte	0	1	2	3	4	5
Dado	ID (LSB)	ID (MSB)	0x83	ESTADO LEDES	CRC (LSB)	CRC (MSB)

#### • ESTADO LEDES:

- **0x01**: LED TX;
- **0x02**: LED RX;
- **0x04**: LED ST.

## 9.25 Comando de aplicação (<0x80)

### 9.25.1 Configuração de SENHA (0xCD)

Comando para configuração de SENHA de um rádio. Ele é aceito apenas localmente pela serial.

#### Envio

Byte	0	1	2	3	4	5	6	
Dado	ID (LSB)	ID (MSB)	0xCD	0x04	SENHA (LSB)	SENHA	SENHA	
7	8	9						
SENHA (MSB)	CRC (LSB)	CRC(MSB)						

#### Resposta

Byte	0	1	2	3	4	5	6
Dado	ID (LSB)	ID (MSB)	0xCD	0x04	RSV	CRC (LSB)	CRC (MSB)

**SENHA** é um valor inteiro de até 9 dígitos separado em 4 bytes. Por exemplo, se a SENHA a ser configurada for 12345678 e o ID do rádio, 6, o comando a ser enviado pela serial será: 06 00 CD 04 4E 61 BC 00 4E A1.



### 9.25.2 Comando de aplicação (<0x80)

Comandos de aplicação são aqueles entre 0x02 e 0x7F (com estes inclusos). Uma vez recebidos pelo rádio, eles são enviados diretamente para a serial. No caso do comando configurado como interface transparente, o *payload* do pacote também é encaminhado para a interface transparente, conforme especificado na Seção 6. O tamanho máximo do *payload* é de 232 bytes

## Atenção

**Comandos de aplicação, diferentemente dos comandos de operação (comandos descritos acima), não possuem respostas automáticas. Assim, a aplicação ligada ao rádio deve necessariamente enviar as respostas para todos os comandos recebidos (com exceção do comando 0x01 ou do 0x02 até 0x0A quando configurada a resposta automática).**

## 10 Características Físicas

### 10.1 Dimensões

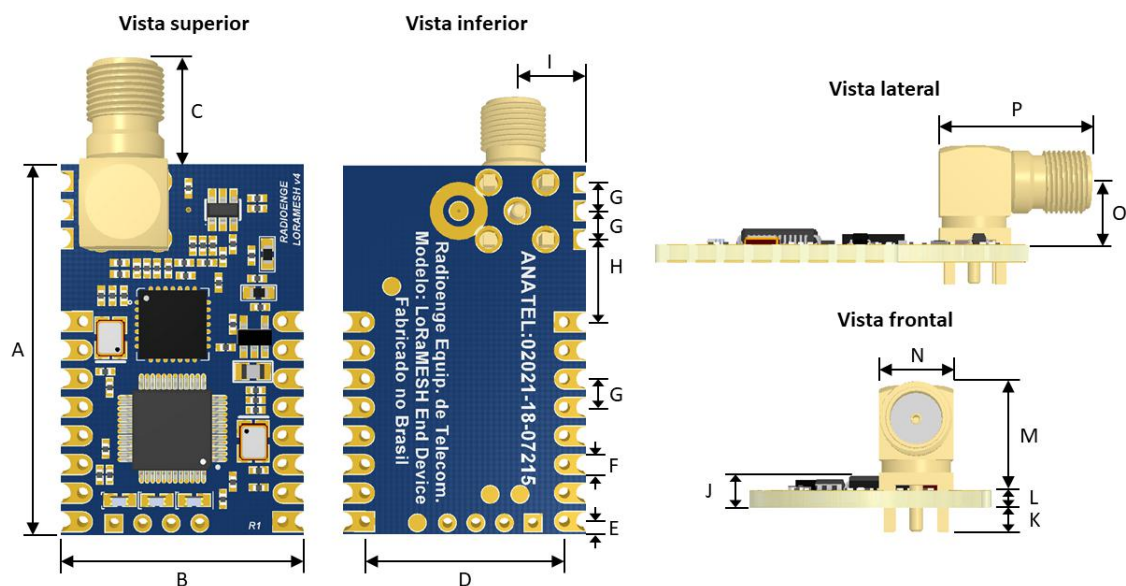


Figura 6: Vistas e dimensões do módulo

#### Dimensões do Módulo

Cota	Tamanho [mm]	Cota	Tamanho [mm]
A	32,9	I	6,10
B	21,6	J	3,24
C	13,0	K	1,98
D	17,8	L	1,62
E	1,15	M	9,80
F	1,78	N	6,00
G	2,54	O	6,58
H	7,37	P	20,0

## 10.2 Footprint recomendado

### 10.2.1 SMD

O **footprint** apresentado abaixo pode ser obtido na página do EndDevice Radioenge no formato compatível com o Altium.

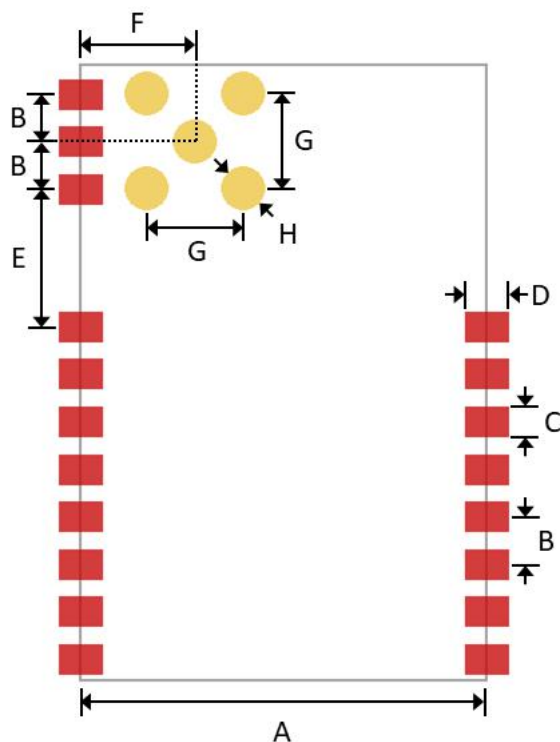


Figura 7: *Footprint* recomendado e dimensões

#### Dimensões do Módulo

Cota	Tamanho [mm]	Cota	Tamanho [mm]
A	21,6	E	7,37
B	2,54	F	6,10
C	1,52	G	5,08
D	2,29	H	2,40

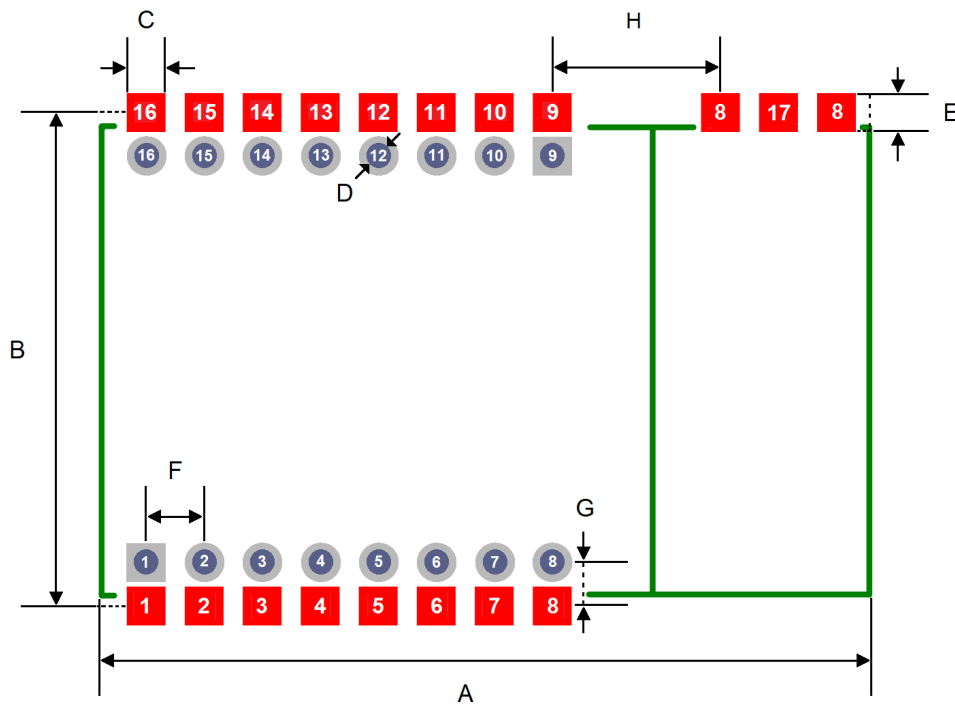
**10.2.2 PTH**


Figura 8: Footprint para utilizar os pinos

Tabela 16: Dimensões do Módulo

Cota	Tamanho [mm]	Cota	Tamanho [mm]
A	33,9	E	1,70
B	21,6	F	2,54
C	1,70	G	1,90
D	1,00	H	7,37

## 11 CRC 16

### 11.1 Linguagem C

Os dois bytes de CRC são calculados sobre todos os demais bytes através do algoritmo em C apresentado no quadro abaixo e são dispostos na forma Little-Endian (LSB:MSB).

```
/** * @brief Calcula CRC16.
 * @param data_in: Ponteiro para o buffer contendo os dados.
 * @param length: Tamanho do buffer
 * @retval Valor de 16 bits representando o CRC16 do buffer fornecido. */

#define CRC_POLY (0xA001)
uint16_t CalculaCRC(uint8_t* data_in, uint32_t length)
{
    uint32_t i;
    uint8_t bitbang, j;
    uint16_t crc_calc;

    crc_calc = 0xC181;
    for(i=0; i<length; i++)
    {
        crc_calc ^= ((uint16_t)data_in[i]) & 0x00FF;

        for(j=0; j<8; j++)
        {
            bitbang = crc_calc;
            crc_calc >>= 1;

            if(bitbang & 1)
            {
                crc_calc ^= CRC_POLY;
            }
        }
    }
    return crc_calc;
}
```

## 11.2 Python

```
def crc(buffer, tamanho):
    bitbang = 0
    crc_calc = semente_crc
    for x in range(0, tamanho):
        crc_calc ^= buffer[x] & 0x00FF
        for j in range(0, 8):
            bitbang=crc_calc
            crc_calc>>=1
            if(int(bitbang) & 1):
                crc_calc ^= polinomio_crc16
    return crc_calc
```

Uso da função de cálculo do CRC16.

```
#Calcula o verificador CRC16
ID = 254
id_lsb=int(ID)%256
id_msb=int(ID)/256
byte_funcao = 256
envio=bytearray([int(id_lsb), int(id_msb), byte_funcao, 0, 0, 0, 0, 0, 0, 0])
crc_calc = crc((envio), len(envio)-2)
crc_lsb = crc_calc%256 #separa o CRC calculado em byte LSB
#print(crc_lsb)
crc_msb = int(crc_calc / 256) & 0xFF #separa o CRC calculado em byte MSB
```

### 11.3 Java / Android

```
public class CRC {
    private static final int CRC_POLY = 0xA001; // polinômio utilizado para o cálculo
    public static int CalculaCRC(byte [] b, int tam)
    {
        int bitbang,i,j;
        int CRC_calc;

        CRC_calc = 0xC181;


        for(i=0;i<tam;i++)
        {
            CRC_calc ^= ((int) (b[i])) & 0x00FF;
            for(j=0;j<8;j++)
            {
                bitbang = CRC_calc;
                CRC_calc >>= 1;
                int c = bitbang & 0x01;
                if((c & 0x00FF) ==1)
                {
                    CRC_calc ^= CRC_POLY;
                }
            }
        }
        return CRC_calc;
    }
}
```

#### Uso da função de cálculo do CRC-16

```
arrayCmd = new byte[] (byte) 0xFE, (byte) 0x00, (byte) 0x88, (byte) 0x04, (byte) 0x00,
(byte) 0x00, (byte) 0x15, (byte) 0xB8;
int id = Integer.parseInt(idRadioET.getText().toString()) %256;
arrayCmd[0]= (byte) id;
id = Integer.parseInt(idRadioET.getText().toString()) / 256;
arrayCmd[1]= (byte) id;
arrayCmd = CRC.Calcula.Add(arrayCmd, arrayCmd.length - 2);
```

## 12 Contato

- **WhatsApp:**

 +55 (41) 3052-9444

- **Site:** <https://www.radioenge.com.br/contato/>