Radioenge

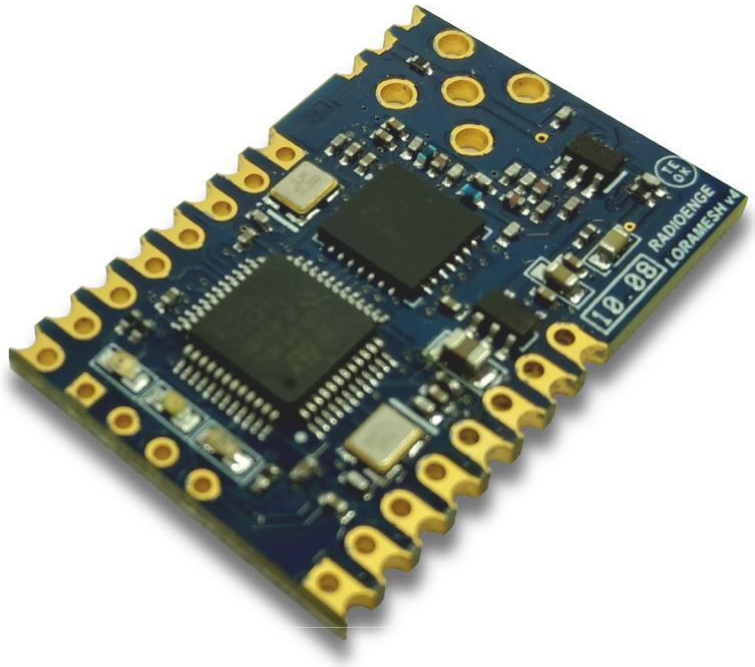
Módulo LoRaMESH

Manual de Utilização

Revisão - fevereiro 2020

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 EndDevice possui duas interfaces UART para a configuração de operação e transmissão 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 1,8 a 12 Vcc (+3,3 Vcc sem regulador)
- Dimensões: 33 x 22 x 3 mm
- Temperatura de operação: -5°C a +55°C
- Baixo consumo
- µProcessador 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

Sumário

1. Tecnologia LoRaMESH

LoRaMESH é o sistema de roteamento em malha (mesh) e comunicação sem fio com auto-rota e auto-cura desenvolvido pela Radioenge. A comunicação flui entre os escravos e o mestre da rede através de N-saltos, ou seja, passando por vários es-cravos de forma automática por meio do roteamento MESH Radioenge. A topologia da rede é mostrada na figura abaixo.

O LoRaMESH EndDevice Radioenge implementa a camada física LoRa® ou FSK e a comunicação entre escravos e mestre é feita de forma transparente para a aplicação, facilitando a criação da rede sem fio em projetos de IoT e proporcionando alta escalabilidade, uma vez que novos dispositivos são facilmente incorporados à rede.

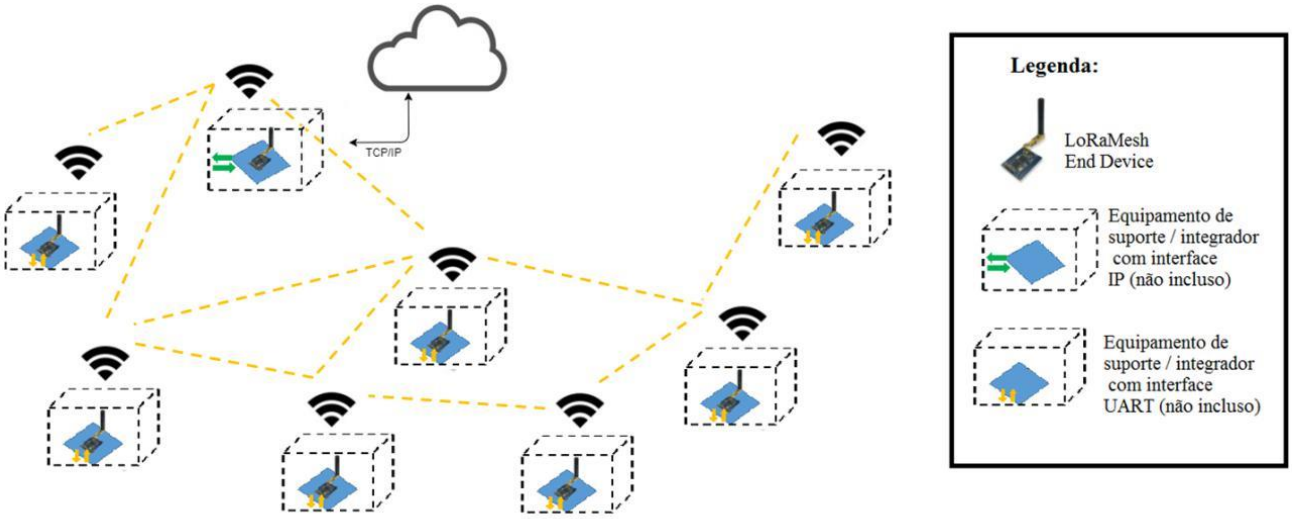


Figura 1 - Diagrama de uma rede Mesh

2. Pinagem

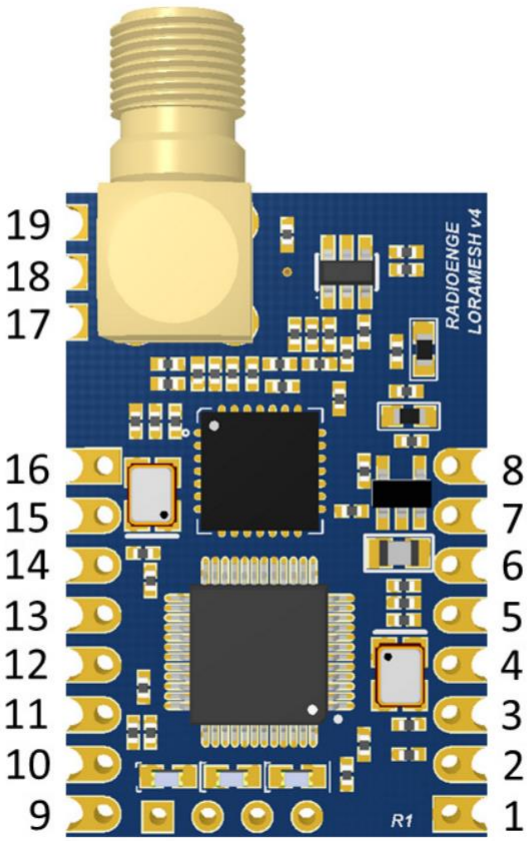


Figura 2 - Numeração dos pinos

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	Alimentação	Conectado ao ground
18	ANT	Saída RF	Saída de RF para antena externa
19	GND	Alimentação	Conectado ao ground

3. Especificações

3.1. Especificações 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	LoRa/MESH 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

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	-0,3	4,0	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	ºC
Temperatura de operação	-5	+55	ºC
Potência máxima na entrada RF	-	0	dBm

3.3. Parâmetros de operação

Parâmetro	Mín.	Normal	Máx.	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, Io = 8 mA)	-	-	0,4	V
Tensão de saída em nível alto (GPIO, Io = 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

3. Especificações

3.3. Parâmetros de operação (continuação)

Parâmetro	Mín.	Normal	Máx.	Un.
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

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

Coding Rate 4/8

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

Coding Rate 4/7

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

Coding Rate 4/6

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

Coding Rate 4/5

4. Exemplo de Aplicação

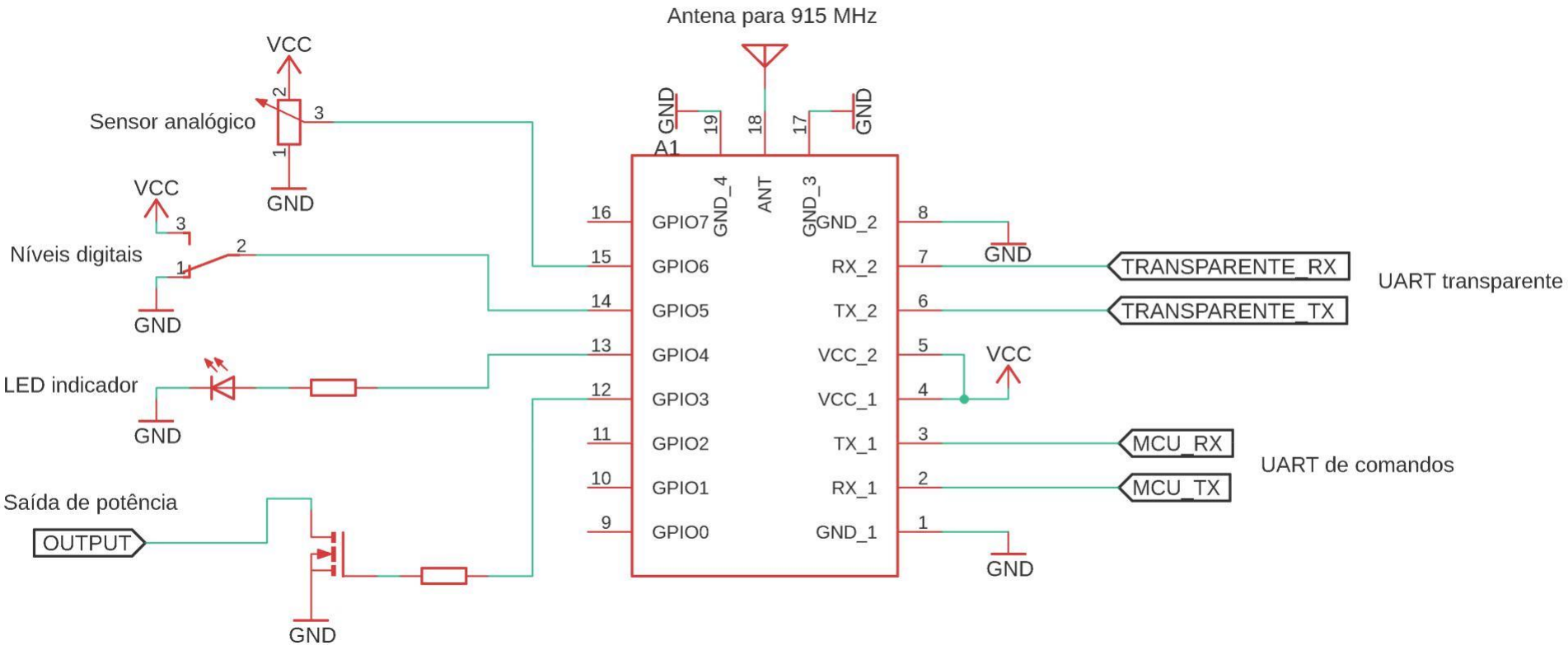


Figura 3. Exemplo de aplicação do EndDevice Radioenge

5. Descrição das Interfaces

5.1. Interface de comandos

O módulo possui uma interface serial para configuração dos parâmetros de operação e envio de pacotes. O baudrate é configurável entre **9600**, **38400** e **57600** bps, sendo o padrão de fábrica **9600** bps. A tabela 1 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 6 – Descrição do protoco-lo serial.

Tabela 1. Parâmetros da interface serial

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

5.2. Interface transparente

Além da interface de configuração, o EndDevice possui uma interface serial que pode ser usada para o envio e recebimento de pacotes de dados de forma trans-parente ao dispositivo.

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.

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

Comunicando via serial de um escravo, apenas o campo MENSAGEM é envia-do/recebido.

Os parâmetros desta interface são iguais a interface de comandos e o *baudra-te* é igualmente configurável (Tabela 1).

5.3. GPIOs

Este dispositivo possui 8 pinos de entrada/saída digitais de uso geral, dos quais dois (GPIO1 e GPIO2) podem ser configurados como entradas analógicas. A função destes pinos (entrada/saída) e o estado (alto/baixo) são configuráveis tanto pela interface serial de comandos como via rádio, por meio dos mesmos comandos.

5. Descrição das Interfaces

5.4. Entradas analógicas

O EndDevice Radioenge oferece até 2 entradas analógicas, multiplexadas com duas GPIOs, com resolução de 12 bits.

A configuração e leitura das entradas pode ser feita tanto via rádio quanto via serial de comandos.

5.5. Saída de antena

O dispositivo conta com duas possibilidades de conexão de antena:

- Conector SMA-M, para a conexão direta de uma antena ao módulo;
- Pad acastelado (Pino 18), para a conexão com uma PCI base. Neste caso de-ve-se manter a impedância controlada de 50 Ω na placa base.

Deve-se usar apenas um modo de conexão. Caso use o conector SMA, deve-se dei-xar o pino 18 desconectado. Em caso de usar o pino 18, deve-se remover o conector SMA.

5.6. Bypass do regulador

Caso não seja necessário o uso do regulador, quando a alimentação é de 3,3 Vcc, pode-se realizar o *bypass* deste regulador adicionando um resistor de 0 Ω (0402) na posição indicada na figura abaixo.

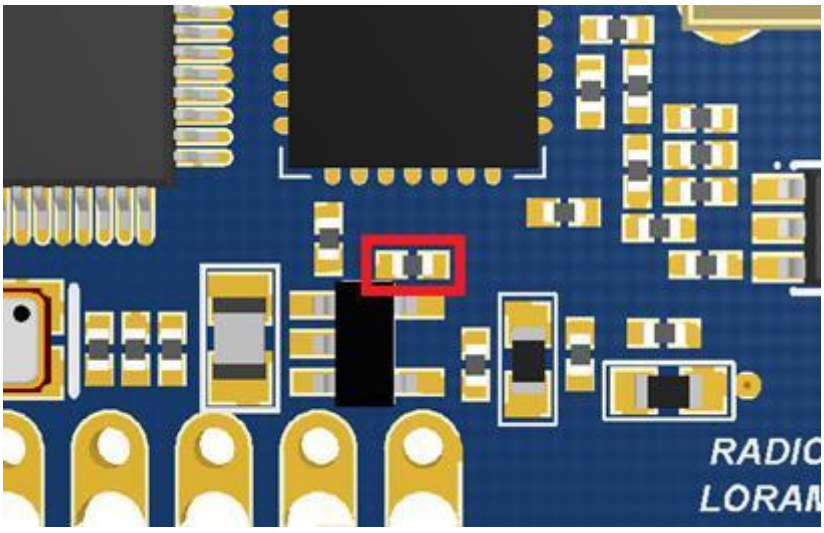


Figura 3. Posição do resistor de *bypass*

O uso do *bypass* é indicado para aplicações alimentadas a bateria, onde é desejável o mínimo consumo possível.

5. Descrição das Interfaces

5.7. 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 como forma de indicar o correto funcio-namento.

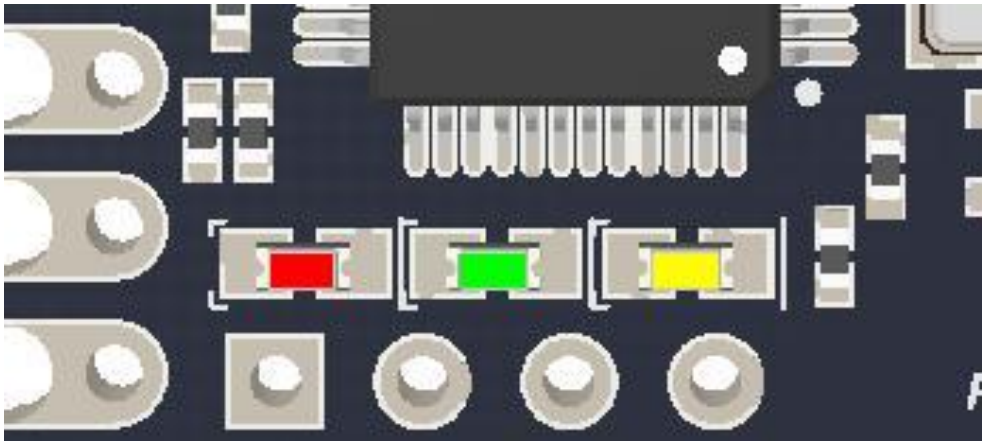


Figura 4. LEDs de sinalização

6. Descrição do Protocolo Serial

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

ID (LSB)	ID (MSB)	Função	DadoN	...	Dado0	CRC (LSB)	CRC (MSB)

ID indica o identificador único do dispositivo dentro de uma rede para o qual o comando é endereçado. **Função** especifica o comando enviado ao EndDevice, conforme a tabela **X**. Os campos **Dados** representam os parâmetros a serem enviados ou dados de aplicação.

6.1. CRC-16 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 (CRCL:CRCH).

Qualquer comando, com exceção do Local Read, pode ser executado tanto via interface serial de comandos quanto via rádio (remotamente).

```
/**
* @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;
```

}

6. Descrição do Protoco-lo Serial

6.2 – CRC-16 – 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 = 212
envio=bytearray([int(id_lsb),int(id_msb), byte_funcao, 0, 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
```

6. Descrição do Protoco-lo Serial

6.3. CRC-16 – 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);
```

6.4. Lista de comandos seriais

Comando	Descrição
0xD6	Leitura / Escrita de parâmetros da modulação
0xE2	Local Read – Leitura dos parâmetros do módulo conectado via UART
0xD4	Remote Read – Leitura dos parâmetros de qualquer módulo na rede
0xCA	Write Config – Escrita dos parâmetros da rede no dispositivo (NET e ID)
0xC2	Comando I/O – Configura, lê e envia o estado dos pinos de GPIO
0xE7	Diagnóstico – Adquire informações de operação do nó especificado
0xD8	Ruído – Lê o nível de ruído somente no canal atual
0xD5	RSSI – Lê as potências dos sinais entre o nó indicado e o nó vizinho
0xD2	Trace Route – Traça a rota necessária para o mestre comunicar com um nó específico
0x28	Envia um pacote de dados para a interface serial transparente do destino
< 0x80	Qualquer comando neste intervalo é enviado integralmente à interface de comandos do destino

6.5. 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.

6.5.1. Leitura de parâmetros da modulação (0xD6)

Envia uma requisição de leitura do tipo de modulação e dos parâmetros LoRa para o ID especificado.

Envio

ID_L	ID_H	0xD6	0x00	0x01	0x00	CRC16
Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Bytes 6-7

Resposta

ID_L	ID_H	0xD6	0x00	Potência	Largura de banda	*Spreading factor*	*C o d i n g rate*	CRC16
Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7	Bytes 8-9

A **potência** é especificada em dBm. A **largura de banda** pode ser 0 (125 kHz), 1 (250 kHz) ou 2 (500 kHz). ***Spreading Factor*** pode ter um valor de 7 a 12, para operação com modulação LoRa, ou 5, que indica operação em modo FSK. ***Coding Rate*** pode ter valores de 1 a 4, representando *coding rates* de 4/5 a 4/8, respectivamente.

6. Descrição do Protoco-lo Serial

6.5.2. Escrita de parâmetros da modulação (0xD6)

Envia o tipo de modulação e dos parâmetros LoRa para o ID especificado.

Envio

ID_L	ID_H	0xD6	0x01	Potência	Largura de banda	*Spreading factor*	*Coding rate*	CRC16
Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7	Bytes 8-9

Resposta

ID_L	ID_H	0xD6	0x01	Potência	Largura de banda	*Spreading factor*	*Coding rate*	CRC16
Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7	Bytes 8-9

6.5.3. Local read (0xE2)

Envia uma requisição dos parâmetros de rede do dispositivo local, como ID, NET e Unique ID, além de outros parâmetros de operação do dispositivo.

Envio

ID_L	ID_H	0xE2	0x00	CRC16
Byte 0	Byte 1	Byte 2	Bytes 3-5	Bytes 6-7

Resposta

ID_L	ID_H	0xE2	NET_L	NET_H	UID0	UID1	UID2	UID3	Versão de HW	RSD
Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7	Byte 8	Byte 9	Byte 10
Rev. de FW	RSD	*Spread Spectrum*	RSD	Versão de FW	Banco Mem.	RSD	0x01	RSD	CRC16	
Byte 11	Bytes 12-14	Byte 15	Byte 16	Byte 17	Byte 18	Bytes 19-25	Bytes 26-27	Byte 28	Bytes 29-30	

NET é o identificador de uma rede e pode assumir valores entre 0 e 2047. Todos os rádios de uma mesma rede devem estar configurados com a mesma NET. Redes diferentes podem ocupar o mesmo espaço físico, desde que tenham valores de NET diferentes. O campo **UID** é o identificador único (Unique ID) do dispositivo e cada módulo produzido possui um Unique ID diferente. **Versão de HW**, V**ersão de FW** e **Revisão de FW** são informações relativas às versões de *hardware* e *firmware* do módulo. ***Spread Spectrum*** indica se a modulação ativa é a FSK (0x01) ou a LoRa (0x00). **Banco de Memória** pode ser 0 ou 1 e indica qual banco de memória está ativo.

6.5.4. Remote read (0xD4)

De forma similar ao Local Read, envia uma requisição dos parâmetros de rede a um dispositivo remoto.

Envio

ID_L	ID_H	0xD4	0x00	CRC16
Byte 0	Byte 1	Byte 2	Bytes 3-5	Bytes 6-7

Resposta

ID_L	ID_H	0xD4	NET_L	NET_H	UID0	UID1	UID2	UID3	Versão de HW	RSD
Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7	Byte 8	Byte 9	Byte 10
Rev. de FW	RSD	*Spread Spectrum*	RSD	Versão de FW	Banco Mem.	RSD	0x01	RSD	CRC16	
Byte 11	Bytes 12-14	Byte 15	Byte 16	Byte 17	Byte 18	Bytes 19-25	Bytes 26-27	Byte 28	Bytes 29-30	

6.5.5. Write Config (0xCA)

Envia configurações de ID e NET para o dispositivo local ou remoto. Para que o dispositivo aceite as alterações, deve-se informar seu Unique ID, obtido através do comando Local Read ou Remote Read.

Envio

ID_L	ID_H	0xCA	NET_L	NET_H	UID0
Byte 0	Byte 1	Byte 2	Bytes 3	Byte 4	Byte 5
UID1	UID2	UID3	0x00	CRC16	
Byte 6	Byte 7	Byte 8	Bytes 9-13	Bytes 14-15	

Resposta

ID_L	ID_H	0xD4	NET_L	NET_H	UID0	UID1	UID2	UID3	Versão de HW	RSD
Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7	Byte 8	Byte 9	Byte 10
Rev. de FW	RSD	*Spread Spectrum*	RSD	Versão de FW	Banco Mem.	RSD	0x01	RSD	CRC16	
Byte 11	Bytes 12-14	Byte 15	Byte 16	Byte 17	Byte 18	Bytes 19-25	Bytes 26-27	Byte 28	Bytes 29-30	

6.5.6. Comando I/O (0xC2)

Permite a configuração do modo de operação de cada GPIO, como entrada digital, saída digital ou entrada analógica (apenas GPIO1 e GPIO2) e realiza a leitura ou escrita nas GPIOs.

Este comando possui subcomandos para cada função específica – configura-ção, leitura e escrita.

6.5.6.1. Subcomando de configuração (0x02)

Para uma GPIO dada pelo campo **Pino**, pode-se configurar seu modo de ope-ração e habilitar/desabilitar o *pull-up* ou *pull-down* interno. **Pino** pode assumir valores entre 0 e 7, relativos aos GPIO0 a GPIO7. **PULL** pode ser 0 (*pull-up* e *pull-down* de-sabilitados), 1 (*pull-up* habilitado) ou 2 (*pull-down* habilitado). **Input/Output** pode ser 0 (entrada digital), 1 (saída digital) ou 3 (entrada analógica).

Envio

ID_L	ID_H	0xC2	0x02	Pino	PULL	Input/Output	CRC16
Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Bytes 7-8

Resposta

ID_L	ID_H	0xC2	0x02	Erro	Pino	Payload	CRC16
Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Bytes 7-8

O campo **Erro** indica que houve erro na execução do comando caso seja 0x01, em caso de sucesso seu valor será 0x00. O campo ***Payload*** traz as configurações gravadas e é subdividido em bits da seguinte forma:

Payload

X	PULL	X	Input/Output
Bits 15-10	Bits 9-8	Bits 7-2	Bits 1-0

6.5.6.2. Subcomando de leitura (0x00)

Realiza a leitura de uma GPIO dada pelo campo **Pino**, seja ela entrada digital ou analógica. O modo de operação da GPIO é indicado na resposta.

Envio

ID_L	ID_H	0xC2	0x00	Pino	0x00	CRC16
Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Bytes 5- 6	Bytes 7-8

Resposta

ID_L	ID_H	0xC2	0x00	Erro	Pino	Payload	CRC16
Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Bytes 7-8

O campo **Erro** indica que houve erro na execução do comando caso seja 0x01, em caso de sucesso seu valor será 0x00. O campo ***Payload*** traz o resultado da lei-tura do pino especificado.

Payload

An/Dig	X	Leitura
Bit 15	Bits 14-12	Bits 11-0

O campo **An/Dig** indica se a leitura é digital (1) ou analógica (0). O campo **Leitura** possui o valor de 12 bits, no caso de uma leitura analógica. O bit 0 do campo **Leitura** indica o nível lógico do pino, no caso de uma leitura digital.

6.5.6.3. Subcomando de Escrita (0x01)

Envia o nível lógico desejado para um pino configurado como saída.

Envio

ID_L	ID_H	0xC2	0x01	Pino	Alto/Baixo	0x00	CRC16
Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Bytes 7-8

Resposta

ID_L	ID_H	0xC2	0x01	Erro	Pino	Payload	CRC16
Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Bytes 7-8

O campo **Erro** indica que houve erro na execução do comando caso seja 0x01, em caso de sucesso seu valor será 0x00. O bit 0 do campo ***Payload*** traz o nível lógi-co enviado.

6.5.7. Diagnóstico (0xE7)

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

Envio

ID_L	ID_H	0xE7	0x00	CRC16
Byte 0	Byte 1	Byte 2	Bytes 3-5	Bytes 6-7

Resposta

ID_L	ID_H	0xE7	Temp. Mín.	Temp. Atual	Temp. Máx.	Tensão Mín.	Tensão Atual	Tensão Máx.	RSD
Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7	Byte 8	Bytes 9-14
0x01	RSD	*0x0C*	RSD	0x00	0x55	RSD	0x00	CRC16	
Byte 15	Bytes 16-26	Byte 27	Bytes 28-35	Bytes 36-43	Byte 44	Bytes 19-25	Bytes 26-27	Byte 28	

As temperaturas são dadas em ºC com valores inteiros. As tensões são dadas em Vcc x 10. Por exemplo, 3,3 V é representado pelo valor inteiro 33.

6.5.8. Ruído (0xD8)

De forma semelhante ao comando Espectro, envia uma requisição de leitura dos níveis de ruído mínimo, médio e máximo do canal atual.

Envio

ID_L	ID_H	0xD8	0x00	CRC16
Byte 0	Byte 1	Byte 2	Bytes 3-5	Bytes 6-7

Resposta

ID_L	ID_H	0xD8	Ruído Mín.	Ruído Méd.	Ruído Máx.	CRC16
Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Bytes 6-7

6.5.9. RSSI (0xD5)

Através do comando de RSSI verifica-se a potência do sinal recebido por um determinado rádio e a potência do sinal deste rádio recebido por um outro rádio da rede, em dBm. Uma diferença significativa entre o nível de RSSI na ida e na volta pode significar um problema no rádio. Um nível de sinal muito próximo do nível de ruído indicado pelo comando Espectro significa um sinal fraco. Como normalmente existem vários rádios na rede, o campo Gateway indica o ID do rádio que recebeu o sinal diretamente e mediu o RSSI na volta.

Envio

ID_L	ID_H	0xD5	0x00	CRC16
Byte 0	Byte 1	Byte 2	Bytes 3-5	Bytes 6-7

Resposta

ID_L	ID_H	0xD5	Gateway L	Gateway H	RSSI ida	RSSI volta	CRC16
Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Bytes 7-8

6.5.10. Trace Route (0xD2)

Envia uma requisição para traçar a rota de saltos entre o mestre e o nó especificado pelo campo ID. O resultado é uma sequência de IDs dos nós pelos quais o comando foi repetido até alcançar o destino.

O tamanho da resposta depende da quantidade de saltos necessários para alcançar o dispositivo de destino.

Envio

ID_L	ID_H	0xD2	0x00	CRC16
Byte 0	Byte 1	Byte 2	Bytes 3-5	Bytes 6-7

Resposta

ID_L	ID_H	0xD2	ID1_L	ID1_H	...	IDN_L	IDN_H	CRC16
Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	...	Byte n-3	Byte n-2	Bytes n-1 - n

6.5.11. Pacote de aplicação (<0x80)

Comandos menores que 0x80, exceto o 0x28, que é enviado para a UART transparente, são tratados como comandos de aplicação e são passados integralmente para a interface UART de comandos do destino. O *payload* deve ser menor ou igual a **232** bytes. Caso seja necessário enviar um pacote maior, a aplicação deve quebrar o pacote em blocos menores, confirmando o recebimento de cada bloco a fim de transferir o pacote completo e manter sua integridade. A definição da função de cada comando fica a cargo da aplicação. O EndDevice LoRaMESH apenas veri-fica a integridade do pacote via CRC e, em caso de erro, não repassa o comando à frente.

Atenção

Mensagens que não possuem resposta inerente devem ser tratadas pela aplicação no destino.

A aplicação DEVE enviar respostas a toda mensagem de aplicação recebida, tanto para garantir a entrega de tais mensagens quanto para garantir a integridade do roteamento da rede.

Envio

ID_L	ID_H	0x01 - 0x7F	Dado0	...	DadoN	CRC16
Byte 0	Byte 1	Byte 2	Byte 3	...	Byte n-2	Bytes n-1 - n

6.5.11.1. Envia para interface transparente (0x28)

Atua de forma semelhante ao comando de aplicação, porém envia apenas o *payload* da mensagem e a ID de origem para a interface UART transparente do dispositivo de destino. Além disso, envia a mesma mensagem para a interface de comandos, porém encapsulada no formato do pacote de envio abaixo. O *payload* máximo deste comando é de **232** bytes.

Envio

ID_L	ID_H	0x28	Dado0	...	DadoN	CRC16
Byte 0	Byte 1	Byte 2	Byte 3	...	Byte n-2	Bytes n-1 - n

Da mesma forma, este comando não possui uma mensagem de resposta inerente. Deve-se então tomar as medidas expostas no quadro do item 10.

6.5.11.2. Exemplo de comando de aplicação

Seja uma aplicação que deve efetuar leituras de sensores e ligar/desligar relés, ambos conectados a um ou mais dispositivos remotos. Pode-se definir um comando de aplicação para cada função necessária, como por exemplo “Requisita leitura”, “Muda estado do relé” e “Acknowledge”. Como já foi citado no item 11, é necessário que haja uma resposta a toda mensagem enviada. O comando “Acknowledge” tem como função confirmar para o mestre se o comando “Muda estado do relé” foi en-tregue e executado de forma correta ou não, já que este comando naturalmente não possui resposta.

Primeiramente, deve-se definir os comandos com algum valor entre 0x01 e 0x7F. Neste caso, os comandos escolhidos foram 0x10, para “Requisita leitura”, 0x11, para “Muda estado do relé”, e 0x01, para “Acknowledge”.

Por fim, define-se os pacotes da seguinte forma:

Envio - Requisita Leitura

ID_L	ID_H	0x10	CRC16
Byte 0	Byte 1	Byte 2	Bytes 3-4

Resposta - Requisita Leitura

ID_L	ID_H	0x10	Leitura do Sensor	CRC16
Byte 0	Byte 1	Byte 2	Byte 3	Bytes 4-5

Envio - Muda estado do relé

ID_L	ID_H	0x11	ON=1;OFF=0	CRC16
Byte 0	Byte 1	Byte 2	Byte 3	Bytes 4-5

Resposta - Acknowledge

ID_L	ID_H	0x01	Ok = 1; Erro = 0	CRC16
Byte 0	Byte 1	Byte 2	Byte 3	Bytes 4-5

6. Descrição do Protoco-lo Serial

Fica a cargo da aplicação gerar estes pacotes, incluindo o cálculo e verificação do CRC16, além de tratar os campos de comando e *payload* de acordo com o proto-colo de aplicação.

ID Broadcast

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. Por exemplo, pode-se enviar um comando para acionar uma GPIO de todos os módulos da rede endereçando o comando para o ID 2047.

Os comandos descritos na seção anterior podem ser enviados pelo *software* da Radioenge *RD915 Mesh*. Para isso, deve-se utilizar uma base para a comunicação com um computador, como o *IOT RS232* ou o *IOT USB*, ambos produtos Radio-enge.

Primeiramente deve-se identificar no “Gerenciador de dispositivos” a porta COM relativa à base IOT conectada. No caso do *IOT USB* haverá duas portas COM, sendo uma relativa à interface de comandos e a outra à interface transparente.

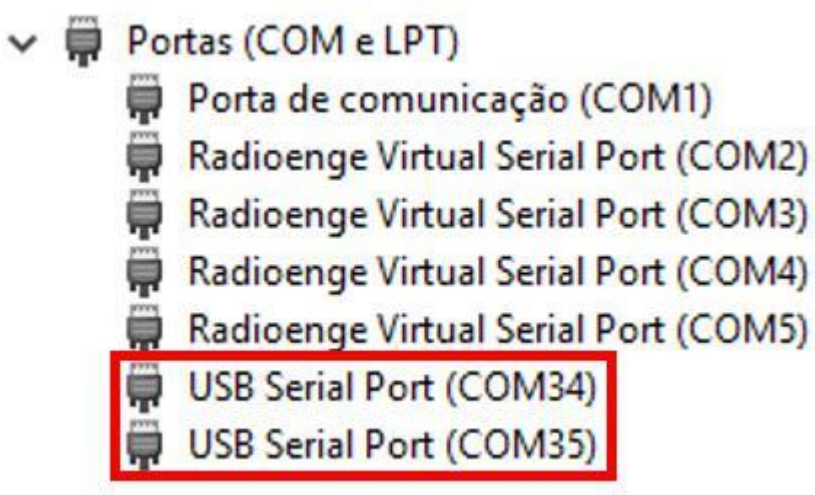


Figura 5. Identificação da porta COM

A porta COM com menor número é sempre a interface de comandos, já a maior é sempre a interface transparente. Como se deseja configurar o dispositivo, deve-se utilizar a porta COM34, neste caso.

Depois de identificar o número da porta COM, execute o software “RD915”. Na aba **CONN**, selecione a porta citada anteriormente, configure o *baudrate* corretamente (neste caso, 9600 bps) e clique em **Terminal** para abrir a interface com o dispositivo.

7. Tutorial de Configuração

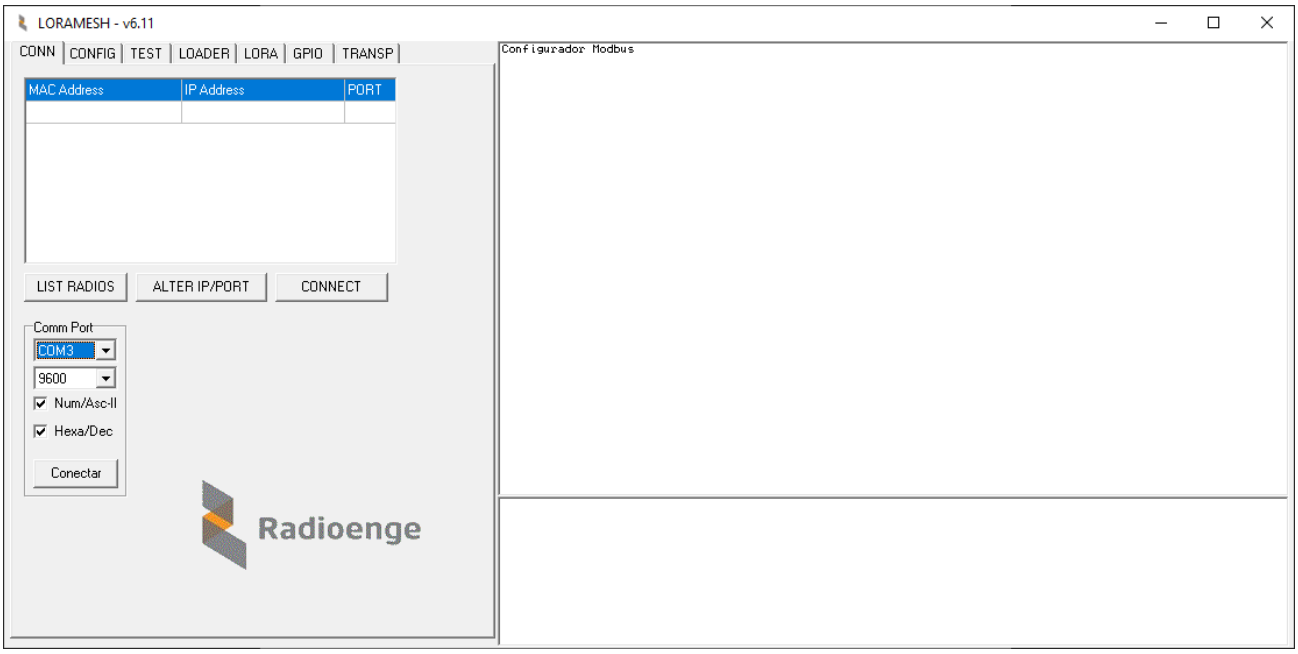


Figura 6. Iniciando a comunicação com o dispositivo

Na aba **CONFIG**, utilize o comando **LOCAL READ** para ler as informações do dispositivo conectado. O *software* irá preencher automaticamente o espaço **UNIQUE ID**. Para configurar o rádio, preencha o campo **Senha** com o valor desejado e o campo **ID** com o identificador desejado para este dispositivo. Por fim, pressione o botão **WRITE CONFIG**.

Atenção

O campo SENHA é o identificador de uma rede. Diferentes redes podem operar em um mesmo espaço físico com Senhas diferentes. Todos os pacotes comunicados entre os nós são criptografados a partir do número da Senha. Ainda assim, recomenda-se que a própria aplicação criptografe os pacotes antes de enviá-los aos dispositivos de rádio, de forma a garantir maior segurança.

7. Tutorial de Configu-ração

Figura 7. Aba CONFIG e comando de LOCAL READ

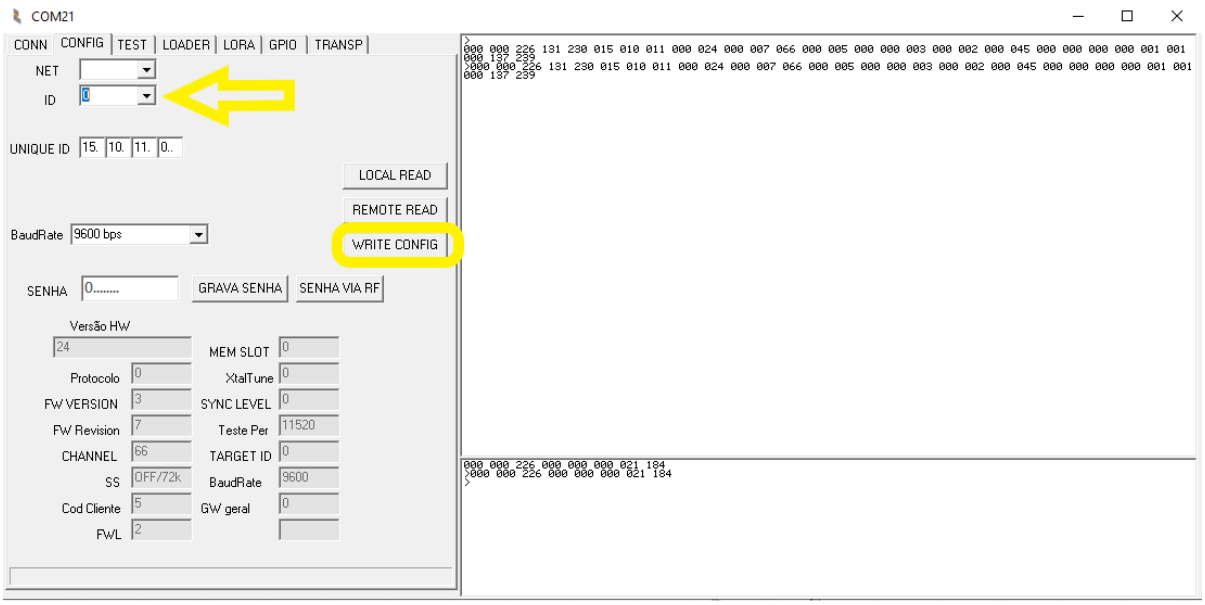


Figura 8. WRITE CONFIG – Configurando o ID do rádio

7. Tutorial de Configuração

Na aba **TEST** é possível realizar alguns testes de funcionamento, comunicando-se com os rádios da rede, somente se o rádio conectado ao computador seja o ID 0, ou seja, o rádio mestre. O teste **PING** é equivalente ao comando **REMOTE READ** da aba **CONFIG**. Através dele, o rádio mestre requisita as informações do rádio cujo **ID** foi selecionado. No exemplo da figura 9, o comando de leitura remota foi enviado ao rádio de ID 1.

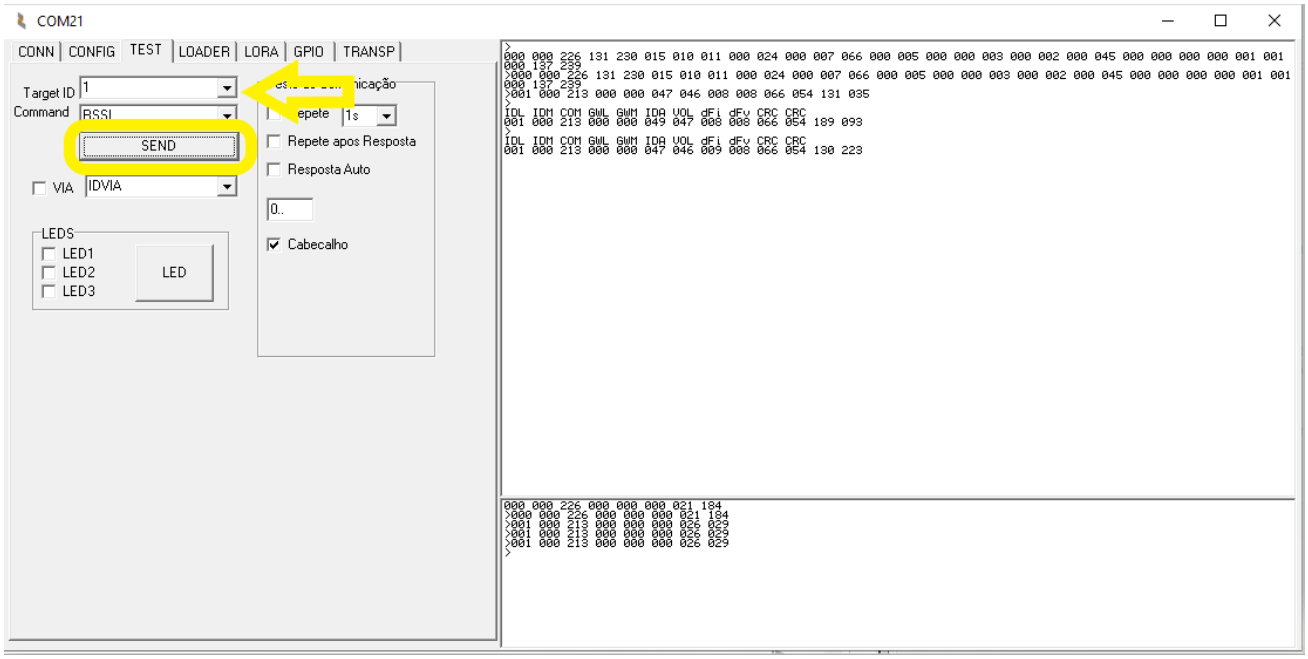


Figura 9. Teste PING via aba TEST

É possível ainda reconfigurar o **ID** dos rádios da rede remotamente, através do rádio mestre. Para isso, digite o **ID** atual do rádio na rede e faça uma leitura remota com o fim de ler o **UNIQUE ID**. Em seguida, basta digitar o novo **ID** e executar o comando **WRITE CONFIG**, da mesma forma que foi executado no dispositivo local.

Para configurar os parâmetros de modulação LoRa, clique na aba **LORA** e selecione o **ID** do dispositivo desejado. Nesta aba pode-se configurar a potência de transmissão, em dBm, a largura de banda, em kHz, o *spreading factor* e o *coding rate*. Ao configurar uma rede, é necessário que todos os dispositivos desta rede operem com exatamente os mesmos parâmetros de modulação LoRa, com exceção da potência. Caso contrário, os dispositivos não serão capazes de se comunicar.

7. Tutorial de Configuração

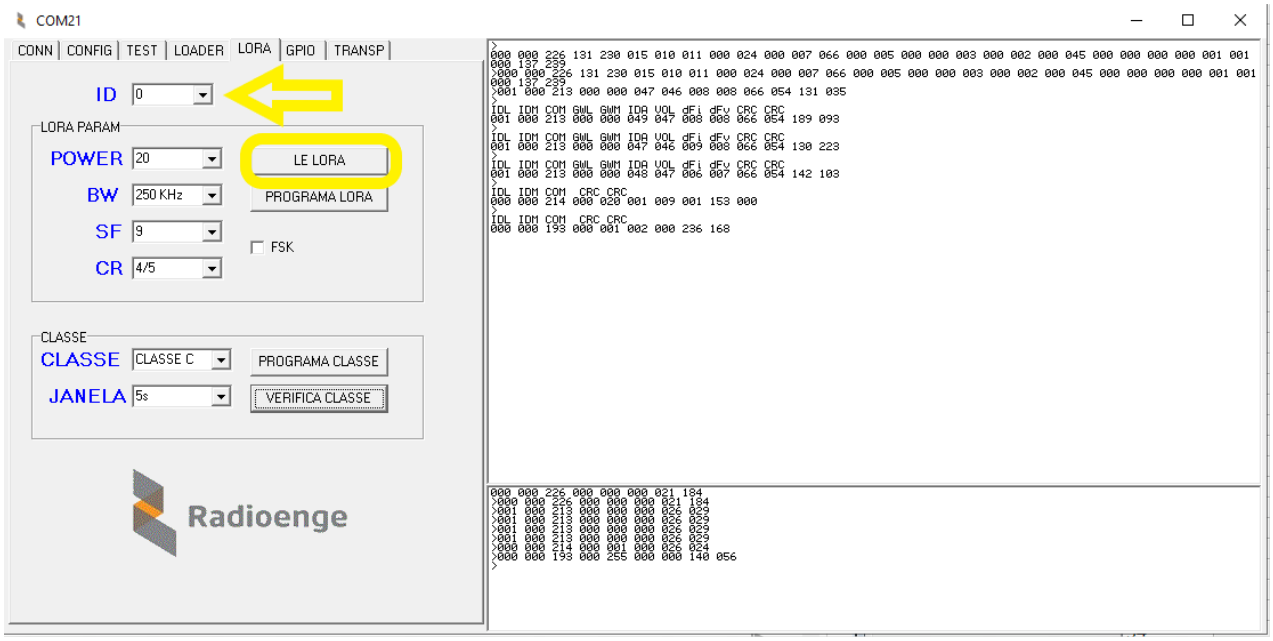


Figura 10. Aba LORA e parâmetros configuráveis

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 mantem 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.

8. Características Físicas

8.1. Dimensões

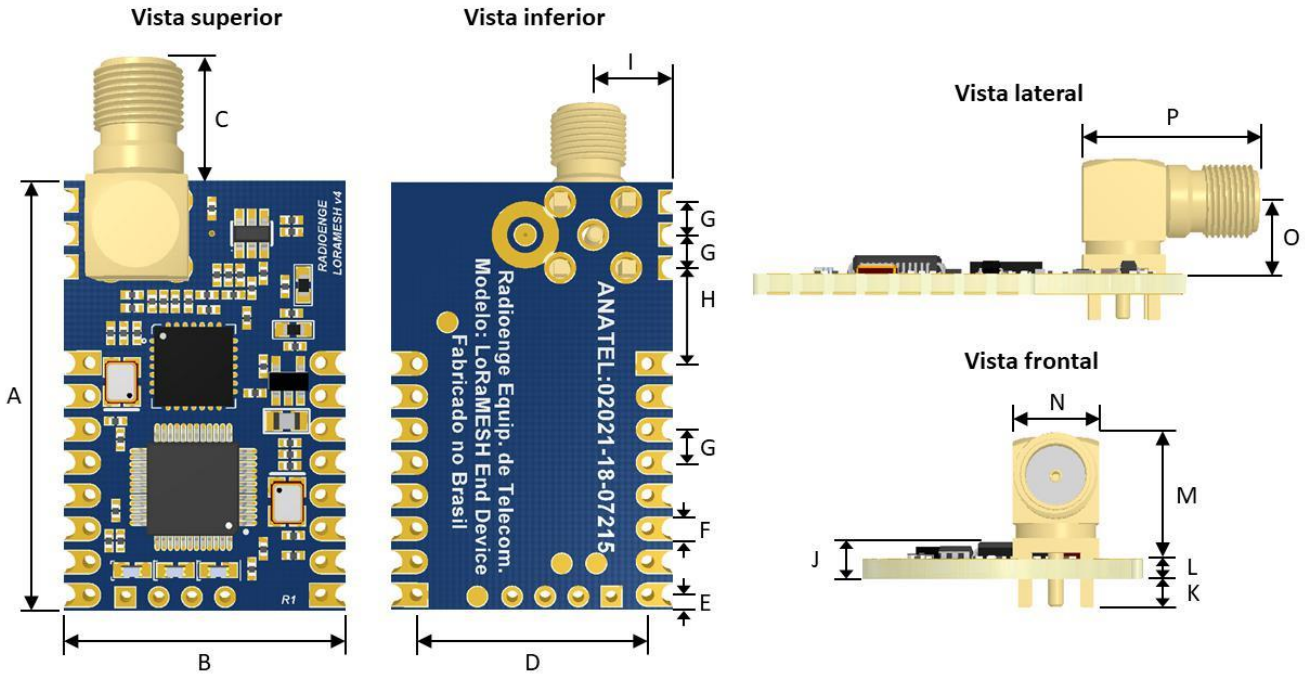


Figura 11. Vistas e dimensões do módulo

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

8. Características Físicas

8.2. *Footprint* recomendado

O *footprint* apresentado abaixo pode ser obtido na página do EndDevice Radioenge nos formatos compatíveis com Eagle e Altium.

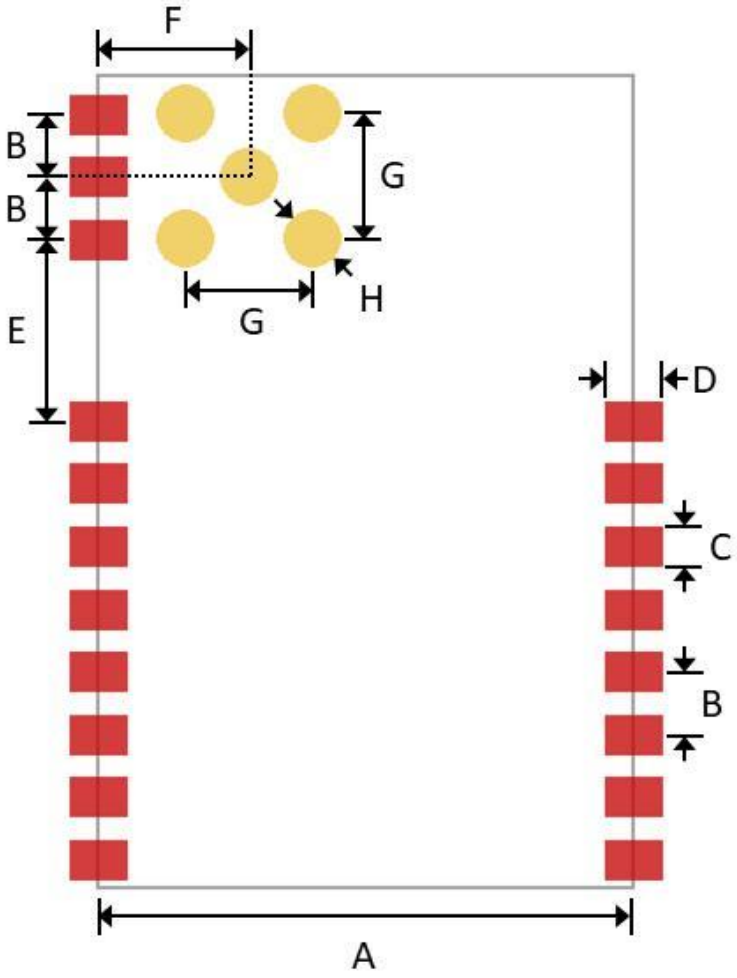


Figura 12. *Footprint* recomendado e dimensões

Dimensões do *Footprint*

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

9. Contato

Manuais e softwares estão disponíveis em nossa página WEB:

https://www.radioenge.com.br/solucoes/iot/34-modulo-loramesh.html

Dúvidas:

https://forum.radioenge.com.br

Dúvidas específicas enviar e-mail para:

iot@radioenge.com.br