

Memento das Funções e Comandos AT empregados com a Biblioteca “LoRaWAN.h” do node RD49C

O módulo LoRaWAN - protocolo de rede 1.0.3 (Classe A ou C) – **End-Device (RD49C) Radioenge** é um produto voltado à concepção de uma rede LPWA (Low Power Wide Area) e opera na banda ISM 915 MHz (região **AU915**). Possui **10 GPIOs**: (i) 2 entradas analógicas (EA); (ii) 6 entradas/saídas digitais; e (iii) 2 UART-Transparente ou EA. **µC**: ARM Cortex-M0+ 32 bits (STM32L071CZT6). **Alimentação**: 1,8 a 12 Vcc (3,3 Vcc sem regulador).

ANATEL: 02021-18-07215.

Comandos: AT+
 =? (para Get – solicita/obtem a leitura do valor pré-estabelecido) Exemplo: AT+CLASS=?
 =__ (para Set – estabelece/define o valor em HEX – espaçar os bytes com “:”) Ex: AT+DADDR=26:0c:1e:78 (somente comando para Run - executa)
 ? (para Help – verifica se existe o comando)

- **AT+DEUI** Get o DevEUI (“MAC” RD49C) (OTAA)
 - **AT+APPEUI** Get/Set o AppEui/JoinEui (se OTAA)
 - **AT+APPKEY** Get/Set o AppKey (se OTAA)
 - **AT+DADDR** Get/Set o DevAddr (se ativação ABP)
 - **AT+APPSKEY** Get/Set o AppSKey (se ABP)
 - **AT+NWKSKEY** Get/Set o NwkSKey (se ABP)

- **AT+CLASS** Get/Set a classe dispositivo (=A ou C)
 - **AT+ADR** Get/Set o ADR (=0 ou 1)
 - **AT+TXP** Get/Set o Tx Power (=0: Máx / 15: Min)
 - **AT+DR** Get/Set o Datarate(=0 ... 13)
 - **AT+PNM** Get/Set o Public Network (=0 ou 1)
 - **AT+RX2FQ** Get/Set a frequência da janela Rx2
 - **AT+RX2DR** Get/Set o datarate da janela Rx2
 - **AT+RX1DL** Get/Set o delay da janela Rx1
 - **AT+RX2DL** Get/Set o delay da janela RX2
 - **AT+JN1DL** Get/Set o delay do Join janela Rx1
 - **AT+JN2DL** Get/Set o delay do Join janela Rx2
 - **AT+NJM** Get/Set o modo Join [0: ABP / 1: OTAA]
 - **AT+NWKID** Get/Set o Network ID
 - **AT+JOIN** Run o procedimento de join
 - **AT+NJS** Get o status do join [0 ou 1]
 - **ATZ** Run um reset no dispositivo

- **AT+DCS** Get/Set o ETSI Duty Cycle (=0 ou 1)
 - **AT+SENDB** Envia dados hexadecimais junto com o Nr da porta definida pelo aplicativo (=port:payload)
 - **AT+SEND** Envia dados de texto (ASCII) junto com o Nr da porta definida pelo aplicativo (=port:payload)
 - **AT+VER** Get a versão do firmware (atual 1.3.35B0)
 - **AT+CFM** Get/Set a opção de confirmação (=0 ou 1)
 - **AT+SNR** Get o SNR do último pacote recebido
 - **AT+RSSI** Get o RSSI do último pacote recebido
 - **AT+BAT** Get o nível da bateria [255: erro / 254: 100% ... 1: 0%]
 - **AT+BAUDRATE** Get/Set o **baudrate** da interface UART
 - **AT+SFCNT** Get/Set a persistência do frame counter (=0 ou 1)
 - **AT+NBTRIALS** Get/Set o Nr de tentativas de retransmissão
 - **AT+KEEPALIVE** Get/Set os **pacotes de keepalive** do dispositivo (=enable:port:confirmation:periodicity_ms)
 - **AT+TXCFM** Envia dados de texto junto com a porta, acrescido de confirmação e Nr de tentativas (=port:conf:nbtrials:payload)
 - **AT+CHMASK** Get/Set a **máscara** dos **canais de frequências**
 - **AT+GPIOC** Configura os pinos de I/O (=pin:mode:pull)
 - **AT+WPIN** Run a escrita dos pinos (=pin:level)
 - **AT+RPIN** Run a leitura dos pinos (=pin)
 - **AT+ADC** Run lê os pinos analógicos (=pin) [0: 0V ... 4095: 3,3V]

“**baudrate**” na interface UART [0: 9600 / 1: 19200 / 2: 43000 / 3: 115200] Obs.: 8-N-1-N (pacote,Par,stopbit,Ctrl)
 “**máscaras**” dos **canais**: ff00:0000: 0000:0000:0002:0000 (TTN) ou 00ff:0000: 0000:0000:0001:0000 (EveryNet)
 “**mode**” [0: INPUT / 1: OUTPUT_Push-Pull / 2: OUTPUT_Open-Drain / 3: AF_PP / 4: AF_OD / 5: ANALOG / 6: IT_RISING / 7: IT_FALLING / 8: IT_RISING_FALLING]
 “**pull**” [0: NO_PULL / 1: PULL_UP / 2: PULL_DOWN]
 “**pin**” [0: xy / 1: xy / 2: x / 3: x / 4: x / 5: x / 6: x / 7: xy / 8: xy / 9: x] Leg: “x” – uso geral e/ou “y” – entrada analógica

FUNÇÕES PÚBLICAS

- SoftwareSerial* **SerialCommandsInit** (uint8_t rxPin, uint8_t txPin, uint32_t baudRate);
 - SoftwareSerial* **SerialTranspInit** (uint8_t rxPin, uint8_t txPin, uint32_t baudRate);
 - Status_Typedef **ReceivePacketCommand** (char* payload, uint8_t* payloadSize, uint32_t timeout);
 - Status_Typedef **ReceivePacketTransp** (char* payload, uint8_t* payloadSize, uint32_t timeout);
 - Status_Typedef **InitializeOTAA** (char* p_appkey, char* p_appeui);
 - Status_Typedef **IsJoined** ();
 - Status_Typedef **JoinNetwork** (uint8_t retries);
 - Status_Typedef **SendString** (char* string, uint8_t port);
 - Status_Typedef **SendRaw** (char* payload);
 - Status_Typedef **SendAtCommand** (AT_Commands_e, CommandType_e, char* payload);

“**Status_Typedef**”: [RAD_OK ou RAD_ERROR]

“**AT_Commands_e**”: [AT_DEVADDR, AT_APPKEY, AT_APPSKEY, ...]

“**CommandType_e**”: [AtGet, AtSet ou AtRun]

Fontes:

- (i) <https://github.com/Radioenge/LoRaWAN> ; (ii) [End-DeviceRD49C_PINOUT](#); (iii) [fórum](#); e
 (iv) <https://www.radioenge.com.br/uploads/fe3eaca2f4e3fd565143af8cb9703d7d1560427722-manual-lorawan-v2.1.pdf>.

