

LoraWan

Alexandro Vanderley dos Santos

Resumo—O objetivo desse artigo é auxiliar na estruturação de uma rede utilizando Lora® e os servidores da TTN (The Things Network), com o propósito de operar sensores remotamente.

I. INTRODUÇÃO

O controle e monitoramento remoto de sensores e dispositivos eletrônicos como atuadores, sensor de temperatura, pressão, umidade, intensidade luminosa entre outras aplicações, está em evidência, hoje. Quase tudo que utilizamos tende a não necessitar de uma conexão por fio, um exemplo bem comum são os dispositivos de telefonia celular.

Para a maioria dos sensores, a velocidade de transmissão de dados não é um quesito relevante na construção de um projeto, por exemplo, a intensidade da luz solar durante o dia, apenas pequenas amostras são tomadas durante o período diurno, mantendo-se ocioso durante a noite. Neste caso uma rede com baixa taxa de transmissão, mas com um consumo reduzido, torna viável a coleta de dados e evita que tenhamos um consumo de energia alto, mantendo, caso este fosse alimentado por uma bateria, a carga por mais tempo.

LoRa®, a camada física, é uma tecnologia de rádiofrequência que permite comunicação à longas distâncias com consumo mínimo de energia. Os módulos enviam e recebem dados de Gateways específicos (similar as redes wifi, mas com alcance muito maior), que os encaminham via conexão IP para servidores locais ou remotos [1].

O protocolo LoRaWAN™, camada lógica da rede, implementa os detalhes de funcionamento, segurança, qualidade do serviço, ajustes de potência, visando maximizar a duração da bateria dos módulos, e os tipos de aplicações, tanto do lado do módulo quanto do servidor [1].

A The Things Network [5], um membro colaborador da LoRa Alliance, responsável pela tecnologia LoRa®, disponibiliza formas de visualização de dados coletados, servindo como captador de dados e formatando-os de maneira que possam ser interpretados pelo usuário.

Aplicando esses conceitos construiremos uma rede de sensores, desde a configuração do rádio até a forma de observar os dados (aplicação), bem como a interação com dispositivos externos.

II. MATERIAL UTILIZADO

A. Hardware

- Discovery Kit da STMicroelectronics, modelo STM32L072CZY6TR MCU [2];

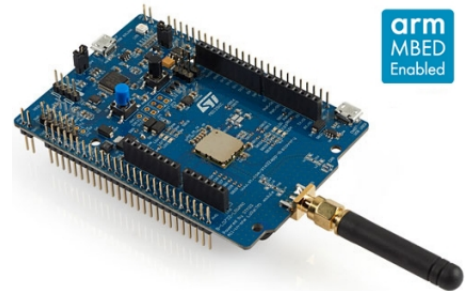


Figura 1: Rádio Lora.

- Cabo micro USB ;



Figura 2: Cabo micro USB.

- Gateway Lora®;
- Desktop ou notebook;
- Sensores ou dispositivos externos.

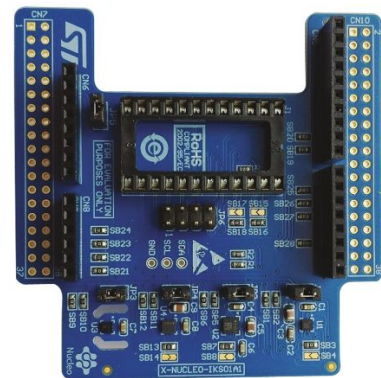


Figura 3: Módulo sensor IKS01A2 (Temperatura, pressão, unidade, acelerômetro e magnetômetro).

-
- A collection of various electronic components including resistors, capacitors, diodes, transistors, integrated circuits, and connectors.

B. Software

- ### III. PREPARAÇÃO DO SISTEMA

- Passo 1: Abrir um terminal;



Figura 5: Abrindo o terminal.

- Passo 2: Instalar o Cmake:

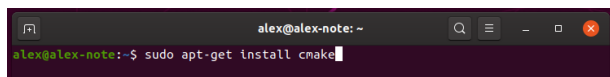


Figura 6: Instalando Cmake.

- Passo 3: instalar o Cutecom;

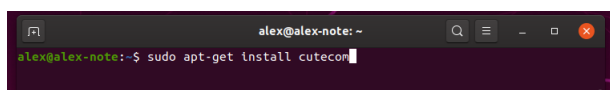


Figura 7: Instalando o Cutecom.

- Passo 4: Instalar o compilador GCC-ARM;

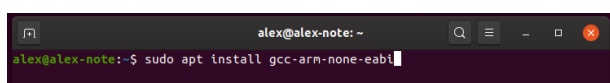


Figura 8: Instalando o GCC-ARM.

- Passo 5: Instalar o gerenciador de versão GIT;

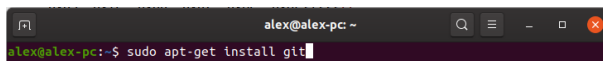


Figura 9: Instalando o GIT.

- Passo 6: Clique **aqui** para baixar o software LoraMac compactado ou utilize o git;

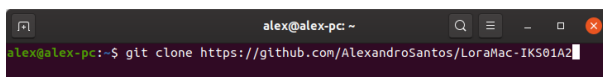


Figura 10: Clonando o software do gitHub.

- Passo 7: descompactar LoraMac.

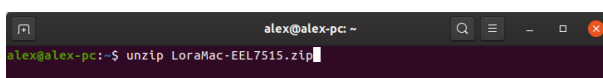


Figura 11: Descompactando LoraMac-EEL7515.

Atenção: Caso o usuário não use o Ubuntu, alguns erros, por falta de bibliotecas, podem ocorrer. Desta forma, tais bibliotecas devem ser instaladas, conforme a necessidade de cada distro linux.

B. Preparando o Discovery Kit da STMicroelectronics

- Passo 1: Conectar o cabo usb;
- Passo 2: Após conectar, o Ubuntu deverá alertar sobre a inserção de um disco removível (DIS_L072Z).

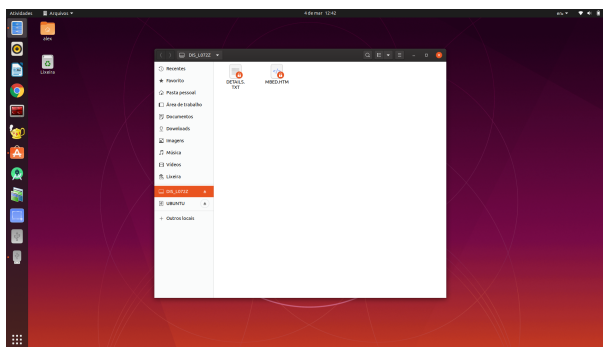


Figura 12: Abrindo o disco removível DIS_L072Z.

Atenção: Para que o disco realmente fique disponível para cópia, é necessário que você abra pelo menos uma vez, após conectar o Kit. Isto equivale a dizer que devemos montar o dispositivo para uso. Neste disco, devemos copiar o arquivo `LoRaMac-classA.bin`, futuramente compilado. Lembrando que o linux é sensível a letras MAIÚSCULAS e minúsculas.

IV. CRIANDO UMA APLICAÇÃO NA TTN

- Passo 1: Abrir o site da **TTN**;
- Passo 2: Criar uma conta e acessá-la;

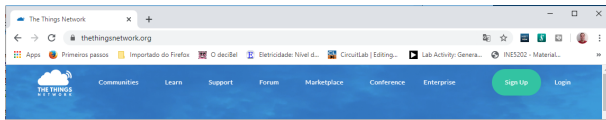


Figura 13: Criar conta e acessar

- Passo 3: Entrar em Console (seta ao lado da foto);



Figura 14: Acessar o console da TTN

- Passo 4: Acessar APPLICATIONS para abrir, ou criar, suas aplicações;



Figura 15: Link para acessar as aplicações

- Passo 5: Criar uma nova aplicação em **add application**;

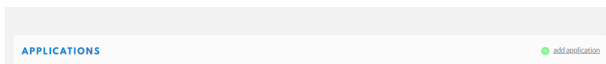


Figura 16: Acessar o console da TTN

- Passo 6: Preencha os campos e clique em **Add application**;

Figura 17: Criando a aplicação

- Passo 7: Clique no botão **Devices** para adicionar um dispositivo (Rádio);

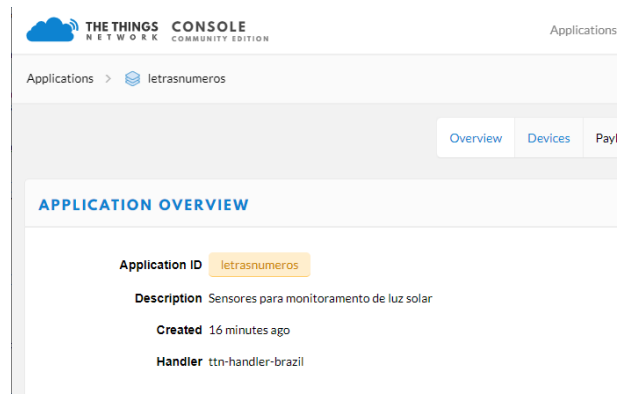


Figura 18: Criando um dispositivo

- Passo 8: Clique em **register device**;

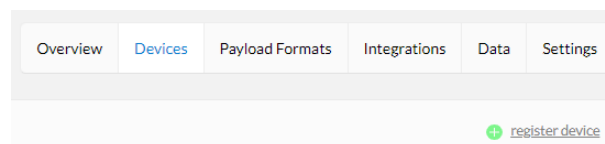


Figura 19: Registrar um dispositivo

- Passo 9: Preencha o campo **Device ID**;

REGISTER DEVICE [bulk import devices](#)

Device ID
This is the unique identifier for the device in this app. The device ID will be immutable.
sensorluz

Device EUI
The device EUI is the unique identifier for this device on the network. You can change the EUI later.
this field will be generated

App Key
The App Key will be used to secure the communication between you device and the network.
0 bytes

App EUI
70 B3 D5 7E D0 02 BC 8F

Figura 20: Registrando um dispositivo

Neste passo, se você já possui um Device EUI e uma App Key, basta clicar na figura da caneta para inserir os valores. Caso não tenha, a TTN gerará uma App Key e uma Device EUI.

- Passo 10: Clique em **Settings**;

DEVICE OVERVIEW

Application ID: lettrasnumeros

Device ID: sensorluz

Activation Method: OTAA

Device EUI: 00 17 AA 38 F8 E0 78 14

Application EUI: 70 B3 D5 7E D0 02 BC 8F

App Key: [redacted]

Status: never seen

Frames up: 0 [reset frame counters](#)

Frames down: 0

Figura 21: Configurar **Activation Method** em ABP

- Passo 11: Configurando **Activation Method** em ABP;

Activation Method

OTAA ABP

Figura 22: Registrando um dispositivo

Ativando o modo ABP, apenas salve em **Save**. Após este passo, já temos todas as informações para configurar o software LoraMac, baixado do repositório.

DEVICE OVERVIEW

Application ID: lettrasnumeros

Device ID: sensorluz

Activation Method: ABP

Device EUI: 00 17 AA 38 F8 E0 78 14

Application EUI: 70 B3 D5 7E D0 02 BC 8F

Device Address: 26 03 14 D7

Network Session Key: 0F 78 42 D4 25 EC 0B CC 49 A5 CA 54 57 E7 97 8B

App Session Key: 6E 19 04 1B 3D FF 5A 0B B9 AE 61 81 4A 73 C5 89

Status: never seen

Frames up: 0 [reset frame counters](#)

Frames down: 0

Figura 23: Informações para configurar o rádio

Atenção: Alteração de servidor na **aplicação** implicará em perda da configuração do dispositivo, portanto é importante manter os dados da tela acima guardados para futuras alterações. O mesmo ocorre quando alteramos algo no dispositivo, como o Device Address não pode ser editado, é sempre gerado pela TTN, pode haver a necessidade de recompilar o projeto com as novas especificações.

V. CONFIGURANDO O CÓDIGO FONTE LORAMAC

Para que nosso rádio conecte-se de forma adequada ao gateway e a TTN, precisamos editar dois arquivos, seguindo os seguinte passos:

- Passo 1: Abrir, com um editor de sua preferência (gedit, nano, vi), o arquivo **Commissioning.h** localizado na pasta **src/apps/LoRaMac/classA/B-L072Z-LRWAN1**. Neste arquivo, serão editadas as linhas contendo as seguinte definições (#define), seguindo os dados da figura 23:
 - **OVER_THE_AIR_ACTIVATION 0**
Desativa o método de ativação OTAA, passado a ser ABP.
 - **IEEE_OUI 0x00, 0x17, 0xAA**
Utilizar os 3 primeiro bytes de **Device EUI**
 - **LORAWAN_DEVICE_EUI {IEEE_OUI, 0x38, 0xF8, 0xE0, 0x78, 0x14}**
Utilizar os 5 últimos bytes de **Device EUI**
 - **LORAWAN_JOIN_EUI {0x70, 0xB3, 0xD5, 0x7E, 0xD0, 0x02, 0xBC, 0x8F}**
Utilizar **Application EUI**
 - **LORAWAN_APP_KEY**
 - **LORAWAN_APP_S_KEY**
Ambos devem ser setados com **App Session Key**
0x6E, 0x19, 0x04, 0x1B, 0x3D, 0xFF, 0x5A, 0x0B, 0xB9, 0xAE, 0x61, 0x81, 0x4A, 0x73, 0xC5, 0x89
 - **LORAWAN_NWK_KEY**
 - **LORAWAN_F_NWK_S_INT_KEY**
 - **LORAWAN_S_NWK_S_INT_KEY**

- LORAWAN_NWK_S_ENC_KEY

Todos devem ser setados com **Network Session Key** 0x0F, 0x78, 0x42, 0xD4, 0x25, 0xEC, 0x0B, 0xCC, 0x49, 0xA5, 0xCA, 0x54, 0x57, 0xE7, 0x97, 0x8B

- LORAWAN_DEVICE_ADDRESS (uint32_t) 0x260314D7

Deve ser setado com **Device Address**

- Passo 2: Salve o arquivo Commissioning.h;

VI. COMPILANDO O PROJETO

Para apenas compilar use **./create.sh**, este pequeno script criará o diretório **build**. No caminho **build/src/apps/LoRaMac**, você encontrará o arquivo **LoRaMac-classA.bin**. Este deverá ser copiado para o disco removível **DIS.L072Z**, caso queira rodar o programa na placa.

Para compilar e programar a placa você pode utilizar **./program.sh** que todos os passos serão feitos automaticamente.

VII. VERIFICANDO A TRANSMISSÃO

A. Local

Utilizando o programa Ccutecom, podemos observar o comportamento do rádio, quais pacotes foram transmitidos e recebidos. Para executar o programa digite no terminal:

sudo cutecom

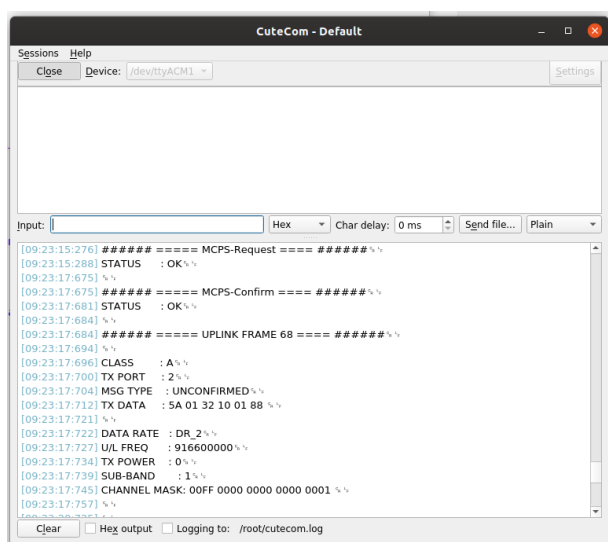


Figura 24: Tela do programa Ccutecom

Ao abrir você deve selecionar o **Device** (Porta serial /dev/ttyACM*) e pressionar o botão **Open**. Com o rádio conectado, as informações começarão a rolar a cada 5 segundos, tempo este pré-configurado no software LoraMac.

B. Remoto

No site da TTN, com a aplicação aberta, pressione o botão **Data**.

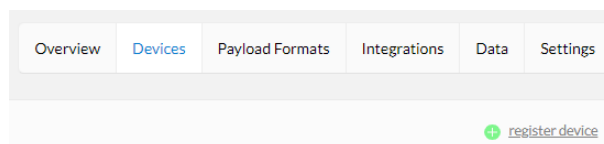


Figura 25: Seleção de Data

Agora, podemos monitorar os pacotes recebidos e enviados pelo dispositivo.

| Filters | | | | | |
|----------|----------|------------|----------------------------|------------------------------|--|
| uplink | downlink | activation | ack | error | |
| time | counter | port | | | |
| 09:21:22 | 0 | | | | |
| 09:21:21 | 45 | 2 | payload: 5A 01 31 0E 02 06 | potValue: 1.1682243076218814 | |
| 09:21:07 | 0 | | | | |
| 09:21:07 | 42 | 2 | payload: 5A 01 3C 1D 02 7B | potValue: 1.1682243076218814 | |
| 09:20:37 | 0 | | | | |
| 09:20:36 | 36 | 2 | payload: 5A 01 2F 29 04 81 | potValue: 1.1682243076218814 | |
| 09:20:32 | 0 | | | | |
| 09:20:31 | 35 | 2 | payload: 5A 01 37 3C 01 13 | potValue: 1.1682243076218814 | |
| 09:20:21 | 0 | | | | |
| 09:20:21 | 33 | 2 | payload: 5A 00 FC 25 01 59 | potValue: 1.1681739528496222 | |
| 09:19:12 | 0 | | | | |
| 09:19:12 | 19 | 2 | payload: 5A 01 1A 45 04 4F | potValue: 1.1682243076218814 | |

Figura 26: Dados transmitidos e recebidos

Também podemos verificar as características das mensagens e o comportamento do rádio como potência, tempo de resposta e relação sinal ruído.

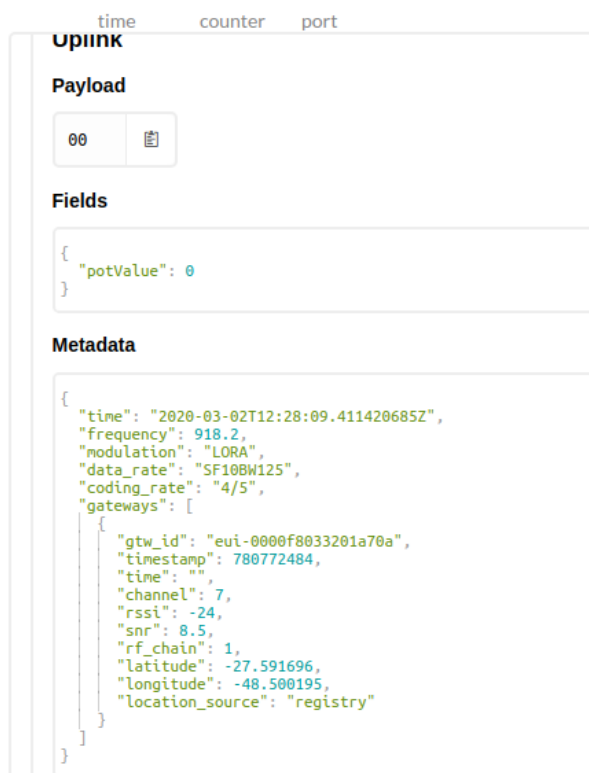


Figura 27: Características da mensagem

Para visualização dos dados em um formato legível, podemos configurar esta apresentação na aba **Payload Formats**. O exemplo da figura 28 está configurada para o programa fornecido.

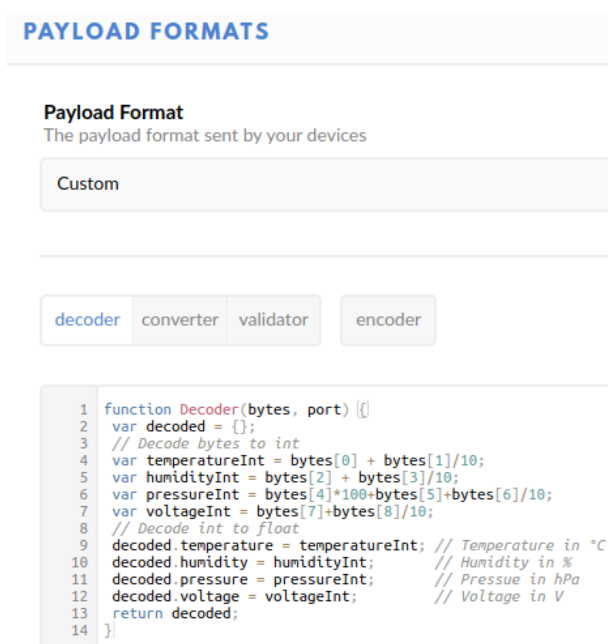


Figura 28: Decodificação do pacote de dados.

VIII. PERSONALIZAÇÃO

No programa LoRaMac fornecido é possível ligar ou desligar um dos leds da placa e, ao mesmo tempo, fazer uma leitura da tensão no pino A0, informações como temperatura, pressão e umidade também são transmitidas periodicamente, a saber, a cada 5 segundos.

Com o intuito de personalizar a aplicação, podemos utilizar as portas digitais, analógicas e de comunicação para implementar novas funcionalidades, respeitando a estrutura básica do código fonte. Para tanto, precisamos editar o arquivo **main.c** encontrado na pasta **src/apps/LoRaMac/classA/B-L072Z-LRWAN1**.

No arquivo **main.c** encontramos algumas configurações para o rádio e para conexão com o gateway, como por exemplo :

```
#define APP_TX_DUTYCYCLE
```

que informa ao módulo o ciclo de transmissão em milissegundos, atualmente 5000 (5 s).

A função **PrepareTxFrame** é responsável por montar o pacote de dados para o envio, nela podemos implementar, por exemplo a leitura de um sensor, lembrando que serão necessárias as bibliotecas responsáveis pelo controle do sensor ou dispositivo anexado, escrito em C. Dentro do arquivo **tools.c** temos uma biblioteca para os sensores HTS221, temperatura e umidade, e LPS22HB, pressão atmosférica do Kit IKS01A2.

A função **McpsIndication**, além de outras atividades, é também responsável pelo recebimento dos dados e tratamento deles, nesta parte podemos acionar dispositivos remotamente ou requisitar uma leitura adicional, por exemplo.

Na pasta **src/system** encontram-se todos os métodos necessários para utilizar o microcontrolador embutido no Kit. Drivers para utilização da Gpio, i2c, SPI e ADC são alguns exemplos, estes, por sua vez, mais utilizados na adaptação de sensores e atuadores externos.

Com este conhecimento, já torna-se possível desenvolver muitas aplicações, cabe a necessidade e disponibilidade de equipamentos, sensores e conhecimento em linguagem C para aprimorar e expandir as funcionalidades do Kit.

REFERÊNCIAS

- [1] <<https://newtoncbraga.com.br/index.php/electronica/52-artigos-diversos/11992-conheca-a-tecnologia-lora-e-o-protocolo-lorawan-lor001>>. Acesso em: 04 mar. 2020.
- [2] <<https://www.st.com/en/evaluation-tools/b-l072z-lrwan1.html>>. Acesso em: 04 mar. 2020.
- [3] <<https://ubunut.com>>. Acesso em: 04 mar. 2020.
- [4] <<https://Loramac.com>>. Acesso em: 04 mar. 2020.
- [5] <<https://www.thethingsnetwork.org/>>. Acesso em: 04 mar. 2020.