

UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL
ESCOLA DE ENGENHARIA
ENGENHARIA ELÉTRICA

RENAN RODRIGUES

**APLICAÇÃO DE MÉTODOS PARA
LIDAR COM PARADIGMAS DE
SINCRONIZAÇÃO EM REDES DE
SENSORES SEM FIO LORA**

Porto Alegre
2022

RENAN RODRIGUES

**APLICAÇÃO DE MÉTODOS PARA
LIDAR COM PARADIGMAS DE
SINCRONIZAÇÃO EM REDES DE
SENSORES SEM FIO LORA**

Projeto de Diplomação (PD-ELE) apresentado à COMGRAD-ELE da Universidade Federal do Rio Grande do Sul como parte dos requisitos para a obtenção do título de *Bacharel em Engenharia Elétrica*.

ORIENTADOR: Prof. Dr. Ivan Müller

Porto Alegre
2022

RENAN RODRIGUES

**APLICAÇÃO DE MÉTODOS PARA
LIDAR COM PARADIGMAS DE
SINCRONIZAÇÃO EM REDES DE
SENSORES SEM FIO LORA**

Este Projeto de Diplomação foi julgado adequado para a obtenção dos créditos da Disciplina de PD do curso *Engenharia Elétrica* e aprovado em sua forma final pelo Orientador e pela Banca Examinadora.

Orientador: _____
Prof. Dr. Ivan Müller, UFRGS
Doutor pela Universidade Federal do Rio Grande do Sul –
Porto Alegre, Brasil

Banca Examinadora:

Me. Max Feldman, UFRGS
Mestre pela Universidade do Rio Grande do Sul - Porto Alegre, Brasil

Prof. Dr. Marcelo Götz, UFRGS
Doutor pela Universität Paderborn – Paderborn, Alemanha

Porto Alegre, novembro de 2022.

DEDICATÓRIA

Aos meus pais, Isabel e Eder que sempre me apoiaram na minha jornada acadêmica.

RESUMO

Atualmente, somos capazes de controlar, monitorar e realizar diversas tarefas remotamente. Esta redução da distância física é feita através da conexão de diversos objetos através de protocolos de comunicação. IoT (*Internet of Things*) é a tecnologia para conectividade de vários objetos com a Internet. Por serem comumente aplicados aos sistemas embarcados, estes se tornaram uma parte importante de nossas vidas. Esses sistemas são alimentados normalmente por baterias e precisam apresentar elevada autonomia. Portanto, a eficiência desses sistemas requer tecnologias que consumam pouca energia, concomitantemente com capacidade de cobrir longas distâncias. Tecnologias *Low-Power Wide-Area Network (LPWAN)*, tais como o *Long Range*, LoRa) estão emergindo, oferecendo conectividade acessível com comunicação sem fio de baixo consumo de energia sobre grandes áreas geográficas. LoRa está ganhando popularidade principalmente em sistemas embarcados que precisam transferir pequenas quantidades de dados em longas distâncias. Tendo em vista os benefícios para sistemas embarcados e a crescente popularidade das redes LoRa no mercado de IoT, este trabalho apresenta uma implementação prática destas redes, objetivando agregar técnicas de sincronismo a este protocolo. A rede é composta por nodos sensores e coletor LoRa genéricos, de baixo custo, autogerenciáveis, focando na solução sincronização das transmissões entre nodos presentes na rede de rádio sensores sem fio através da utilização de *slots* de tempo e *offsets* de correção temporal para sequenciar as transmissões dos nodos.

Palavras-chave: *LoRa, Low-Power Wide-Area Network, Sincronização de Transmissões, Redes Sensores Sem Fio.*

ABSTRACT

Nowadays, we are able to control, monitor, and do a lot more from remote distance. This reduction of physical distance is done by the connection of various objects by means of communication protocols. IoT (Internet of Things) is the technology that connects various objects with Internet. Commonly associated with embedded systems, these devices have become an important part of our lives. These systems commonly are battery operated and need a high autonomy. Thus, the efficiency of these systems requires a technology that has a low power consumption and also that covers long distances. Low-Power Wide-Area Network (*LPWAN*) technologies such as Long Range (*LoRa*) are emerging and offering affordable connectivity with power efficient wireless communication over large geographical areas. *LoRa* is gaining popularity mainly in embedded systems that need to transfer small amounts of data at short intervals over long distances. *LoRa* is gaining popularity mainly in embedded systems that need to transfer small amounts of data over long distances. Regarding the benefits for embedded systems and the growing popularization of the *LoRa* network in the IoT market, this paper proposes an implementation of a generic low cost stable and self-manageable *LoRa* network of sensor nodes focusing on the solution of transmissions synchronization between the nodes present on the wireless sensor network by using time slots and time corrections offsets to sequence the nodes transmissions.

Keywords: *LoRa*, Low-Power Wide-Area Network, Transmissions synchronization, Wireless Sensor Network.

SUMÁRIO

LISTA DE ILUSTRAÇÕES	8
LISTA DE ABREVIATURAS	10
1 INTRODUÇÃO	12
2 MOTIVAÇÃO	14
3 FUNDAMENTAÇÃO TEÓRICA	16
3.1 Redes <i>Low Power Wide Area</i> (LPWA)	16
3.2 O protocolo de comunicação LoRa	17
3.2.1 Parâmetros Regionais LoRa	18
3.2.2 Parâmetros de Transmissão LoRa	18
3.2.3 Modulação <i>Chirp Spread Spectrum</i> (CSS)	19
3.3 Protocolos de Comunicação SPI e I²C	20
3.3.1 Protocolo de comunicação I ² C	20
3.3.2 Protocolo de comunicação SPI	22
3.4 Filtro Média Móvel	25
3.5 Unix Time	26
3.5.1 Problema do ano 2038	26
4 MÉTODOS E MATERIAIS	27
4.1 Elementos de Hardware	27
4.1.1 Microcontrolador	27
4.1.2 Módulo LoRa	28
4.1.3 Real Time Clock	29
4.1.4 Dispositivo Completo	29
4.2 Implementação do <i>Firmware</i>	30
4.2.1 Nodo coletor LoRa	31
4.2.1.1 Gerenciador de <i>time slot</i>	32
4.2.1.2 Conteúdo do pacote de <i>advertisement</i>	33
4.2.1.3 <i>Timeout</i> : Nodo coletor	34
4.2.2 Nodo sensor LoRa	34
4.2.2.1 Identificador único dos nodos sensores	36
4.2.2.2 Conteúdo do pacote de um nodo sensor	36
4.2.2.3 <i>Offset</i> de correção do <i>RTC</i> do nodo sensor	36
4.2.2.4 Métodos para evitar transmissão concorrente	39
4.2.2.5 <i>Timeout</i> : Nodo coletor	41

4.2.3	Operação plena da rede	41
5	EXPERIMENTOS E RESULTADOS	43
5.1	Validação do sincronismo entre nodos	43
5.2	Validação das conexões na rede por inicialização individual de nodo sensor	58
5.3	Validação das conexões na rede por nodo sensor em transmissão concorrente	60
5.4	Validação do mecanismo de <i>timeout</i>	62
5.4.1	Validação do <i>timeout</i> : Nodo sensor	62
5.4.2	Validação do <i>timeout</i> : Nodo sensor	64
6	CONCLUSÃO	67
	REFERÊNCIAS	69

LISTA DE ILUSTRAÇÕES

Figura 1:	Exemplo de <i>RSSF</i>	14
Figura 2:	Largura de banda necessária versus capacidade de alcance de diferentes tecnologias sem fio.	17
Figura 3:	Transmissão <i>LoRa</i>	19
Figura 4:	Exemplo de camada física I ² C.	21
Figura 5:	Exemplo de transferência de dado I ² C.	21
Figura 6:	Um controlador-transmissor endereçando um receptor com um endereço de 7 bits.	22
Figura 7:	(a) Topologia SPI ponto-a-ponto. Um mestre conectado a um único dispositivo escravo. (b) Topologia de um mestre conectado a múltiplos dispositivos escravos.	23
Figura 8:	Transmissão comum de protocolo SPI.	24
Figura 9:	Configurações de clock para o protocolo SPI.	25
Figura 10:	Registrador de controle do protocolo SPI.	25
Figura 11:	Diagrama do funcionamento do filtro média móvel.	26
Figura 12:	Dimensões da placa Arduino Nano.	28
Figura 13:	Módulo transceptor LoRa RMF95W.	29
Figura 14:	Diagrama do hardware do nodo LoRa.	30
Figura 15:	Diagrama de estados do nodo coletor LoRa.	31
Figura 16:	Diagrama do gerenciador de <i>time slots</i> do nodo coletor.	32
Figura 17:	Diagrama do conteúdo do pacote LoRa de <i>advertisement</i> de um nodo coletor.	33
Figura 18:	Diagrama de estados do nodo sensor LoRa.	35
Figura 19:	Diagrama do conteúdo do pacote LoRa de transmissão de um nodo sensor.	36
Figura 20:	Diagrama de <i>offset</i> de correção para um nodo sensor com seu relógio adiantado em relação ao do nodo coletor.	38
Figura 21:	Diagrama de <i>offset</i> de correção para um nodo sensor com seu relógio atrasado em relação ao do nodo coletor.	39
Figura 22:	Diagrama da lógica de cancelamento de transmissão para tentativas de conexão concorrentes.	40
Figura 23:	Diagrama de sequência do processo de conexão de um nodo sensor à <i>RSSF</i> LoRa.	42
Figura 24:	Diagrama de conexão de pontas de prova em nodo coletor e nodo sensor para ensaio de sincronismo.	43
Figura 25:	Sinais relativos ao primeiro <i>ADV</i> do primeiro ensaio de verificação da correção de <i>offset</i>	44

Figura 26:	Medição RTC do nodo coletor (Canal 3) relacionado ao segundo ADV do primeiro ensaio de verificação da correção de <i>offset</i>	45
Figura 27:	Medição <i>time slot</i> de ADV no nodo sensor (Canal 1) relacionado ao segundo ADV do primeiro ensaio de verificação da correção de <i>offset</i>	46
Figura 28:	Medição RTC do nodo coletor (Canal 3) relacionado ao terceiro ADV do primeiro ensaio de verificação da correção de <i>offset</i>	47
Figura 29:	Medição <i>time slot</i> de ADV no nodo sensor (Canal 1) relacionado ao terceiro ADV do primeiro ensaio de verificação da correção de <i>offset</i>	48
Figura 30:	Medição SCLK da comunicação SPI do módulo LoRa (Canal 4) relacionado ao segundo ADV do primeiro ensaio de verificação da correção de <i>offset</i>	49
Figura 31:	Medição SCLK da comunicação SPI do módulo LoRa (Canal 4) relacionado ao terceiro ADV do primeiro ensaio de verificação da correção de <i>offset</i>	50
Figura 32:	Medição do tempo decorrido entre o envio do <i>advertisement</i> pelo nodo coletor (Canal 4) e seu <i>parse</i> no nodo sensor (Canal 2).	51
Figura 33:	Diagrama de conexão de pontas de prova em nodo genérico para ensaio verificação de impacto do <i>halt</i> do RTC no sincronismo.	52
Figura 34:	Medida do sinal do <i>halt</i> (Canal 4) para o segundo ensaio.	53
Figura 35:	Medida do sinal do contador do RTC (Canal 1) para o segundo ensaio.	54
Figura 36:	Medição RTC do nodo coletor (Canal 3) relacionado ao primeiro ADV do primeiro ensaio de verificação da correção de <i>offset</i> após ajuste de equações.	55
Figura 37:	Medição <i>time slot</i> de ADV no nodo sensor (Canal 1) relacionado ao primeiro ADV do primeiro ensaio de verificação da correção de <i>offset</i> após ajuste de equações.	56
Figura 38:	Medição <i>time slot</i> de ADV no nodo coletor (Canal 3) relacionado ao segundo ADV do primeiro ensaio de verificação da correção de <i>offset</i> após ajuste de equações.	57
Figura 39:	Medição da diferença entre <i>time slot</i> do nodo coletor (Canal 3) e do nodo sensor (Canal 1) relacionado ao segundo ADV do primeiro ensaio de verificação da correção de <i>offset</i> após ajuste de equações.	58
Figura 40:	Gráfico do espectro das transmissões dos nodos em conexão na rede LoRa.	59
Figura 41:	Sinais recebidos em analisador de pacotes LoRa durante teste conexão concorrente de nodos sensores.	61
Figura 42:	Sinais recebidos em analisador de pacotes LoRa durante teste de <i>timeout</i> em nodo coletor.	63
Figura 43:	Sinais recebidos em analisador de pacotes LoRa durante teste de <i>timeout</i> em nodo sensor.	65

LISTA DE ABREVIATURAS

ACK	<i>Acknowledgement</i>
ADV	<i>Advertisement</i>
BW	<i>Band-width</i>
CF	<i>Carrier Frequency</i>
CPHA	<i>Clock Phase</i>
CPOL	<i>Clock Polarity</i>
CR	<i>Coding Rate</i>
CSS	<i>Chirp Spread Spectrum</i>
dB	Decibéis
FEC	<i>Forward Error Correction</i>
FIFO	<i>First-In, First-Out</i>
I ² C	<i>Inter-Integrated Circuit</i>
I/O	<i>Input/Output</i>
ID	Identificador
IoT	<i>Internet of Things</i>
ISM	<i>Industrial, Scientific and Medical</i>
LoRa	<i>Long Range</i>
LPWA	<i>Low Power Wide Area</i>
LPWAN	<i>Low Power Wide Area Network</i>
M2M	<i>Machine-to-machine</i>
MAF	<i>Moving Average Filter</i>
MIPS	Milhões de Instruções Por Segundo
MISO	<i>Master In-Slave Out</i>
MOSI	<i>Master Out-Slave In</i>
NACK	<i>Not Acknowledged</i>
NB	<i>Narrow-band</i>

RDS	Rádio Definido por Software
RISC	<i>Reduced Instruction Set Computer</i>
RSSI	<i>Received Signal Strength Indicator</i>
RSSF	Redes de Sensores sem Fio
RTC	<i>Real Time Clock</i>
Rx	<i>Receive</i>
SCA	<i>Serial Clock</i>
SCM	Serviço de Comunicação Multimídia
SCLK	<i>Serial Clock</i>
SDA	<i>Serial Data</i>
SF	<i>Spreading Factor</i>
SMP	Serviço Móvel Pessoal
SNR	<i>Signal to Noise Ratio</i>
SPI	<i>Serial peripheral interface</i>
SS	<i>Slave Select</i>
TP	<i>Transmission Power</i>
Tx	<i>Transmit</i>
WSN	<i>Wireless sensor network</i>

1 INTRODUÇÃO

IoT (*Internet of Things*) está ganhando muita importância em quase todos os campos do desenvolvimento humano, tais como, saúde, indústria, eletrônica de consumo, automotivo e muitos outros (DEVALAL; KARTHIKEYAN, 2018). Podemos controlar as luzes e equipamentos em nossas casas ou sentados em nosso escritório de onde quer que estejamos. De acordo com o estudo (SINHA; WEI; HWANG, 2017), os objetos conectados na Internet serão responsáveis por mais de 25 bilhões de conexões até 2025.

Setores industriais e outras atividades de campo são atualmente muito comumente regulados e controlados a partir de um local remoto. As pessoas são capazes de controlar, monitorar e fazer muito mais à distância. Isto é feito conectando vários objetos, reduzindo a distância física. IoT é a conectividade de vários objetos com a rede Internet. A maioria destas aplicações estão relacionadas a sistemas embarcados, que neste aspecto, se tornaram uma parte importante de nossas vidas. Porém, estes sistemas têm restrições de energia muito altas, já que a maioria deles é operado por bateria, não alimentados externamente, necessitando muitas vezes de uma bateria com alta capacidade (DEVALAL; KARTHIKEYAN, 2018). Ainda, os sistemas embarcados requerem tecnologias que consumam pouca energia ainda que cobrindo longas distâncias de enlace de comunicação. Muitas tecnologias, tais como ZigBee, Wi-Fi, Bluetooth, utilizadas popularmente, consomem pouca energia e são adequadas para sistemas operados por bateria (DEVALAL; KARTHIKEYAN, 2018), por outro lado, não são capazes de cobrir grandes distâncias de enlace de rádio.

Para atender essa demanda por baixo consumo de energia e grande enlace, as tecnologias *Low Power Wide Area Network (LPWAN)* se mostram promissoras. Estas redes, que foram apresentadas comercialmente pela primeira vez em 2009 (SIGFOX..., s.d.), têm sido utilizadas cada vez mais, tanto no âmbito comercial quanto no acadêmico. Como demonstrado nas análises exploradas em (RAZA; KULKARNI; SOORIYABANDARA, 2017), tecnologias *LPWAN* oferecem conjuntos únicos de recursos, incluindo conectividade de área ampla para dispositivos de baixa energia e baixa taxa de dados, não fornecidos por tecnologias sem fio tradicionais. Espera-se que seu mercado seja enorme, aproximadamente um quarto de todos os 30 bilhões de dispositivos IoT/M2M (*Machine-to-Machine*) devem ser conectados à Internet usando redes *LPWAN*, com tecnologias proprietárias ou tecnologias celulares.

Segundo (RAZA; KULKARNI; SOORIYABANDARA, 2017), os setores comerciais que aplicam densamente as tecnologias *LPWAN* incluem, mas não se limitam a cidade inteligente, aplicações pessoais de IoT, redes inteligentes, medições inteligentes, logística, monitoramento industrial, agricultura, etc.

Em relação essas tecnologias *LPWAN*, LoRa (*Long Range*) é uma das que vem mais se destacando nos últimos anos. LoRa propicia de conectividade sem fio para aplicações IoT

que evoluiu recentemente e está ganhando popularidade em sistemas embarcados de baixa potência operados por bateria, que precisam transferir pequenas quantidades de dados em longo alcance (DEVALAL; KARTHIKEYAN, 2018). O protocolo é uma especificação de camada física de comunicação sem fio, propriedade da empresa *Semtech*. Este protocolo de comunicação aplica espalhamento espectral, que tem o objetivo de explorar uma grande variabilidade de aplicações e o gerenciamento de quantidades massivas de objetos. Ela é utilizada no protocolo de comunicação LoRaWAN, sendo que milhares de objetos podem estar conectados a um mesmo gateway em grandes distâncias. Dispositivos LoRa podem ser configurados para usar diferentes fatores de propagação, configurações de largura de banda, taxas de codificação e potências de transmissão, resultando em mais de 6720 configurações possíveis (BOR; ROEDIG, 2017). Apesar da camada física ser proprietária, o protocolo de comunicação LoRa é aberto e utiliza o espectro de radio frequência não licenciado ISM, do inglês *Industrial Scientific and Medical* que varia de 433 MHz à 915 MHz para regiões cobertas pelas Américas.

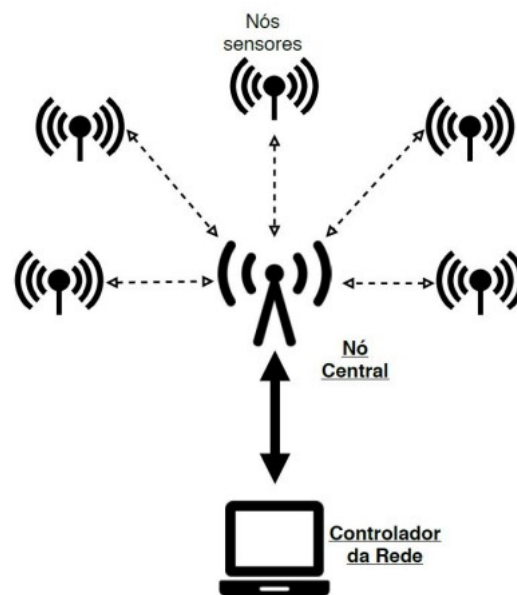
Tendo em vista o crescente aumento de popularidade das redes LPWAN, e pela diversidade de configurações possíveis para o protocolo, este projeto visa a implementação de uma rede de sensores sem fio genérica, comunicantes à um nodo coletor via protocolo LoRa, objetivando consolidar uma rede estável, de baixo custo, e que possa se auto-gerenciar após a inicialização. Para isso, técnicas de sincronismo, eventos em séries temporais e de *timeout* são aplicados para que se estabeleça uma comunicação efetiva entre os nodos sensores e o nodo coletor.

Este trabalho apresenta a topologia do circuito projetado, assim como o motivo por trás da escolha de seus componentes e das técnicas empregadas no processo de desenvolvimento do *firmware*, além das análises experimentais realizadas no protótipo e conclusões.

2 MOTIVAÇÃO

Originalmente abordado no projeto proposto em (VALADÃO, 2021), este trabalho dá continuidade ao desenvolvimento de dispositivos para monitoramento do comportamento reprodutivo dos bovinos, propondo uma rede de sensores sem fio composta por múltiplos nodos sensores e um nodo coletor, como pode ser observado na Figura 1.

Figura 1: Exemplo de *RSSF*.



Fonte: (VALADÃO, 2021).

Para que esta rede sem fio pudesse operar, havia a necessidade de que as transmissões LoRa dos nodos sensores fossem sincronizadas, pois transmissões concorrentes de nodos sensores com mesmo parâmetros de *TP*, *CF*, *SF*, *BW* e *CR*, (definições de parâmetros apresentadas na Seção 3.2.2), ocasionam sinais que interferem um ao outro ao serem recebidos pelo nodo coletor. De forma sintética, objetiva-se o desenvolvimento de uma técnica de acesso ao meio de forma síncrona, aplicada ao LoRa.

A forma como a sincronização era realizada anteriormente tinha como base um arquivo de configuração dos nodos sensores, que configurava tanto o tempo e data do RTC como o *time slot* (segundo do RTC alocado para a transmissão sincronizada de informação) do nodo sensor. Logo, para que a rede pudesse atuar em pleno funcionamento, havia a necessidade de que sempre que fosse se utilizar um nodo sensor, era

preciso conectá-lo a um computador contendo o *firmware* de configuração para configurá-lo e torná-lo apto a operar na rede de sensores estabelecida. Isto implicava em uma dependência de equipamentos para inicialização da rede, tornando o processo lento e trabalhoso. Qualquer problema que ocorresse com um nodos sensor, implicava na necessidade de reconfigurá-lo manualmente para reintroduzi-lo à rede de sensores. Assim, a partir do paradigma de inicialização da rede original, partiu-se para a modificação, eliminando a dependência de outros equipamentos para configuração, visando criar uma rede auto-gerenciável.

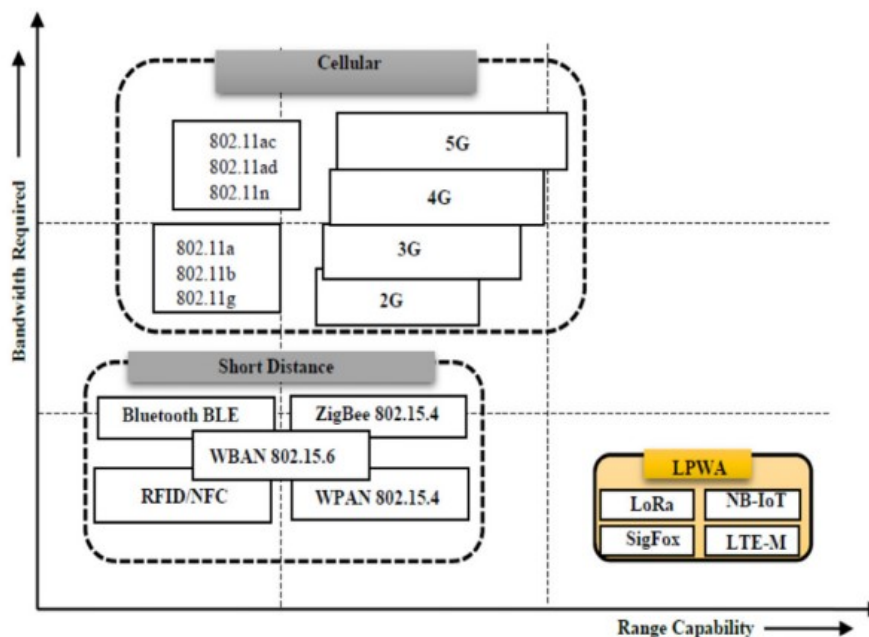
3 FUNDAMENTAÇÃO TEÓRICA

3.1 Redes *Low Power Wide Area* (LPWA)

As redes *Low Power Wide Area* (LPWA) representam um novo paradigma quando se trata de protocolos de comunicação, complementando as tecnologias de redes sem fio de curto alcance para atender às diversas exigências das aplicações de Internet das coisas (*Internet of things* - IoT) (RAZA; KULKARNI; SOORIYABANDARA, 2017)

As tecnologias *LPWAN* conseguem operar com cerca de 140-160 dB de atenuação do caminho total, o que pode resultar em enlaces de muitos quilômetros de alcance, devido às altas sensibilidades típicas do receptor. Sensibilidades de mais de -130 dBm são comuns em *LPWAN*, em comparação com as tecnologias anteriores, entre -90 a -110 dBm. Em outras palavras, receptores -130 dBm de sensibilidade podem detectar sinais 10.000 vezes mais fracos que outros com -90 dBm (LINK LABS, 2016). Na prática, são possíveis redes LPWA projetadas para alcançarem distâncias que pode variar até dezenas de quilômetros com um *link budget* (uma contabilização de todos os ganhos e perdas de potência que um sinal de comunicação sofre) em torno de 150 dB. Duas técnicas, a banda estreita (*narrowband*) e o espalhamento de espectro (*spread spectrum*), são adotadas por diferentes LPWA. A banda estreita proporciona alto *link budget* ao codificar o sinal de banda base com baixa largura de banda (menos de 25 kHz). A técnica *spread spectrum* espalha um sinal de banda estreita sobre uma banda de frequência mais larga com a mesma densidade de potência (RAZA; KULKARNI; SOORIYABANDARA, 2017).

Figura 2: Largura de banda necessária versus capacidade de alcance de diferentes tecnologias sem fio.



Fonte: (SINHA; WEI; HWANG, 2017).

A maioria das tecnologias LPWA existentes no mercado opera em bandas não licenciadas *Sub-GHz*, o que apresenta duas grandes vantagens. A primeira é que, quando comparados a frequências mais altas (i.e. 2,4 GHz), os sinais na banda *Sub-GHz* sofrem menor atenuação devido à propagação de caminhos múltiplos causada por obstáculos (RAZA; KULKARNI; SOORIYABANDARA, 2017). A segunda é que as operadoras não precisam adquirir licenças para operar nessas faixas de rádio, resultando na redução de custos para o consumidor. Contudo, dispositivos que operam no espectro não licenciado estão sujeitos a regulamentações locais que impõem restrições sobre parâmetros, tais como a potência de RF, a taxa cíclica de rádio e o tempo no ar dos pacotes transmitidos.

Como pode ser observado na Figura 2, há diferentes protocolos classificados como LPWA, tais como LoRa, Wi-Fi HaLow, NB-IoT, SigFox e LTE-M. Estes protocolos comparados com os de curta distância ou de celular, exigem menor largura de banda e permitem maior alcance. No entanto, entre si, estes protocolos apresentam diferentes vantagens. A cobertura do LoRa pode ser estendida a grandes regiões como a de uma cidade, enquanto a cobertura NB-IoT é limitada à cobertura LTE; Isto faz com que o NB-IoT não seja conveniente para áreas suburbanas e rurais, onde muitas vezes não há infraestrutura instalada, tornando o LoRa adequado para tais áreas (MIGABO et al., 2017). A distância de cobertura do Wi-Fi HaLow é estimada em 1 km, o que é inferior à do NB-IoT e LoRa (até 10 km).

3.2 O protocolo de comunicação LoRa

LoRa é um protocolo de camada física que emprega o *Chirp Spread Spectrum (CSS)* com *Forward Error Correction (FEC)* integrado. As transmissões usam uma banda larga para combater interferências e para minimizar compensações de frequência causadas por cristais osciladores de baixo custo. Um receptor LoRa pode decodificar transmissões de

até 19,5 dB abaixo do nível de ruído, permitindo assim distâncias de comunicação muito longas. As principais propriedades do LoRa são: longo alcance, alta robustez, resistência a multi-caminhos, resistência ao efeito Doppler e baixa potência de RF (BOR; ROEDIG, 2017).

3.2.1 Parâmetros Regionais LoRa

Os transceptores LoRa disponíveis atualmente podem operar entre 130 e 1020 MHz, e assim também podem operar em faixas licenciadas (BOR; ROEDIG, 2017). A camada física LoRa foi desenvolvida para operação na banda de aplicações industriais, científicas e médicas (*industrial, scientific and medical* - ISM) (UE: 868 MHz e 433 MHz, EUA: 915 MHz e 433 MHz). Pela Especificação de Parâmetros Regionais LoRa (COMITTEE, 2017), no Brasil, as redes LoRa se encaixa na faixa definida como US de 902-928 MHz.

Contudo, a faixa de frequências de 907,5 a 915 MHz é destinada primariamente ao Serviço de Comunicação Multimídia (SCM) e ao Serviço Móvel Pessoal (SMP) segundo a Resolução nº 454/2006 da Agência Nacional de Telecomunicações (ANATEL, 2006). Devido à natureza não exclusiva da atribuição, a faixa ISM ainda pode ser considerada como sendo de 902 a 928 MHz. Porém, equipamentos que façam uso da faixa de 907,5 a 915 MHz estão sujeitos a interferência adicional causadas pelo SMP e SMC.

3.2.2 Parâmetros de Transmissão LoRa

De acordo com (BOR; ROEDIG, 2017), um dispositivo LoRa pode ser configurado para usar diferentes potências de transmissão (*transmission power* - TP), frequência portadora (*carrier frequency* - CF), fator de espalhamento (*spreading factor* - SF), largura de banda (*band-width* - BW) e taxa de codificação (*coding rate* - CR) para sintonizar o desempenho do *link* e o consumo de energia.

- **Potências de Transmissão (TP):** Potências de transmissão em um rádio LoRa pode ser ajustada de -4 a 20 dBm, em passos de 1 dBm, mas devido aos limites de implementação de hardware, a faixa é frequentemente limitada de 2 a 20 dBm. Além disso, devido a limitações de hardware, níveis de potência superiores a 17 dBm só podem ser usados em um *duty cycle* de 1%;
- **Frequência Portadora (CF):** Frequência Portadora CF é a frequência central que pode ser programado em incrementos de 61 Hz compreendidos entre 137 e 1020 MHz. Dependendo do dispositivo transceptor LoRa específico, esta faixa pode ser limitada de 860 a 1020 MHz;
- **Largura de Banda (BW):** Este parâmetro diz respeito à largura das frequências na banda de transmissão. Uma largura de banda maior resulta em maior taxa de dados (portanto, menor tempo no ar), mas uma menor sensibilidade, por causa da integração de ruído adicional. Uma largura de banda menor resulta em maior sensibilidade, mas uma taxa de dados menor. Larguras de banda inferiores também requerem cristais osciladores mais precisos no hardware (menor *ppm* (partes por milhão) de incerteza). Os dados são enviados a uma taxa de *chip* igual à largura de banda; uma largura de banda de 125 kHz corresponde a uma taxa de *chip* de 125 kcps. Embora a largura de banda possa ser selecionada na faixa de 7,8 a 500 kHz, uma rede LoRa típica opera a 500 kHz, 250 kHz ou 125 kHz;
- **Taxa de Codificação (CR):** A taxa de codificação é a taxa FEC referente à correção de erros em avanço utilizada pelo modem LoRa, que oferece proteção contra rajadas

de interferência, assim aumentando a robustez do sinal transmitido. O CR pode ser ajustado para 4/5, 4/6, 4/7 ou 4/8. Um CR mais alto oferece mais proteção, mas aumenta o tempo no ar. Rádios com CR diferentes (e o mesmo CF, SF e BW, ainda podem comunicar entre si se usarem um cabeçalho explícito, pois o CR da carga útil é armazenado no cabeçalho do pacote, que é sempre codificado com CR 4/8;

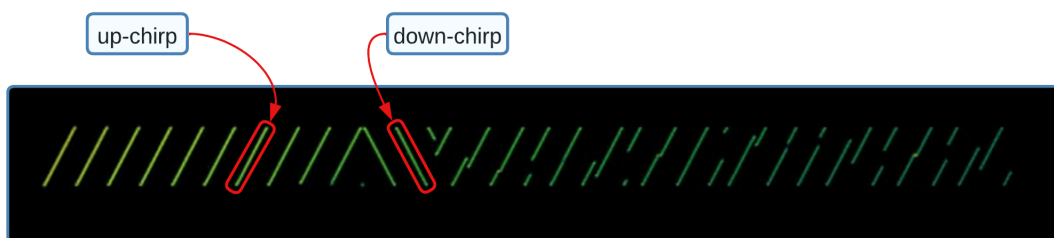
- **Fator de Espalhamento (SF):** Fator de espalhamento é a relação entre a taxa de símbolo e taxa de *chip*. Um fator de espalhamento mais alto aumenta a Relação Sinal/Ruído (*Signal to Noise Ratio* - SNR) e, portanto, a sensibilidade e o alcance, além de aumentar o tempo de transporte do pacote. O fator de espalhamento pode ser selecionado entre 6 a 12. Comunicações de rádio com SF diferentes são ortogonais entre si e é possível a separação da rede usando através da utilização de diferentes valores de SF (BOR; VIDLER; ROEDIG, 2016).

3.2.3 Modulação *Chirp Spread Spectrum* (CSS)

O protocolo LoRa emprega a técnica de modulação *chirp spread spectrum*. Esta técnica é utilizada na comunicação militar e espacial há muito tempo devido a sua robustez e capacidade de longo alcance e agora vem sendo usada comercialmente na comunicação LoRa (DEVALAL; KARTHIKEYAN, 2018).

O *chirp spread spectrum* demanda baixa necessidade de potência de transmissão. A modulação é de espectro espalhado e é uma derivação da modulação *Chirp Spread Spectrum* (CSS) que implementa taxa de dados variável através da utilização de diferentes fatores de espalhamento ortogonais em uma largura de banda fixa (SEMTECH, 2015). Os *chirps* gerados na modulação LoRa são sinais que variam continuamente com a frequência, mas mantêm a continuidade de fase constante ao longo da transmissão. Os sinais *chirp* podem ser *up-chirps* ou *down-chirps*, como pode ser observado na Figura 3, que apresenta o processo de modulação *chirp spread spectrum*.

Figura 3: Transmissão LoRa.



Fonte: Autor.

O número de bits codificados em um símbolo é um parâmetro ajustável, o Fator de Espalhamento (SF). Isto significa que um *chirp* usando o fator de espalhamento SF pode representar 2^{SF} bits por símbolo, o que significa que um símbolo é representado por múltiplos *chips* que são pulsos de um código de espectro espalhado.

$$SF = \frac{\text{taxa de chip}}{\text{taxa de símbolo}} \quad [\text{chip/símbolo}] \quad (1)$$

Como apresentado em (SEMTECH, 2015), é possível obter a taxa de bits, símbolos e chips da modulação LoRa. Inicialmente, a taxa de bits pode ser definida pela Equação 2 abaixo, em bits por segundo (*bps*).

$$R_b = SF * \frac{1}{\left[\frac{2^{SF}}{BW}\right]} \quad [bits/s] \quad (2)$$

Da Equação 2, podem ser definidos o período de símbolo (T_s) em segundos e, analogamente, a taxa de símbolo (R_s) em símbolos por segundo (*symbols/sec*). As Equações 3 e 4 abaixo representam T_s e R_s , respectivamente.

$$T_s = \frac{2^{SF}}{BW} \quad [s] \quad (3)$$

$$R_s = \frac{1}{T_s} = \frac{BW}{2^{SF}} \quad [simbolos/s] \quad (4)$$

Das equações 3 e 4 observa-se que cada símbolo, composto de SF bits possui um total de 2^{SF} chips que ocupam uma largura de banda BW . Seguindo o desenvolvimento matemático proposto por (SEMTECH, 2015), é possível derivar a taxa de chips. A Equação 5 representa a derivação da taxa de chips (R_c) da modulação LoRa.

$$R_c = R_s * 2^{SF} = \frac{BW}{2^{SF}} * 2^{SF} = BW \quad [chips/s] \quad (5)$$

A partir da Equação 5 percebe-se então que a modulação LoRa espalha os dados a uma taxa de chips igual à largura de banda escolhida. Assim, afirma-se que um chip é enviado por segundo por Hertz de banda. Portanto, uma largura de banda de 250 kHz, por exemplo, equivale a uma taxa de chips de 250 kcps. Finalmente, a modulação inclui ainda métodos de FEC a fim de fornecer maior robustez ao sinal transmitido (SEMTECH, 2015). Desse modo, para uma dada taxa de código CR tem-se que a taxa de bits nominal é dada pela Equação 6 abaixo.

$$R_b = SF * \left(\frac{4}{\frac{4+CF}{\frac{2^{BF}}{BW}}} \right) \quad [bsp] \quad (6)$$

3.3 Protocolos de Comunicação SPI e I²C

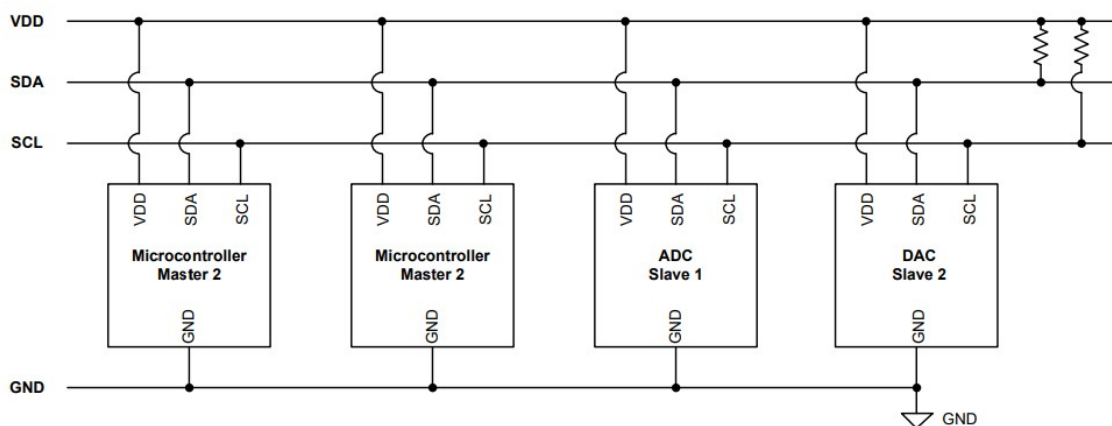
De acordo com (LEENS, 2009b), na extremidade de mais baixo nível dos protocolos de comunicação encontram-se os protocolos de circuito inter-integrado (*Inter-Integrated Circuit*, I²C) e de interface periférica serial (*Serial peripheral interface*, SPI). Ambos os protocolos são bem adequados para comunicações entre circuitos integrados para comunicação lenta com periféricos embarcados. Os dois protocolos coexistem nos sistemas eletrônicos digitais modernos.

3.3.1 Protocolo de comunicação I²C

O protocolo circuito inter-integrado (I²C), desenvolvido pela *Philips* em 1982 (WU, 2021), é um barramento único, serial, multi-mestre. O barramento mostrado na Figura 4 é formado a partir de duas linhas bidirecionais entre dispositivos mestre e escravo. Estas linhas são conectadas em uma fonte de tensão positiva por resistores de *pull-up*, como representado na Figura 4. As duas linhas do barramento I²C representado na Figura 4 desempenham a seguinte função:

- *Serial Clock (SCL)*: Linha usada pelo dispositivo mestre para sincronicamente temporizar a entrada e saída da informação para/dos dispositivos escravo;
- *Serial Data (SDA)*: Linha usada para a transmissão de informação serial entre dispositivos mestres e escravos conectados. Tornando assim, o I²C *half-duplex*.

Figura 4: Exemplo de camada física I²C.

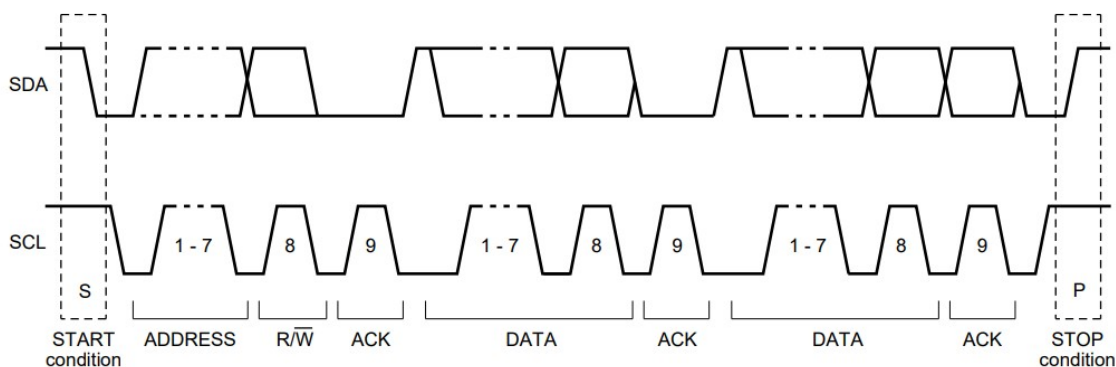


Fonte: (WU, 2021).

Já que os barramentos são completamente compartilhados entre os dispositivos, sempre que o mestre tiver a necessidade de ler ou escrever em algum dispositivo escravo ele irá endereçá-lo especificamente. Esta primeira ação de endereçamento de dispositivo que se deseja comunicar é feita originalmente através de sete bits, representados na Figura 5 pelos bits ADDRESS e na Figura 6 pelo TARGET ADDRESS.

Quando o mestre endereça um escravo, o mestre aguardará uma mensagem de reconhecimento (*acknowledge* - ACK) do dispositivo escravo para que possa se dar prosseguimento a operação desejada. Este bit de reconhecimento pode ser observado nas Figuras 5 e 6 logo após o bit R/\overline{W} , o qual é responsável por indicar se a operação será de leitura ou escrita (*read* ou *write*).

Figura 5: Exemplo de transferência de dado I²C.

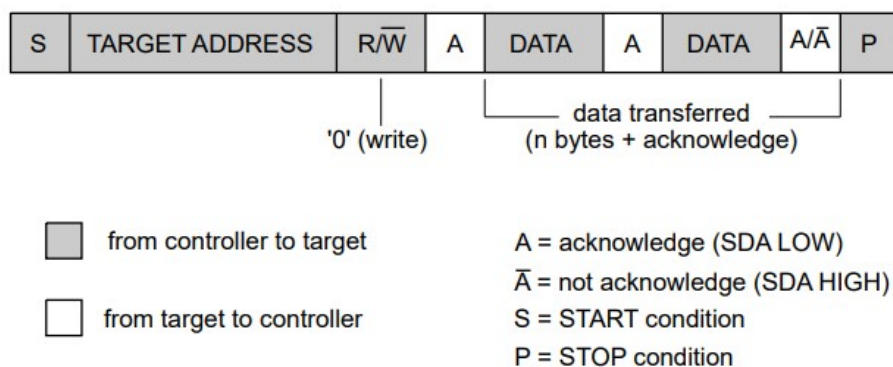


Fonte: (SEMICONDUCTORS, 2014).

Esta relação de ACK entre mestre e escravo não se limita ao endereçamento dos dispositivos controlados, ela é utilizada para verificar o resultado da transmissão de cada byte.

Logo, de acordo com a estrutura de envio especificada em (SEMICONDUCTORS, 2014) e representada nas Figuras 5 e 6, após o *start bit*, Figura 6 como S - *START condition*, bit responsável por sinalizar o início da transmissão e endereçamento do dispositivo escravo, o mestre define o modo de operação de escrita ou leitura (R/\overline{W}). Assim, após receber um bit de *ACK* do dispositivo endereçado, uma sessão de transmissão será iniciada com mensagens de *ACK* ou *NACK* (Não reconhecido - *Not Acknowledge*) enviadas pelo receptor após cada byte transmitido pelo controlador. O *end bit*, o qual pode ser verificado na Figura 6 como P - *STOP condition*, sinalizará que a transmissão se encerrou.

Figura 6: Um controlador-transmissor endereçando um receptor com um endereço de 7 bits.



Fonte:

(SEMICONDUCTORS, 2014).

De acordo com (LEENS, 2009b), a taxa de dados tem de ser escolhida entre 100 kb/s, 400 kb/s, e 3,4 Mb/s, respectivamente, chamado modo padrão (*standard mode*), modo rápido (*fast mode*), e modo de alta velocidade (*high speed mode*). Algumas variantes I²C incluem 10 kb/s (*low speed mode*) e 1 Mb/s (*fast mode +*) como velocidades possíveis.

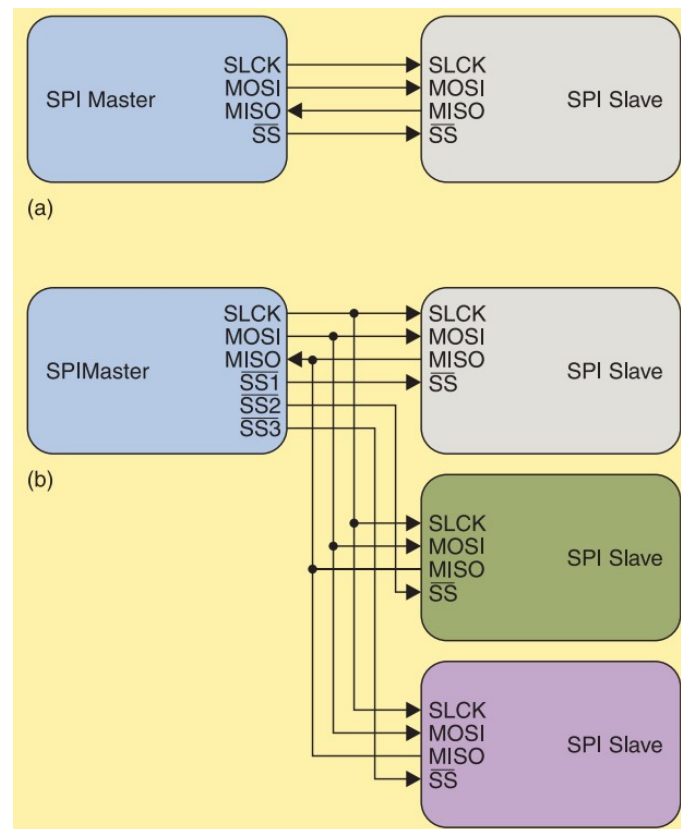
3.3.2 Protocolo de comunicação SPI

Segundo (LEENS, 2009a), o SPI foi introduzido com o primeiro microcontrolador que derivou da mesma arquitetura que o popular microprocessador Motorola 68000, anunciado em 1979.

O módulo SPI permite uma comunicação serial *full-duplex*, síncrona, entre o microcontrolador e dispositivos periféricos. O protocolo é composto por 4 sinais, representados na Figura 7, dentre eles:

- *Serial Clock* (SCLK): Um sinal de relógio enviado pelo mestre a todos os escravos. Todos os sinais do SPI são sincronizados com este sinal do relógio e ele controla quando os dados são enviados e quando são lidos;
- *Slave Select* (SS_n): Para cada dispositivo escravo presente, este sinal é usado para selecionar o escravo com o qual o mestre se comunicará. Quando o sinal se torna baixo, o dispositivo escravo escuta o relógio SPI e os sinais de dados;
- *Master Out-Slave In* (MOSI): Sinal gerado pelo mestre direcionado ao receptor (escravo);
- *Master In-Slave Out* (MISO): O sinal gerado pelo escravo, em que o receptor é o mestre;

Figura 7: (a) Topologia SPI ponto-a-ponto. Um mestre conectado a um único dispositivo escravo. (b) Topologia de um mestre conectado a múltiplos dispositivos escravos.

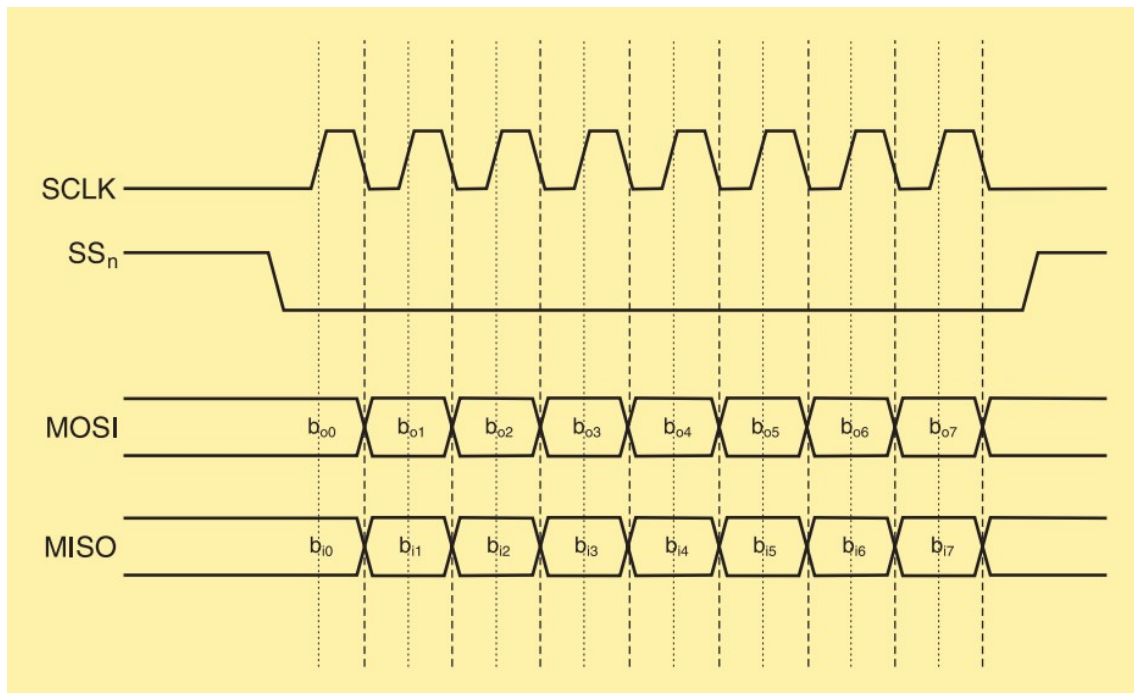


Fonte: (LEENS, 2009a).

O elemento principal deste sistema é o registrador de dados do SPI (SEMICONDUCTORS, 2004). O registrador de dados de 8 bits no mestre e o registrador de dados de 8 bits no escravo são ligados pelos pinos MOSI e MISO para formar um registrador distribuído de 16 bits. Quando uma operação de transferência de dados é realizada, este registro de 16 bits é deslocado em série de oito bits pelo relógio SCLK do mestre, de modo que os dados são trocados entre o mestre e o escravo. Os dados escritos no registrador de dados SPI do mestre tornam-se os dados de saída para o escravo, e os dados lidos do registrador de dados SPI mestre após uma operação de transferência são os dados de entrada do escravo.

Logo, o SPI é um protocolo de comunicação de um único mestre. Isto significa que um dispositivo central inicia todas as comunicações com os escravos. Quando o mestre SPI deseja enviar dados a um escravo e/ou solicitar informações dele, ele seleciona um escravo puxando a linha SS correspondente para um nível baixo, e ativa o sinal do relógio SCLK em uma frequência utilizável pelo mestre e pelo escravo. O mestre gera informações sobre a linha MOSI enquanto ele recolhe amostras da linha MISO, como pode ser visto na Figura 8.

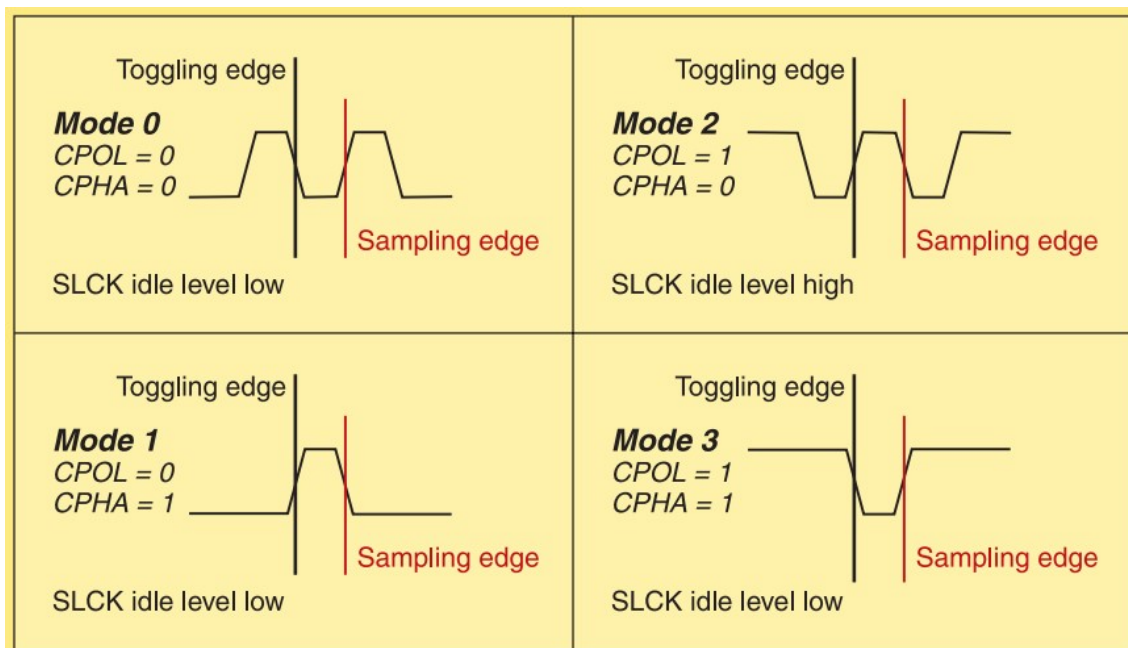
Figura 8: Transmissão comum de protocolo SPI.



Fonte: (LEENS, 2009a).

Além disso, deve-se levar em consideração o bit de controle de fase do relógio (*clock phase control*, CPHA) e o bit de controle de polaridade do relógio (*clock polarity control*, CPOL) ao lidar com o protocolo SPI. Através da manipulação destes bits, um dos quatro formatos de relógio possíveis pode ser aplicado ao sistema SPI, conforme mostrado na Figura 9.

Figura 9: Configurações de clock para o protocolo SPI.



Fonte: (LEENS, 2009a).

O bit CPOL seleciona se o formato de onda do sinal de relógio SCLK será invertido ou não invertido. Já O bit CPHA é usado para acomodar dois protocolos fundamentalmente diferentes através da amostragem de dados em bordas SCLK ímpares (1,3,5,...,15) ou em bordas SCLK pares (2,4,6,...,16). A disposição dos bits CPOL e CPHA no registrador de controle do protocolo SPI podem ser observados na Figura 10.

Figura 10: Registrador de controle do protocolo SPI.

	Bit 7	6	5	4	3	2	1	Bit 0
R	SPIE	SPE	SPTIE	MSTR	CPOL	CPHA	SSOE	LSBFE
W								
Reset:	0	0	0	0	0	1	0	0

Fonte: (SEMICONDUCTORS, 2004).

De acordo com (LEENS, 2009a), o protocolo SPI não define nenhuma taxa máxima de dados, as implementações muitas vezes ultrapassam 10 Mb/s. Não há um esquema de endereçamento específico; não tem um mecanismo de confirmação de recebimento de dados e não oferece nenhum controle de fluxo. O mestre no protocolo SPI não tem conhecimento da existência de um escravo, a menos que algo adicional seja feito fora do protocolo SPI.

3.4 Filtro Média Móvel

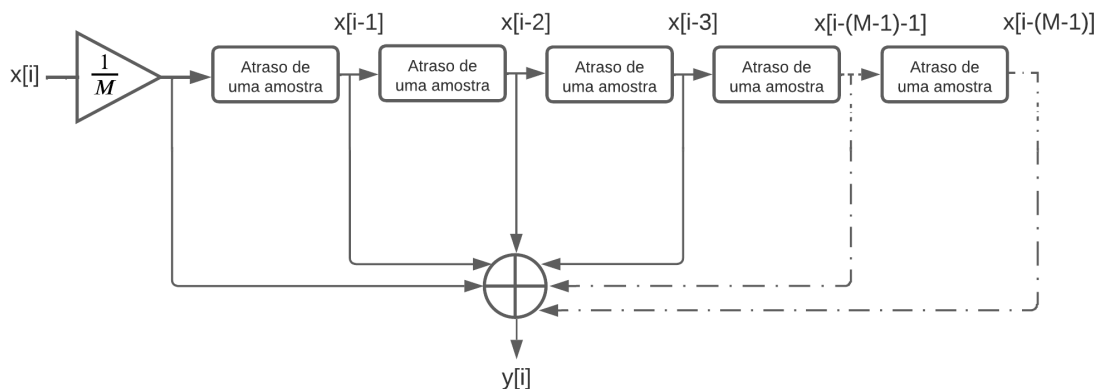
O filtro média móvel (*moving average filters* - MAF) é o filtro mais comum em processamento digital de sinais, sobretudo porque é o filtro digital mais fácil de se aplicar. Em sua simplicidade, o filtro de média móvel é ideal para uma tarefa comum: reduzir o

ruído aleatório enquanto mantém uma acentuada resposta ao degrau embora sua resposta no domínio frequência não seja tão boa quanto outras topologias de filtros passa-baixa disponíveis, tais como o *Gaussian* ou o de *Blackman* (SMITH et al., 1997).

Tal filtro pode ser observado na Equação 7.

$$y[i] = \frac{1}{M} \sum_{j=0}^{M-1} x[i - j] \quad (7)$$

Figura 11: Diagrama do funcionamento do filtro média móvel.



Fonte: Autor.

Em que y é o sinal de saída, x o sinal de entrada, M o número de pontos considerados na média e i o número da iteração atual da média.

3.5 Unix Time

A hora Unix é uma forma de representar um horário ao representar o tempo como o número de segundos desde 1º de janeiro de 1970 às 00:00:00 UTC. Um dos principais benefícios do uso do tempo Unix é que ele pode ser representado como um inteiro, facilitando a análise e o uso em diferentes sistemas (NARRATIVE, s.d.).

3.5.1 Problema do ano 2038

O problema do ano 2038 está relacionado com o tempo Unix, pois os horários após as 03:14:07 UTC de 19 de janeiro de 2038 exigirão que os computadores armazenem o valor como superior a 32 bits. Os sistemas computacionais foram construídos assumindo que o tempo sempre se encaixaria com 32 bits, o que resultará em um comportamento indefinido, causando o travamento, a inutilização ou a criação de outros efeitos indesejáveis.

4 MÉTODOS E MATERIAIS

Nesta seção são apresentados os métodos e materiais utilizados neste trabalho. Os elementos de hardware software, utilizados ou desenvolvidos, são descritos.

4.1 Elementos de Hardware

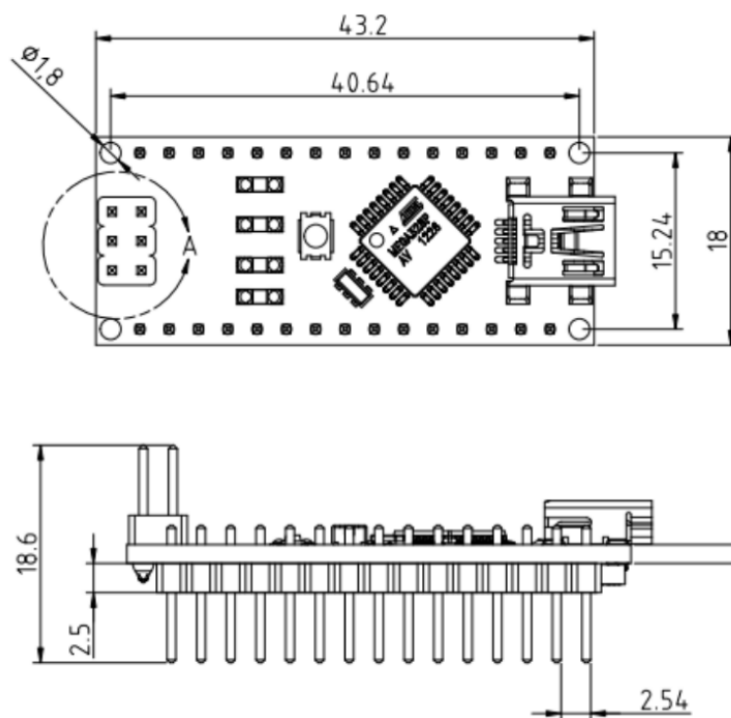
4.1.1 Microcontrolador

Para realizar o processos de controle e comunicação entre dispositivos foi selecionado o microcontrolador de 8-bit ATmega328P. Este microcontrolador, projetado na arquitetura RISC opera entre 1,8-5,5 volts, apresenta baixo consumo de energia e opera a 20 MHz, sendo capaz de realizar 20 Milhões de Instruções Por Segundo (MIPS)(MICROCHIP, 2020). Além disso, o ATmega328P combina memória Flash ISP de 32 kB com capacidades de leitura-escrita e disponibiliza de 23 pinos de I/O programáveis, um conversor analógico-digital de 6 canais de 10 bits e compatibilidade com interfaces do tipo I²C e SPI para a intercomunicação entre dispositivos conectados (MICROCHIP, 2020).

Para empregar este microcontrolador no projeto foi decidido utiliza-lo através do circuito Arduino Nano v3.3, o qual possui um ATmega328P embarcado. A placa oferece 22 pinos de entrada/saída digitais, 8 pinos analógicos (ARDUINO, 2022).

Além das vantagens mencionadas, como o baixo consumo de energia, característica importante para protocolos de comunicação LPWA, o baixo custo de aquisição e características físicas do Arduino Nano foram fatores relevantes para a sua escolha, pois para esta aplicação LoRa, havia necessidade de um projeto compacto com pouco peso, em função da aplicação inicial (monitoramento de animais). Com um comprimento de 43,2 mm e uma largura de 18 mm , além de pesar apenas 7 gramas, o Arduino Nano atendeu os requisitos iniciais. As dimensões do Arduino Nano podem ser observadas na Figura 12.

Figura 12: Dimensões da placa Arduino Nano.

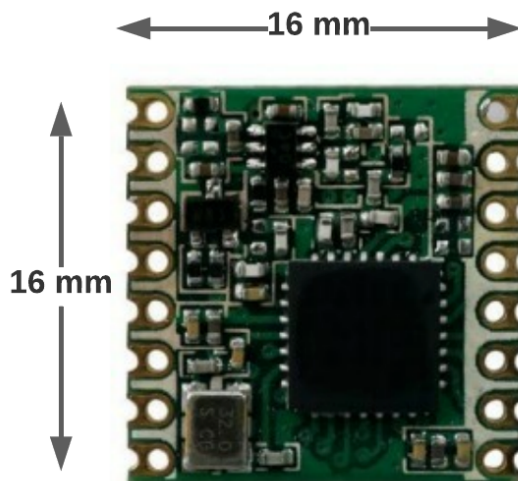


Fonte: Adaptado de (ARDUINO, 2022).

4.1.2 Módulo LoRa

Para este projeto foi empregado o transceptor LoRa RMF95W. Este módulo utiliza o controlador SX1276 e, para esta aplicação, opera na faixa de transmissão ISM americana entre 868 e 915 MHz, apresentando sensibilidade de enlace entre -111 e -148 dBm, conforme a largura de banda utilizada, e taxa de dados de até 37,5 kbps (SEMTECH, s.d.).

Figura 13: Módulo transceptor LoRa RMF95W.



Fonte: (HOPERF, 2020).

4.1.3 Real Time Clock

Para realizar a sincronização na comunicação entre os nodos LoRa, assim como para obter a estampa temporal, foi utilizado um *real time clock* (RTC) no projeto do *hardware*. Foi selecionado o componente DS1307 como RTC do projeto, por se tratar um componente barato, de baixo consumo energético e de ampla utilização.

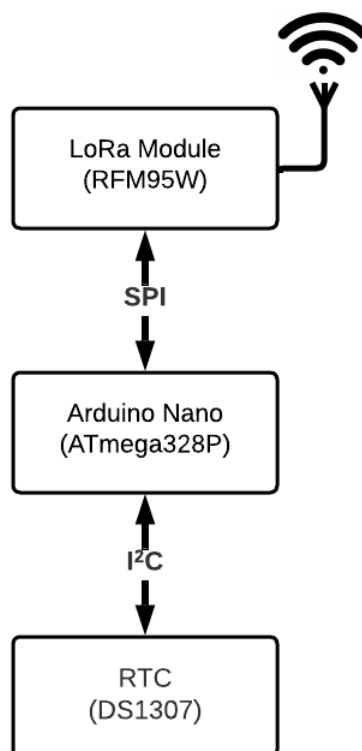
Como pode ser observado em (SEMICONDUCTOR, 2020), o DS1307 é um relógio-calendário decimal completamente codificado em binário, de baixa potência, consumindo menos que 500 nA em modo de *BatteryBackup*, com oscilador em operação. O DS1307 opera como um dispositivo escravo no barramento I^2C , em que seu endereço e dados são transferidos em série através de deste barramento de forma bidirecional. O relógio/calendário fornece informações sobre segundos, minutos, horas, dia, data, mês e ano.

O bit 7 do registrador de configuração do RTC é o bit de parada (*halt*) do relógio. Quando este bit é ajustado para 1, o oscilador é desativado. Quando ajustado para 0, o oscilador é habilitado.

4.1.4 Dispositivo Completo

Tendo especificados os elementos de hardware nas seções precedentes, torna-se necessário uni-los em um dispositivo único para formar o nodo LoRa, que será o componente básico de implementação deste projeto. Na Figura 14, observa-se o diagrama do dispositivo completo. Para as análises e testes do *firmware* desta aplicação, apoiou-se no protótipo previamente produzido no trabalho de (VALADÃO, 2021), utilizado para o sensoriamento do comportamento reprodutivo de bovinos.

Figura 14: Diagrama do hardware do nodo LoRa.



Fonte: Autor.

4.2 Implementação do *Firmware*

O projeto se deu início ao avaliar a necessidade de se inicializar a rede de nodos transceptores LoRa temporalmente sincronizada sem ter que ajustar o RTC de cada nodo, conectando a um computador para configurar os dispositivos. Logo, para o desenvolvimento desta rede LoRa, foi identificado como ponto de partida, trabalhar no *firmware* do nodo coletor de modo a proporcionar ao nodo sensor, identificar uma mensagem conhecida como *advertisement* (ADV), a ser enviada em um tempo determinado. Desta forma, foram inicialmente estipulados os parâmetros a serem usados na transmissão entre nodo coletor e nodo sensor. Foram definidos os pinos de interface entre o Arduino Nano e a módulo LoRa RFM95W, assim como frequência portadora 915 MHz e um fator de espalhamento equivalente a 8, o que confere uma transferência de 8-bits por chip, como apresentado na Seção 3.2.3. Posteriormente, com o intuito de dar andamento a contagem do tempo, foi inicializado o RTC do nodo coletor.

Com o intuito de manter o tempo do coletor o mais próximo possível do tempo real, o RTC passou a ser inicializado com o tempo e data de compilação do *firmware* do coletor no Arduino Nano. Apesar de apresentar vantagens para implementações futuras, a ajustar o tempo do RTC conforme o tempo real, para esta aplicação a rede pode operar no seu próprio tempo.

Tendo configurado os parâmetros de transmissão do módulo LoRa e o RTC, foi possível realizar transmissões de *advertisements* em tempos pré-determinados. Em uma primeira instancia, o coletor passou a transmitir suas mensagens a cada 10 segundos. Por outro

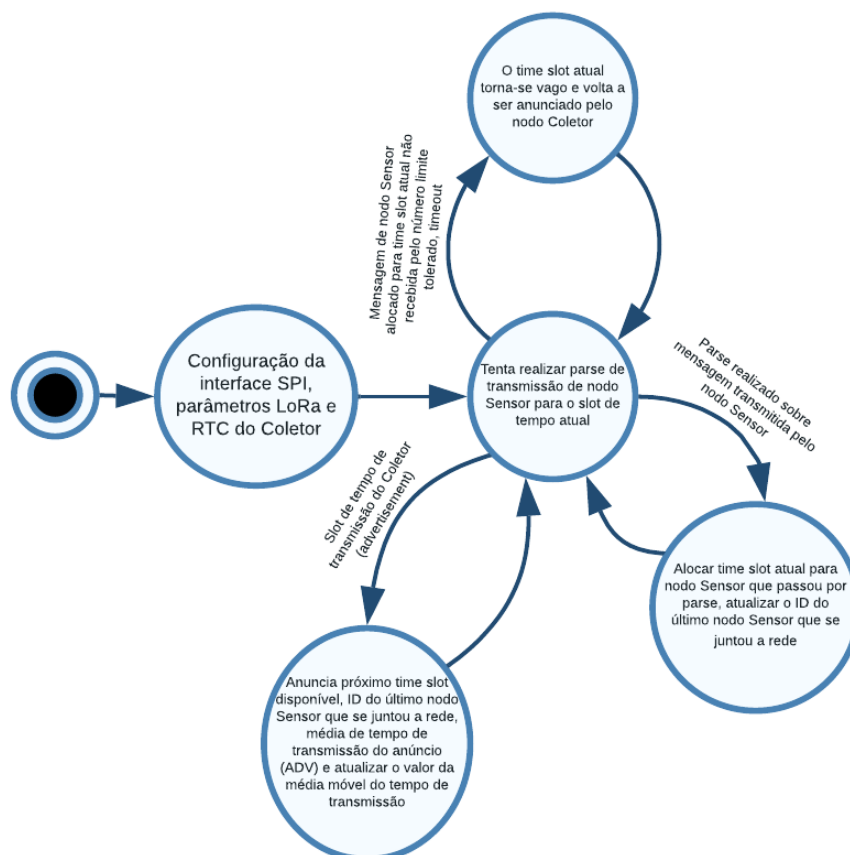
lado, no nodo sensor, para que o sinal de *advertisement* transmitido pelo coletor fosse identificado, os mesmos parâmetros LoRa teriam que se ser estipulados, ou seja, mesma frequência de portadora, mesmo fator de espalhamento e mesma taxa de codificação.

Após ter definido os parâmetros LoRa se da início ao processo de *parse* do sinal LoRa de *advertisement*. Nesta etapa o nodo sensor fica em estado contínuo de detecção de um sinal que possua os mesmos parâmetros LoRa previamente definidos. No entanto, um sinal de uma outra rede LoRa configurada com os mesmos parâmetros passará pelo *parse*, o que é indesejado. Para mitigar este problema, foi definido que mensagens com tamanho de pacote diferentes dos nodos sensores da rede recebidos pelo coletor, seriam ignorados. O mesmo foi feito para os nodos sensores em relação as transmissões de *advertisement*, mensagens com tamanho de pacote diferentes do nodo coletor da rede recebido pelos nodo sensores seriam ignorados. Após estas definições, foi possível realizar a primeira iteração do desenvolvimento, convertendo o nodo sensor de um receptor para um transmissor ao receber uma mensagem de *advertisement* do nodo coletor.

4.2.1 Nodo coletor LoRa

O funcionamento completo do *firmware* implementado para o nodo coletor pode ser observado na Figura 15.

Figura 15: Diagrama de estados do nodo coletor LoRa.



Fonte: Autor.

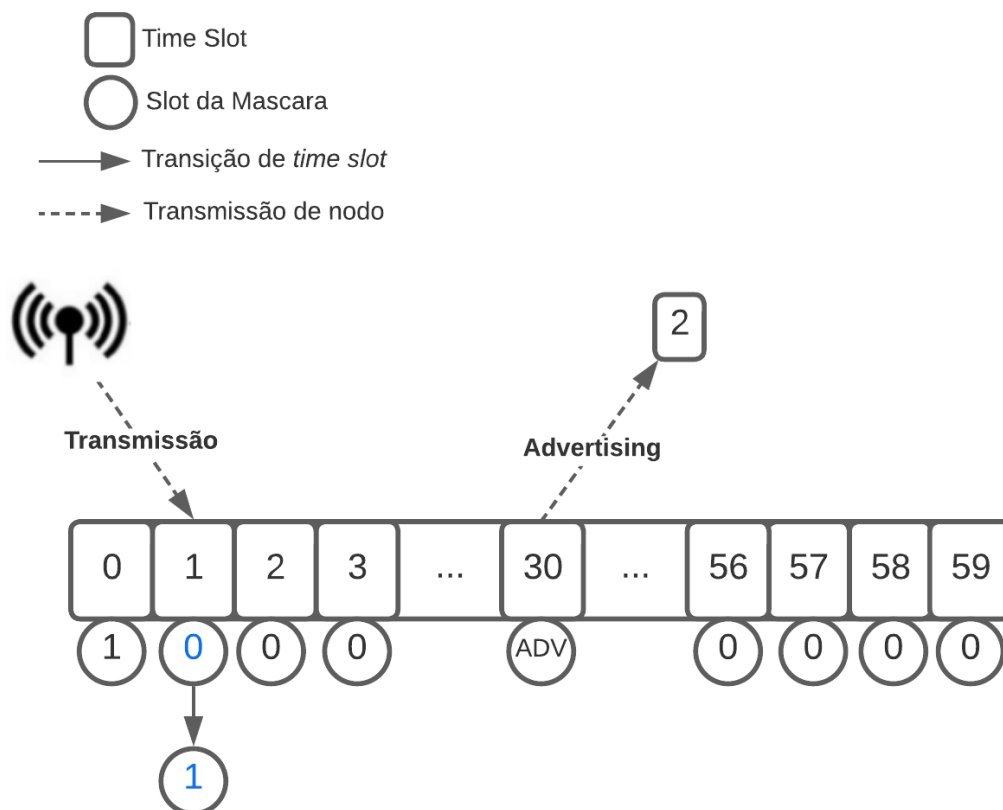
4.2.1.1 Gerenciador de *time slot*

A base da sincronização das transmissões foi feita a nível dos segundos. Isto quer dizer que cada segundo é equivalente a uma posição elegível a transmissão ou recepção de sinal, assim chamada de *time slot*.

Logo, foi preciso uma estrutura para gerenciar quais nodos sensores estariam transmitindo ativamente na rede e à qual *time slot* estariam relacionados, assim como saber quais *time slots* estariam disponíveis para serem anunciados. Para isso, foi criada uma máscara no nodo coletor que relaciona o ID do nodo sensor, definido na Seção 4.2.2.1, com a posição do *time slot*. Para essa implementação em questão, foi usada uma quantidade de *time slots* disponíveis equivalente a 60, sendo uma pertencente a mensagem de *advertisement* do nodo coletor, e as demais as transmissões dos nodos sensores.

Assim, quando um nodo se junta a rede com sucesso, o *time slot* desta máscara equivalente ao *time slot* que anteriormente estava sendo anunciado é preenchido, assim deixando de ser anunciado e abrindo espaço para o anúncio do próximo *time slot*, como demonstrado na Figura 16.

Figura 16: Diagrama do gerenciador de *time slots* do nodo coletor.



Fonte: Autor.

Na Figura 16 o *time slot* anteriormente anunciado foi o 1, o qual foi recebido pelo nodo sensor em questão que passou a enviar com sucesso para esta posição no tempo. Sendo assim, a posição 1 que antes estava disponível para ser anunciada pelo coletor, se

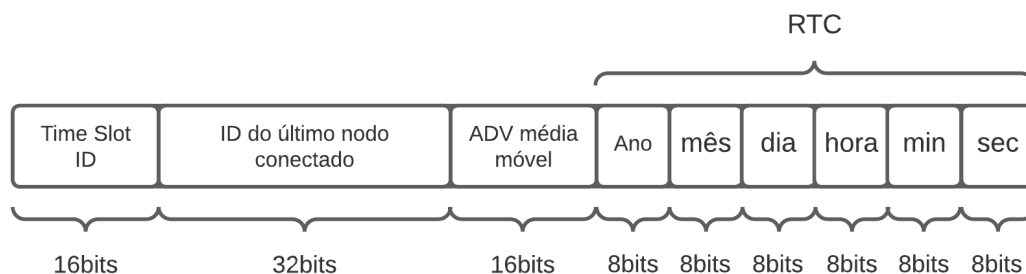
torna indisponível através da transição de 0 para 1 na máscara de *time slots*. Após o *time slot* 1 ter sido ocupado, o próximo na sequência passa a ser anunciado, no caso da Figura 16 sendo o *time slot* 2.

4.2.1.2 Conteúdo do pacote de *advertisement*

Após a conclusão da primeira iteração do desenvolvimento, viu-se a necessidade de definir as informações à serem enviadas para o nodos sensor pelo nodo coletor para que ele pudesse se configurar na sua inicialização. Assim foram definidos os seguintes parâmetros à serem enviados para os nodos sensores em mensagem de *advertisement* do nodo coletor:

1. *ID* do próximo *time slot* alocável;
2. *ID* do último nodo que conectou à rede;
3. Média móvel do tempo de transmissão da mensagem de *advertisement*;
4. Data e tempo registrados pelo RTC.

Figura 17: Diagrama do conteúdo do pacote LoRa de *advertisement* de um nodo coletor.



Fonte: Autor.

Em relação ao *time slot* ID representado na Figura 17, este é definido pelo nodo coletor no gerenciador de *time slots*, especificado na Seção 4.2.1.1, e anunciado para a rede de sensores sem fio para que os nodos sensores que estão em processo de agregação (nodos em processo de tentar realiza *parse* de ADV e não alocados à nenhum *time slot* da rede) saibam qual é o próximo *time slot* que poderão pleitear para entrar na rede e passar a transmitir seus dados sincronicamente ao nodo coletor.

Os últimos 6 bytes enviados correspondem ao tempo e data do RTC do nodo coletor. Em relação ao sincronismo da rede de sensores sem fio, RTC do nodo coletor atua como um relógio mestre, sendo a partir dele que todos os outros nodos sensores realizarão o seu sincronismo de transmissões e saberão o momento em que devem transmitir. Assim, os parâmetros de tempo e data do RTC do nodo coletor ao serem enviados no pacote de

advertisement servem como condições de inicialização para o RTC dos nodos sensores que conseguiram alocar um *time slot*, e por conseguinte se juntar a rede com sucesso, para passar a realizar a transmissão de seus dados.

O *ID* do último nodo conectado é necessário para realizar a etapa final do processo de validação da entrada de um nodo sensor à rede. Esta lógica de validação é mais profundamente apresentada na Seção 4.2.2.4. A média móvel do tempo de transmissão é utilizada na correção de sincronismo, e explicada com mais detalhes na seção 5.1.

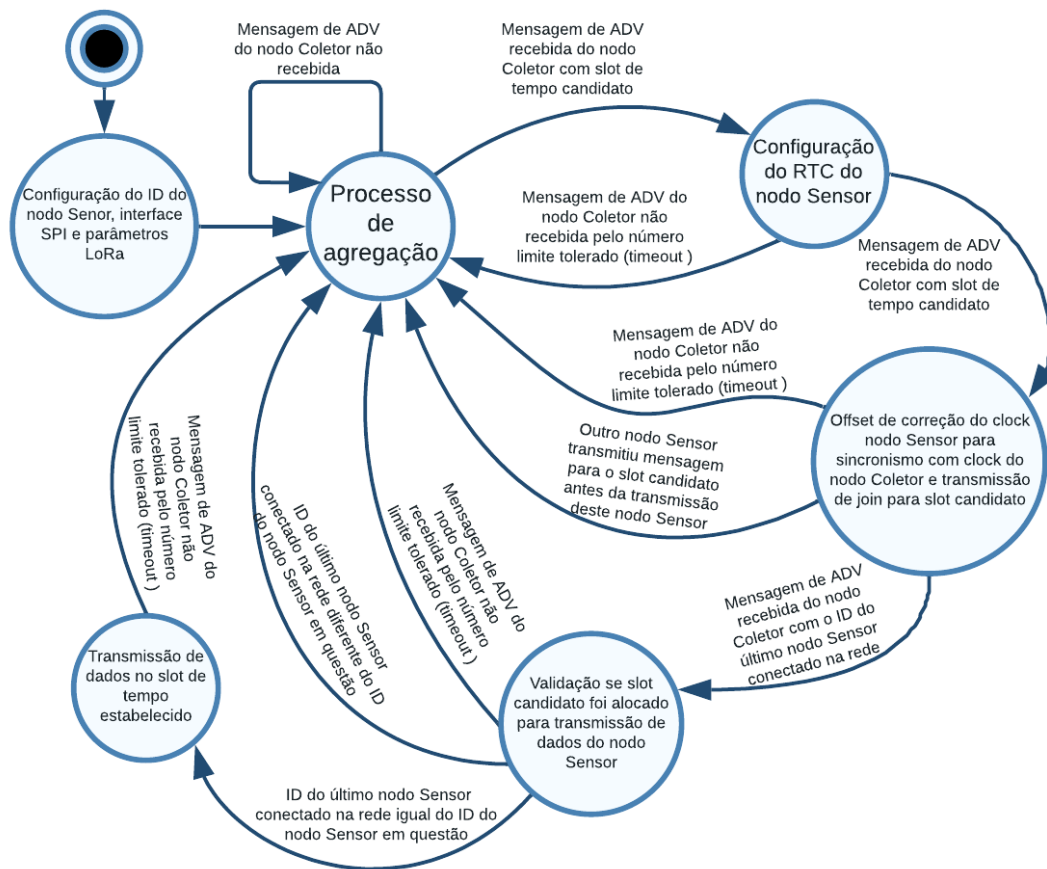
4.2.1.3 *Timeout: Nodo coletor*

Tendo a rede de sensores sem fio operante, é importante que haja um mecanismo que possa lidar com os *time slots* que não estão mais recebendo transmissão de nodos sensores. Assim, foi posto em prática um mecanismo de contagem de *timeout* das vezes em que um *time slot* não recebeu transmissão do nodos sensor que estava previamente conectado à ele. No momento em que esse número de transmissões não recebidas ultrapassa um limite estipulado, o *time slot* é liberado para que mais uma vez possa ser anunciado em *advertisement* e para que outro nodo sensor tenha a chance de entrar na rede por esse *time slot* em questão. Também, já que nesta aplicação explorada se optou por 59 *time slots* para nodos sensores, há a necessidade de criar um contador de *timeout* para cada um dos *time slots* alocados.

4.2.2 **Nodo sensor LoRa**

O funcionamento completo do *firmware* implementado para o nodo sensor pode ser observado na Figura 18.

Figura 18: Diagrama de estados do nodo sensor LoRa.



Fonte: Autor.

A partir do momento em que se realizou a conversão de um nodo sensor de receptor para transmissor com sucesso, como mencionado na Seção 4.2, para se conseguir fechar a rede era preciso que as transmissões fossem feitas no tempo correto para que não houvesse colisão com transmissões de outros nodos sensores. Como foi definido na Seção 4.2.1.2, o RTC do nodo coletor atuaria como relógio mestre para todos os demais nodos sensores sincronizarem suas transmissões. Assim, a partir do momento em que se passou a ser enviado o tempo e data do RTC do coletor no pacote de *advertisement*, como ressaltado na Seção 4.2.1.2, passou a se configurar o RTC do nodo sensor após o *parse* da mensagem de *advertisement* com a data e a hora do RTC do nodo coletor de maneira a estarem com seus relógios sincronizados.

Após configurar o RTC do nodo sensor para operar sincronicamente com o RTC do nodo coletor, para que não ocorressem transmissões conflituosas entre nodos sensores, passou-se a realizar transmissões das informações adquiridas pelo nodo sensor em questão somente no *time slot* em que havia sido recebida a mensagem de *advertisement* do coletor, como especificado na Seção 4.2.1.1. Desta forma, em parte, as transmissões dos nodos coletores não seriam realizadas concorrentemente, consultar Seção 4.2.2.3.

4.2.2.1 Identificador único dos nodos sensores

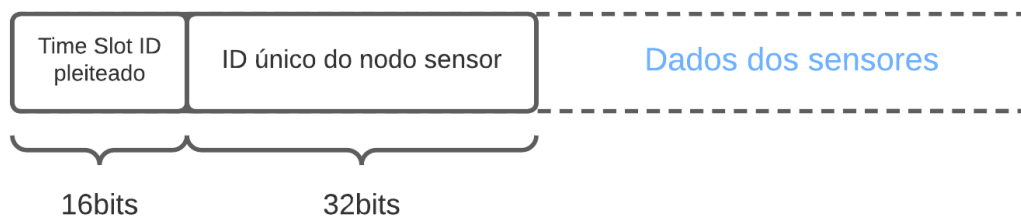
Dada a necessidade de identificar de qual nodo é proveniente os dados em uma rede de sensores sem fio, foi constatada a importância de haver um identificador (ID) único. No entanto, com os componentes utilizados para compor o hardware dos nodos, proposto na Seção 4.1, não havia um identificador único disponível por meio de hardware, logo, foi necessário obtê-lo por software. Para resolver isto, foi definido como identificador único o Unix Time, definido na Seção 3.5, do momento da compilação do *firmware* no nodo sensor. Porém, esta lógica para o ID único implica em algumas premissas:

1. Para que o ID de cada nodo sensor seja de fato único, é imprescindível que o *firmware* seja gravado em somente um nodo por vez;
2. Como abordado na Seção 3.5.1, no ano de 2038 não haverá mais ID disponíveis, visto que o valor do Unix Time será superior aos 32 bits definidos para ele. Entretanto, esta limitação se vê como aceitável.

4.2.2.2 Conteúdo do pacote de um nodo sensor

Para que o processo de conexão de um nodo sensor à rede de sensores sem fio pudesse ser realizada, o nodo coletor precisaria de algumas informações do nodo sensor que está a pleitear um *time slot* da rede para as suas transmissões. Portanto, definiu-se o pacote de transmissão do nodo sensor da seguinte como representado na Figura 19.

Figura 19: Diagrama do conteúdo do pacote LoRa de transmissão de um nodo sensor.



Fonte: Autor.

Desta forma, como visto na Figura 19, se envia o *time slot* pleiteado, recebido em *advertisement*, assim como o ID do nodo sensor, para que o nodo coletor possa saber que se deseja conectar no *time slot* correto oferecido e qual nodo sensor deseja realizar esta operação. Também, como pode ser observado na Figura 19, após as informações necessárias para que o nodo consiga se conectar sincronicamente na rede, vêm os dados dos sensores ao qual o nodo está conectado, por exemplo, temperatura, luminosidade, umidade, etc. No entanto, como mencionado no Capítulo 1, o intuito deste projeto é desenvolver uma rede de nodos sensores LoRa genérica auto-gerenciável. Logo a implementação dos sensores e sua cadeia de aquisição não cabe no escopo deste trabalho.

4.2.2.3 Offset de correção do RTC do nodo sensor

Como a rede de sensores sem fio foi definida que as transmissões devam ser sincronizadas, há a necessidade de se garantir que os relógios RTC do nodo sensor e do nodo

coletor estejam alinhados. Parte deste processo foi descrito no início da Seção 4.2.2, em que o nodo coletor, em sua mensagem de *advertisement*, envia a data e hora de seu RTC com o objetivo de servir como condições iniciais para configurar o RTC do nodo sensor que entrará na rede. No entanto, devido ao tempo de processamento das transmissões de ambos os nodos e rotinas realizadas nos periféricos dos nodos, estes RTC podem não estar com seus tempos suficientemente alinhados. Para isso é necessário um fator de correção, que neste caso se aplica um *offset* no tempo do RTC do nodo sensor para alinhá-lo com o RTC mestre do nodo coletor.

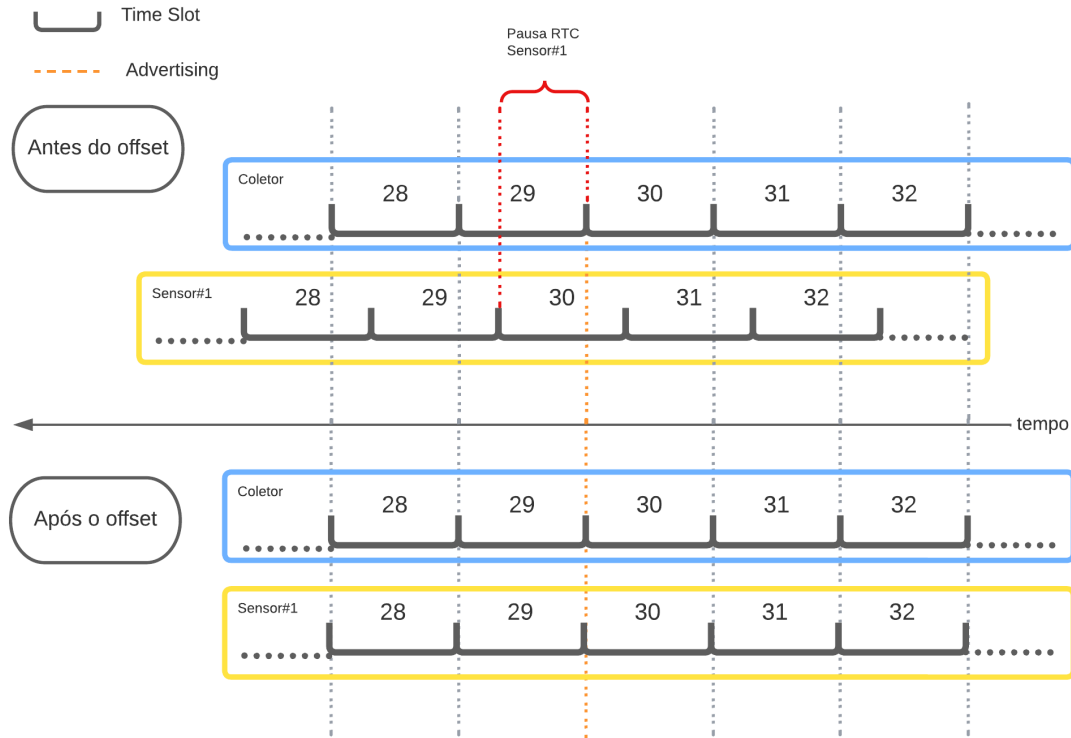
Assim, para a aplicação em questão, tem-se definido 60 *time slots* de transmissão, 59 disponíveis à serem alocados para transmissões de diferentes nodos sensores e 1 para mensagem de *advertisement* do nodo coletor, os quais operam de maneira cíclica. Com isso, utilizou-se das mensagens de *advertisement* para realizar os procedimentos de *offset* de sincronização dos nodos sensores. Desta forma, para um nodo sensor que se encontra com o seu relógio adiantado em relação ao relógio do nodo coletor, realiza-se o seguinte procedimento:

1. Mede-se a diferença do *parse* bem sucedido de um *advertisement* do nodo coletor ($ADV_{parse_{time}}$) e o tempo entre o início do segundo do RTC do nodo sensor ($ADV_{slot_{begin}}$), representado pela Equação 8 (variável: $\Delta Parse$);
2. Realiza-se uma pausa no RTC do nodo sensor através do *halt-bit*, especificado na Seção 4.1.3;
3. Executa-se uma pausa no sistema de um tempo equivalente ao resultado da Equação 9, onde $slotReceivedADVOnsensor$ é o *time slot* em que o nodo sensor recebeu o *advertisement* e $slotADV$ o *time slot* em que o coletor realiza o *advertisement* (uma constante registrada igual em todos os nodos);
4. Libera-se o RTC do nodo sensor da pausa através do *halt-bit*.

$$\Delta Parse = ADV_{parse_{time}} - ADV_{slot_{begin}} \quad (8)$$

A lógica descrita acima pode ser observada na Figura 20.

Figura 20: Diagrama de *offset* de correção para um nodo sensor com seu relógio adiantado em relação ao do nodo coletor.



Fonte: Autor.

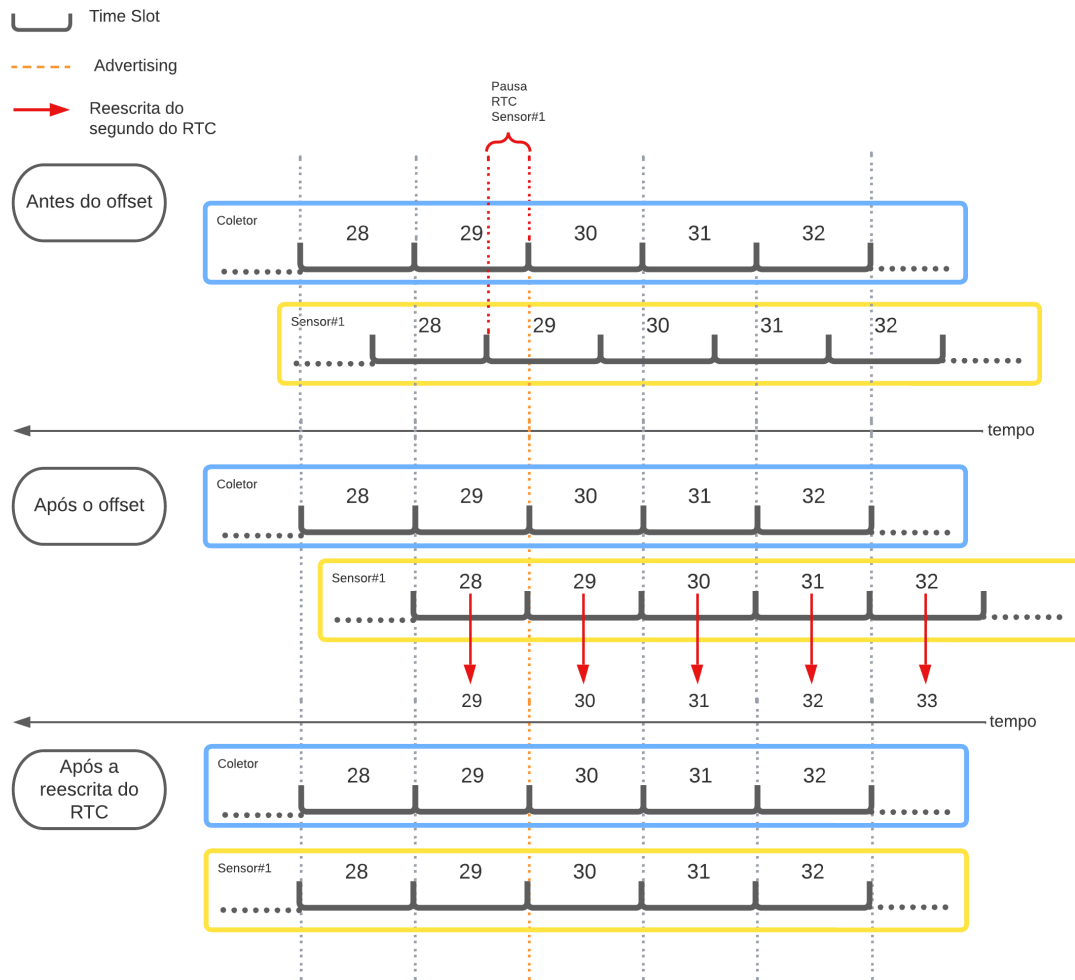
$$slotOffset = \Delta Parse + (slotReceivedADVOnsensor - slotADV) * 10^3 \quad [s] \quad (9)$$

Já se o nodo sensor se encontra com o seu relógio atrasado em relação ao relógio do nodo coletor, prossegue-se com o seguinte:

1. Mede-se a diferença do tempo entre o início do segundo do RTC do nodo sensor e o parse bem sucedido de um *advertisement* do nodo coletor (variável: $\Delta Parse$);
2. Realiza-se uma pausa no RTC do nodo sensor através do *halt-bit*;
3. Executa-se uma pausa no sistema de um tempo equivalente à $\Delta Parse$;
4. Libera-se o RTC do nodo sensor da pausa através do *halt-bit*;
5. Sobrescreve-se o valor dos segundos do RTC do nodo sensor com o resultado da Equação 10 onde $slotReceivedADVOnsensor$ é o *time slot* em que o nodo sensor recebeu o *advertisement* e $slotADV$ o *time slot* em que o coletor realiza o *advertisement* (uma constante registrada igual em todos os nodos).

A lógica descrita acima pode ser observada na Figura 21.

Figura 21: Diagrama de *offset* de correção para um nodo sensor com seu relógio atrasado em relação ao do nodo coletor.



Fonte: Autor.

$$slotOffset = slotADV + (slotADV - slotReceivedADVOnsensor) \quad [s] \quad (10)$$

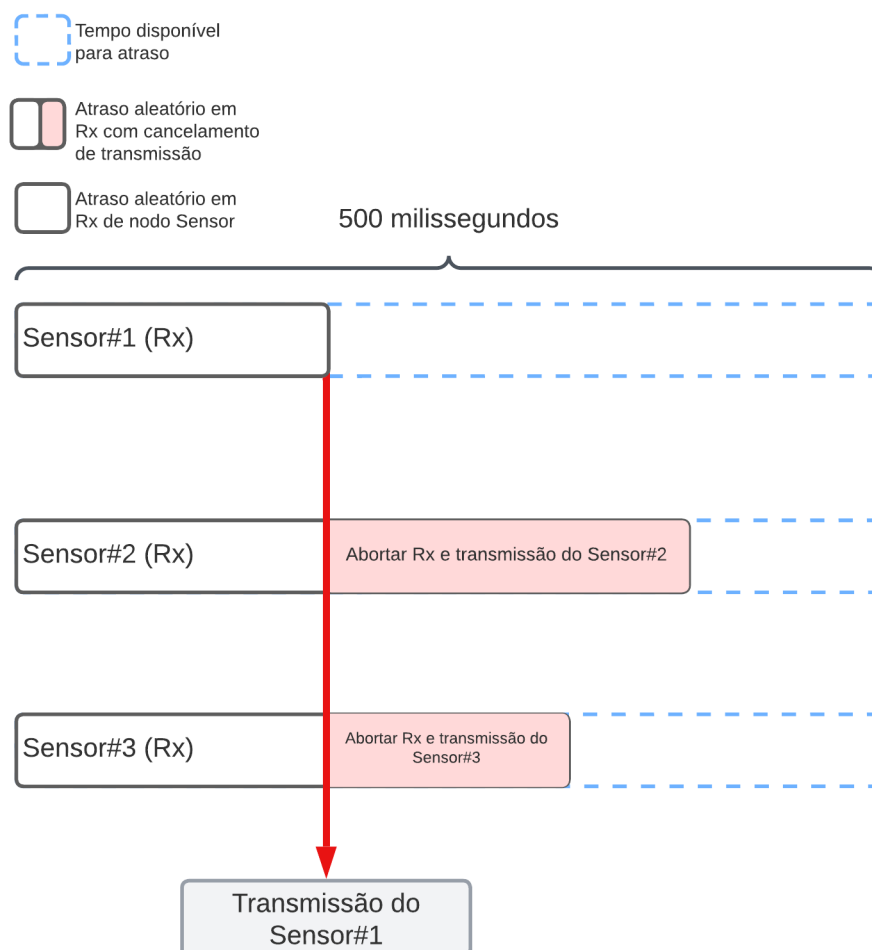
4.2.2.4 Métodos para evitar transmissão concorrente

Conforme o que foi apresentado até o momento, tem-se o suficiente para gerar um rede de sensores estáveis no momento em que se conecta um nodo sensor por vez, isto é, a cada anuncio de *time slot* só há um nodo sensor ativo para pleitear o *time slot*, ou seja, não há concorrência entre nodos sensores. Entretanto, há um problema no momento em que múltiplos nodos sensores tentam se conectar à rede de sensores, já que ocorrerão múltiplas transmissões simultâneas na tentativa de alocação de *time slot*, o que causa interferência nos sinais enviados, visto que os parâmetros LoRa utilizados por todos os nodos tem a necessidade de ser iguais.

Logo para certificar-se de que somente um nodo se ligaria ao *time slot* anunciado, foi implementada uma lógica para abortar o processo de transmissão de um nodo sensor a este *time slot* em questão.

Nodos sensores que recebem um *time slot* em um *advertisement*, tentam transmitir ao nodo coletor quando os segundos do seu RTC equivalem ao *time slot* recebido. Neste momento, a transmissão dos nodos sensores passara por um atraso no nodo Sensor que será de um tempo aleatório contido entre 0 e 500 milissegundos. Paralelamente a isso, enquanto este intervalo estiver em operação, os nodos sensores estarão constantemente tentando realizar o *parse* de mensagens enviadas por outros nodos sensores. Assim, o nodo que passou pelo atraso com sucesso, tem direito a transmitir no *time slot* em questão. Os demais nodos sensores que identificaram uma transmissão de um outro nodo sensor abortarão sua tentativa de alocar aquele *time slot*, retornarão a sua forma de receptor o qual tenta realizar o *parse* de uma mensagem de anúncio e só poderão tentar alocar o *time slot* na próxima mensagem de *advertisement*. Este processo pode ser observado na Figura 22.

Figura 22: Diagrama da lógica de cancelamento de transmissão para tentativas de conexão concorrentes.



Fonte: Autor.

Após um nodo sensor conseguir realizar esta primeira transmissão com sucesso no *time slot* proposto pelo nodo coletor, o segundo ponto de verificação para confirmar a entrada do nodo sensor com sucesso à rede está relacionada com o seu ID único.

Conforme a Seção 4.2.2.2, uma das informações que são enviadas para o nodo coletor no pacote transmitido pelo nodo sensor é o próprio ID único do nodo sensor em questão.

O nodo coletor ao receber uma transmissão de um nodo sensor no *time slot* enviado e sincronizado com o seu RTC, ele atualiza o valor do ID do último nodo sensor que juntou-se à rede para o ID que acabara de receber na transmissão.

Como apresentado na Seção 4.2.1.2, uma das informações enviadas no pacote de *advertisement* é justamente este último ID recebido. Logo, após uma primeira transmissão efetuada com sucesso por um nodo sensor, o que dirá se de fato ele conseguiu alocar o *time slot* em questão e se juntar a rede, é se o último ID conectado recebido na próxima mensagem *advertisement* é igual ao seu ID único.

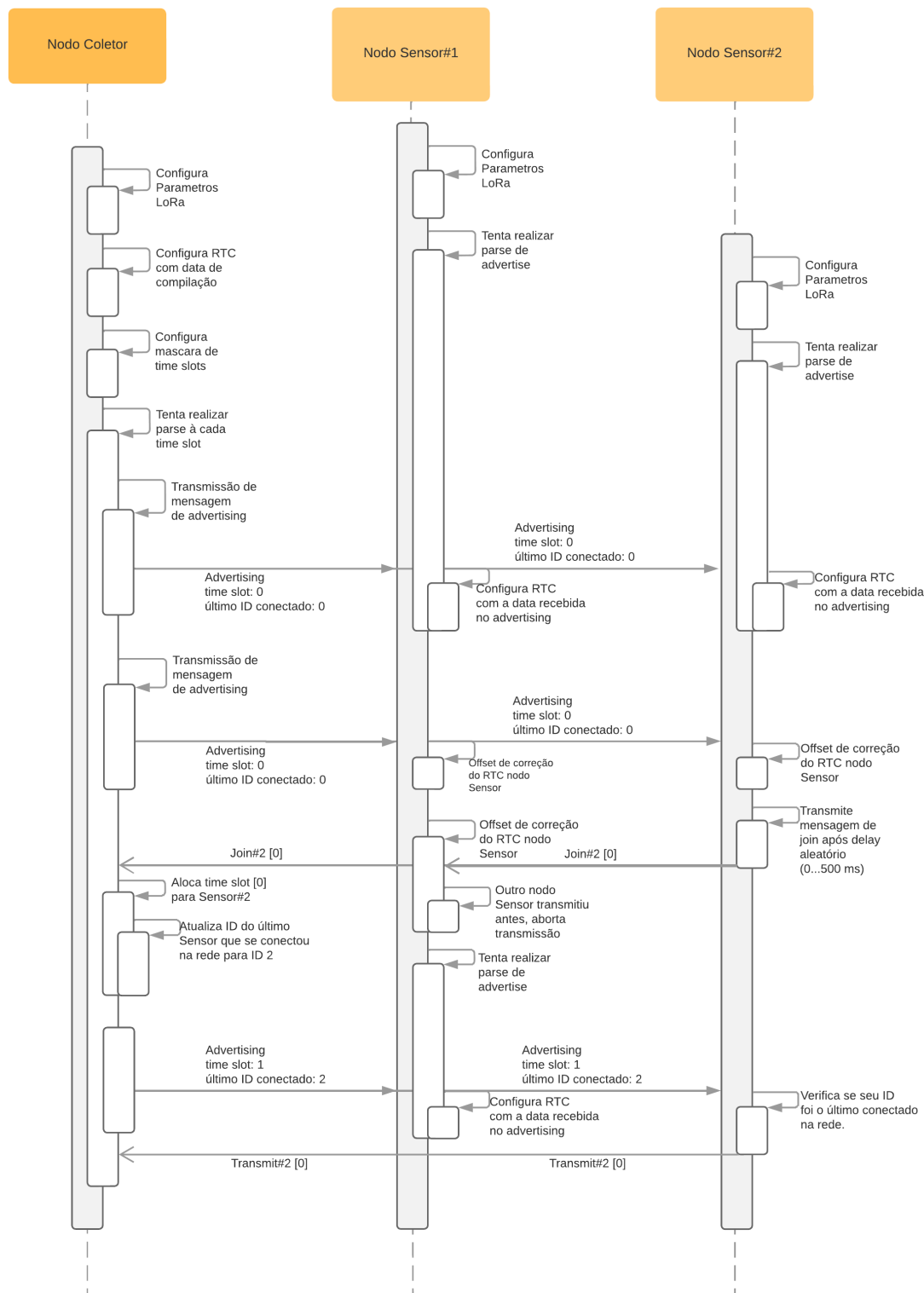
4.2.2.5 *Timeout: Nodo coletor*

Analogamente ao descrito para o nodo coletor na Seção 4.2.1.3, com a rede em operação, é importante que haja um mecanismo que possa gerenciar a possível perda do nodo coletor, de forma que os nodos possam voltar ao seu estado de receptor para tentar novamente receber um *advertisement* e tentar se juntar novamente à rede. Desta forma, um mecanismo de contagem de *timeout* foi aplicado as mensagens de *advertisement* enviadas pelo nodo coletor. No momento que um nodo sensor qualquer deixa de receber mensagens de *advertisement* por um número limite de vezes, o nodo sensor em questão volta ao seu estado de receptor na espera de um novo *advertisement*.

4.2.3 **Operação plena da rede**

Conforme as definições que foram feitas anteriormente nesta Seção 4.2, a Figura 23 expressa o diagrama de sequência que concatena uma operação completa deste sistema previamente definido.

Figura 23: Diagrama de sequência do processo de conexão de um nodo sensor à RSSF LoRa.



Fonte: Autor.

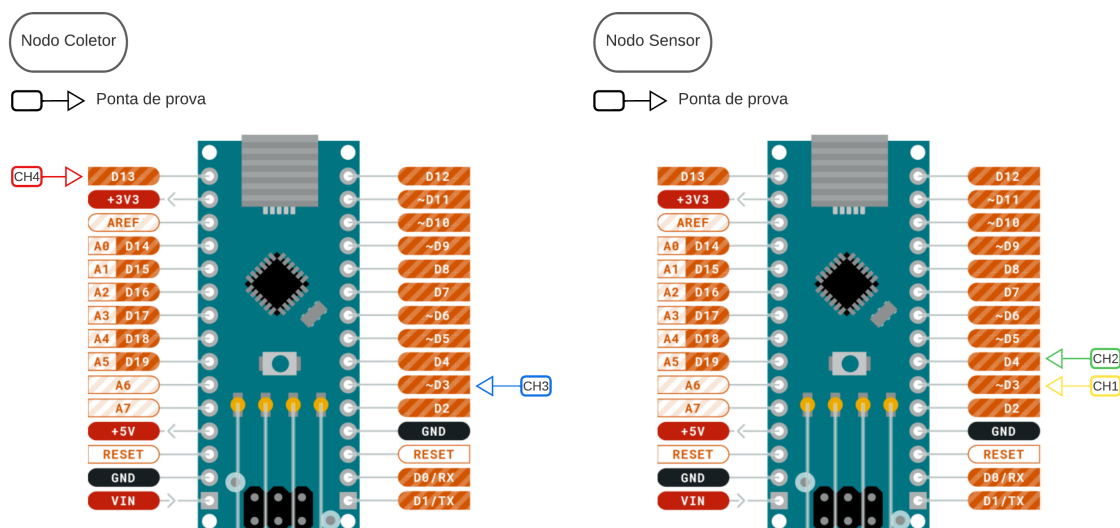
5 EXPERIMENTOS E RESULTADOS

5.1 Validação do sincronismo entre nodos

Para o projeto do nodo genérico LoRa a primeira funcionalidade que havia necessidade de ser analisada era o sincronismo da comunicação entre um nodo sensor e nodo coletor. A ocorrência dessa comunicação síncrona implica em os *time slots* do relógio do nodo coletor e de transmissão do nodo sensor estarem suficientemente alinhados.

Assim, para por esta análise em prática e validar o mecanismo de *offset* apresentado na Seção 4.2.2.3, foi utilizado um osciloscópio de 4 canais em conjunto com dois nodos, um em coletor outro em sensor. As conexões das pontas de provas do osciloscópio foram feitas conforme ilustra Figura 24.

Figura 24: Diagrama de conexão de pontas de prova em nodo coletor e nodo sensor para ensaio de sincronismo.



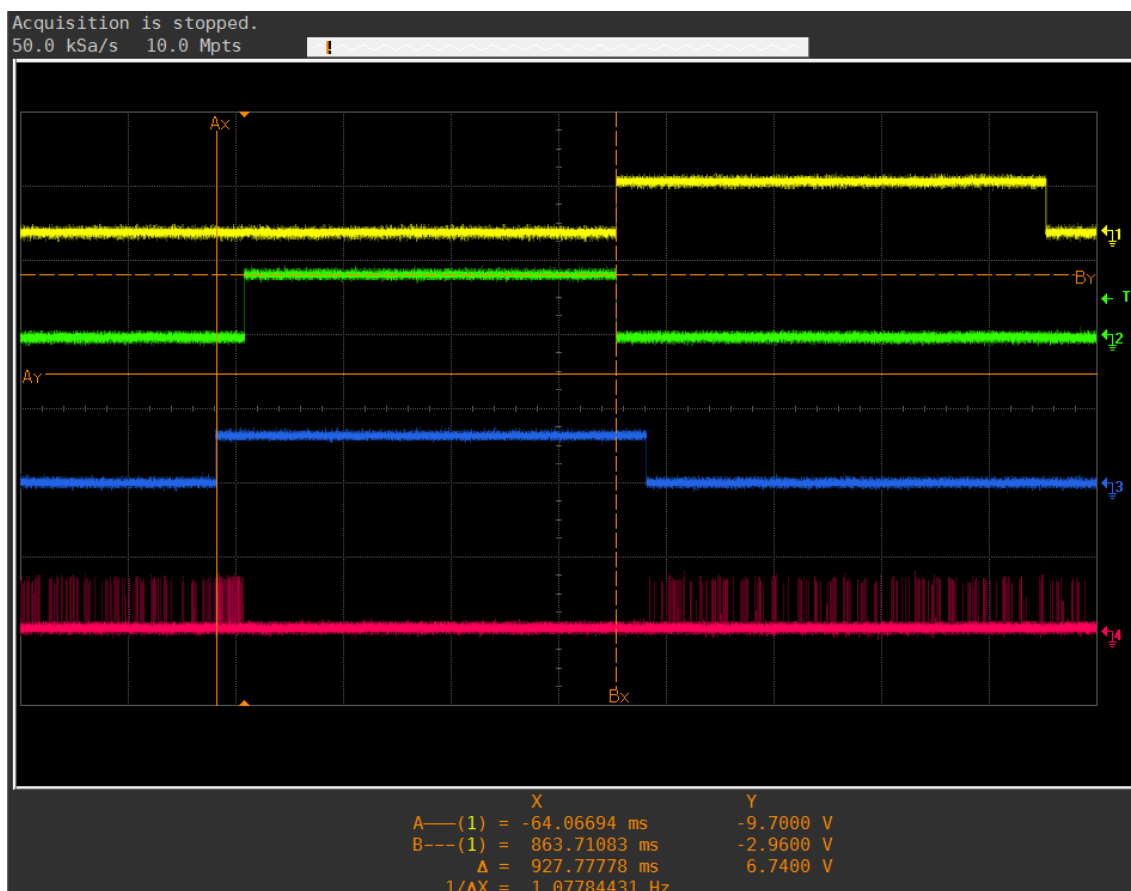
Fonte: Adaptado de (ARDUINO, 2022).

Em relação ao nodo coletor, representado na Figura 24, o Canal 4 foi conectado no pino D13 correspondente ao SCLK da comunicação SPI do módulo LoRa RFM95W apresentado na Seção 4.1.2 e o Canal 3 foi conectado no pino D3 do Arduino Nano, o qual entra em nível lógico alto enquanto o RTC do nodo coletor estiver no *time slot* do *advertisement*. Ainda em relação a Figura 24, para o nodo sensor, o Canal 1 foi conectado ao pino D3 do Arduino Nano, o qual entra em nível lógico alto enquanto o RTC do nodo

sensor estiver no *time slot* em que deveria receber o *advertisement* (definido o mesmo tanto para nodo coletor quanto para nodo sensor). O Canal 2 é conectado ao pino D4 o qual entra em nível lógico alto uma vez que é realizado *parse* sobre a mensagem de *advertisement* do nodo coletor e permanece neste nível alto até que o segundo em que foi recebido acabe. Este *parse* só ocorre a partir do momento em que o nodo sensor já recebeu uma mensagem de *advertisement* ao menos uma vez e configurou seu RTC, pois este segundo *parse* em diante atua na lógica de correção de *offset*, a qual se tem interesse em verificar nesta Seção.

Com as conexões feitas, deu-se início ao primeiro ensaio. O experimento se baseou em realizar o processo de entrada na rede do nodo coletor pelo nodo sensor e verificar no osciloscópio nos sinais monitorados se as correções de *offset* do RTC do nodo sensor estariam sendo feitas de forma correta. Desta forma, para realizar esta observação, verifica-se se o *time slot* de *advertisement* do nodo coletor coincide com o *time slot* em que o nodo sensor deveria receber o *advertisement* após as correções de *offset*. Foi configurado o osciloscópio para apresentar os sinais somente no momento em que um *parse* de *advertisement* é identificado e, por conseguinte, o Canal 2 do nodo sensor entra em nível lógico alto.

Figura 25: Sinais relativos ao primeiro ADV do primeiro ensaio de verificação da correção de *offset*.



Fonte: Autor.

Como mencionado anteriormente, o Canal 1 verifica o pino D3 do nodo sensor que entra em nível lógico alto enquanto o seu RTC estiver no *time slot* em que deveria receber

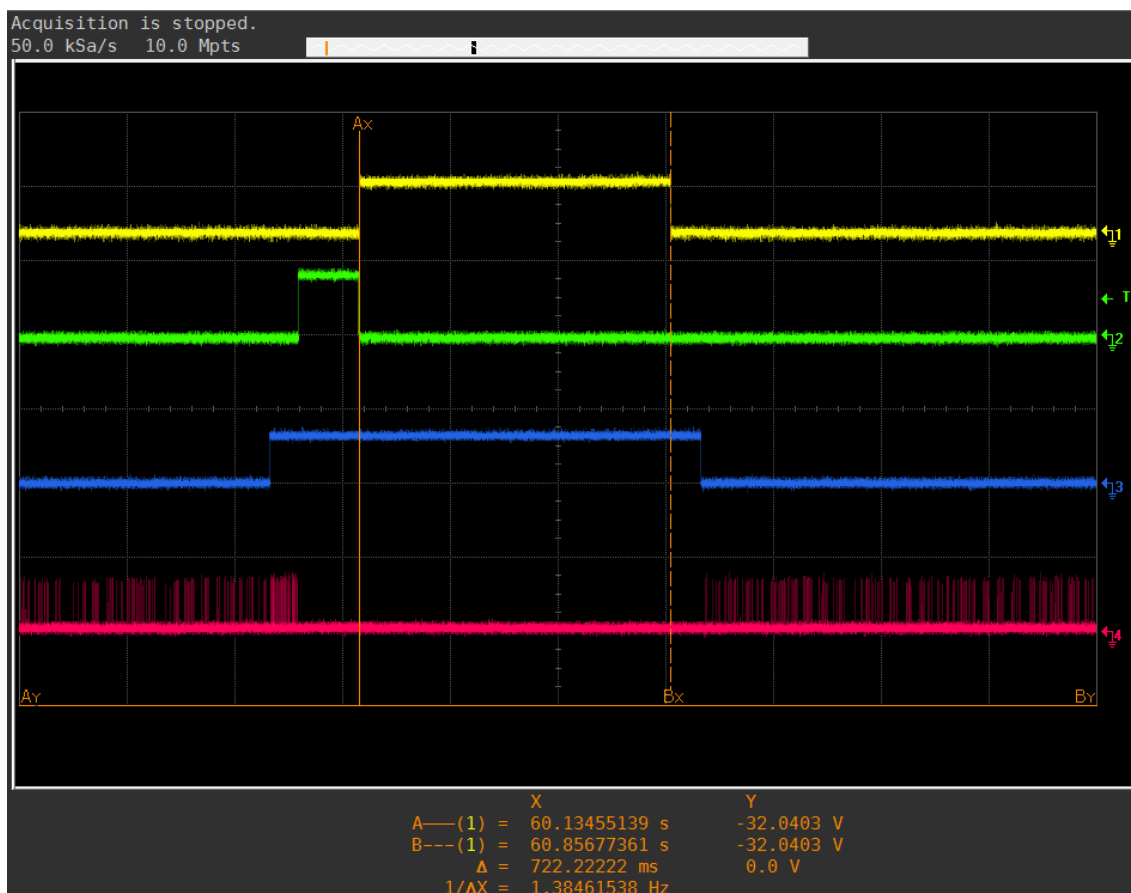
o *advertisement*. Logo, como observado na Figura 25, pelo Canal 2 entrar em nível lógico alto logo antes do Canal 1, significa que o nodo sensor realizou *parse* do *advertisement* um *time slot* anterior em relação ao que deveria. Ou seja, mesmo tendo sido inicialmente configurado com os parâmetros do RTC do nodo coletor previamente, o nodo sensor se encontra com o seu relógio adiantado em relação ao do nodo coletor.

Figura 26: Medição RTC do nodo coletor (Canal 3) relacionado ao segundo ADV do primeiro ensaio de verificação da correção de *offset*.



Fonte: Autor.

Figura 27: Medição *time slot* de ADV no nodo sensor (Canal 1) relacionado ao segundo ADV do primeiro ensaio de verificação da correção de *offset*.

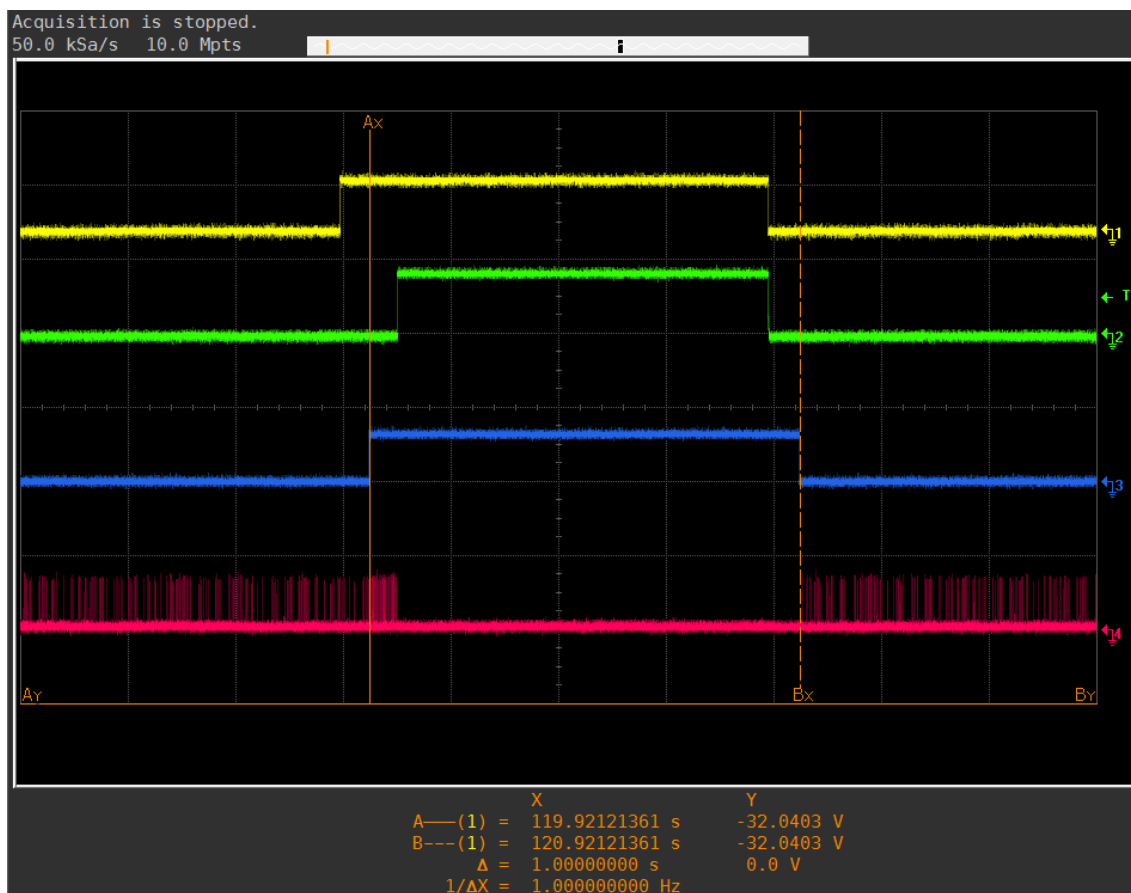


Fonte: Autor.

Como pode ser observado na Figura 26, relativa à segunda mensagem de *advertisement* enviada pelo coletor, o tempo do *time slot* de *advertisement* do nodo coletor medido pelo Canal 3, é de exatamente 1 segundo. Já, com relação a Figura 27 o tempo do *time slot* de ADV do nodo sensor se limita a 722 milissegundos.

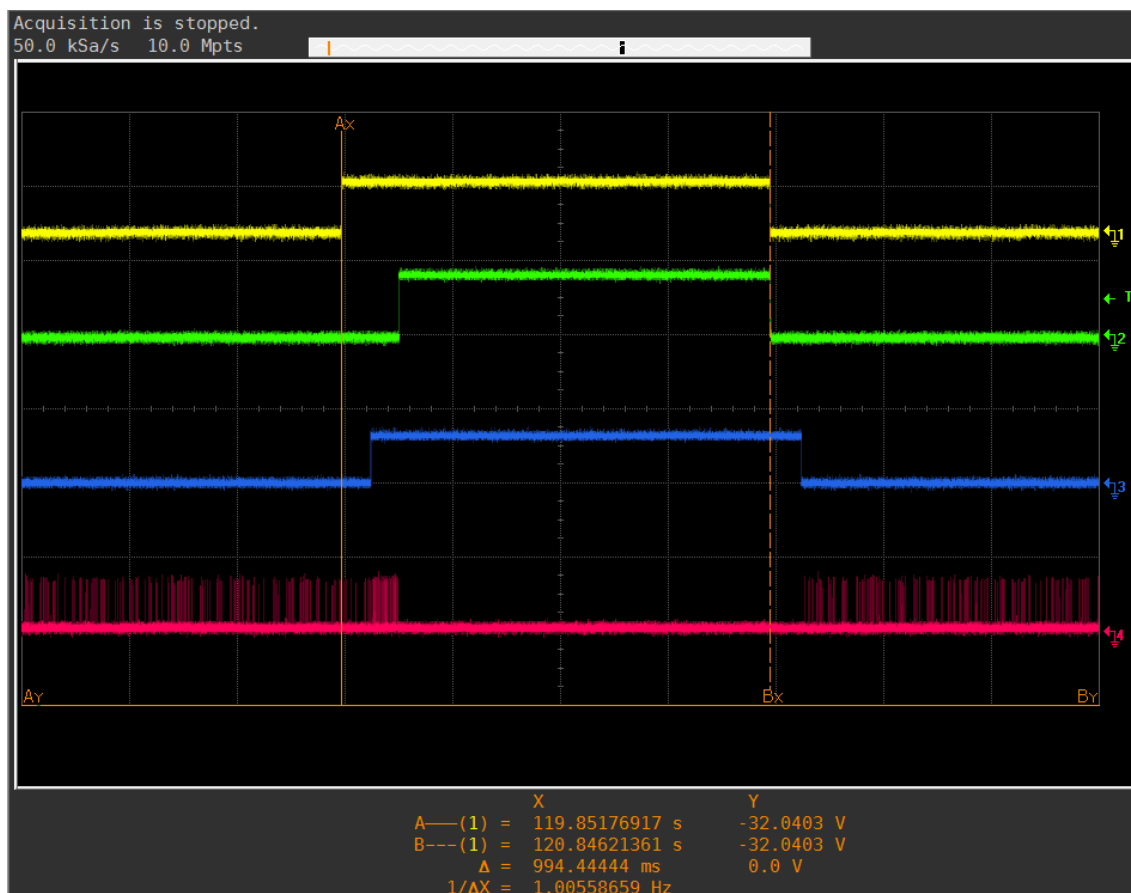
O que se esperaria em um primeiro momento seria uma alongamento do sinal do Canal 1 causado pelo atraso, de forma a borda de subida do sinal do Canal 1 (*ADV* sensor) aproximadamente coincidir com a borda de descida do sinal do Canal 3 (*ADV* coletor), seguido por uma reescrita nos segundos do RTC do nodo sensor para ajustar os segundos e os milissegundos não contabilizados pelo RTC do nodo sensor. Entretanto, apesar de ser possível observar que a reescrita dos segundos do RTC foi realizada com sucesso, pois o nível alto do Canal 4 está se sobrepondo sobre o sinal do alto do Canal 3, as bordas não estão coincidindo, ou seja, a operação de *offset* não foi realizada com sucesso.

Figura 28: Medição RTC do nodo coletor (Canal 3) relacionado ao terceiro ADV do primeiro ensaio de verificação da correção de *offset*.



Fonte: Autor.

Figura 29: Medição *time slot* de ADV no nodo sensor (Canal 1) relacionado ao terceiro ADV do primeiro ensaio de verificação da correção de *offset*.



Fonte: Autor.

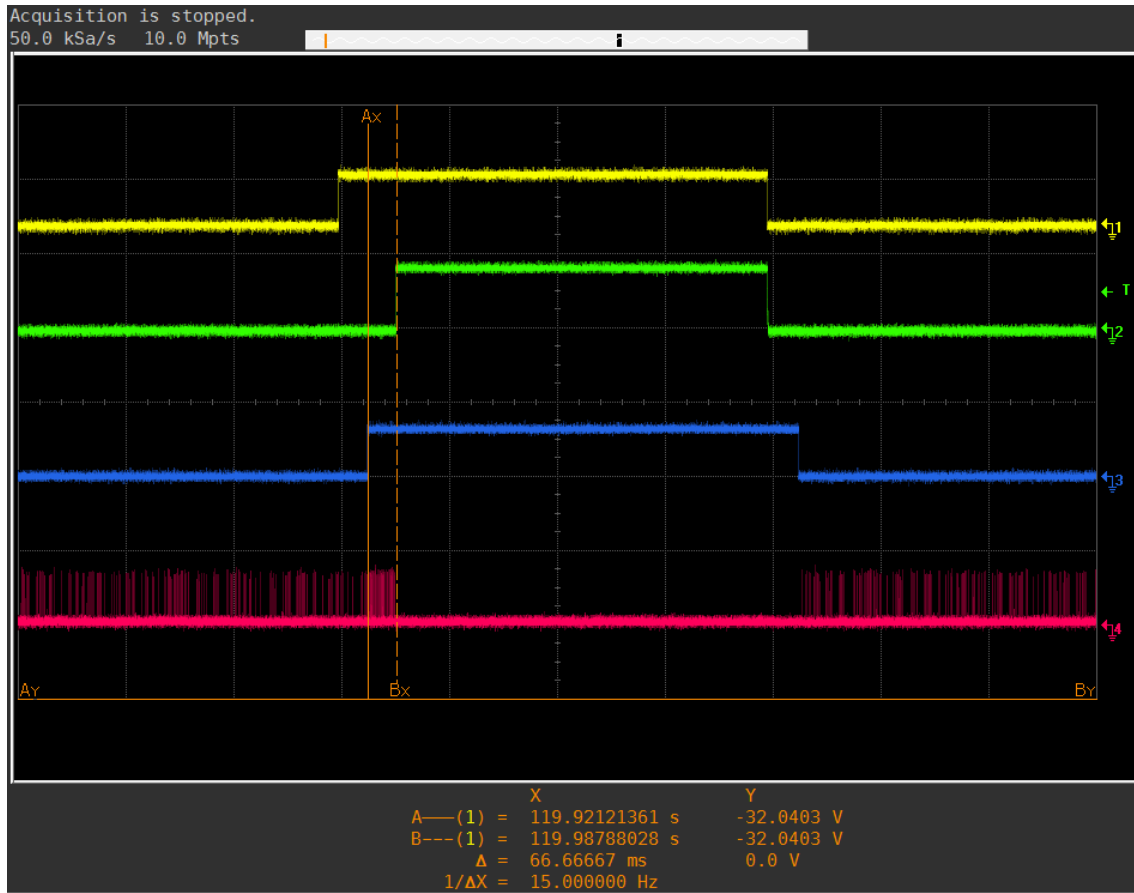
Como pode ser observado na correção realizada no terceiro *advertisement* representados nas Figura 28 e 29, o *offset* continua errôneo sendo que as bordas de subida e descida do sinal do Canal 1 (ADV sensor) não estão coincidindo suficientemente com as bordas do sinal do Canal 2 (ADV coletor). Além do mais, é possível observar na Figura 29 que o sinal do Canal 1 não está completando 1 segundo como devia, o que implica que a forma como o *offset* está sendo calculado precisa ser ajustada.

Figura 30: Medição SCLK da comunicação SPI do módulo LoRa (Canal 4) relacionado ao segundo ADV do primeiro ensaio de verificação da correção de *offset*.



Fonte: Autor.

Figura 31: Medição SCLK da comunicação SPI do módulo LoRa (Canal 4) relacionado ao terceiro ADV do primeiro ensaio de verificação da correção de *offset*.



Fonte: Autor.

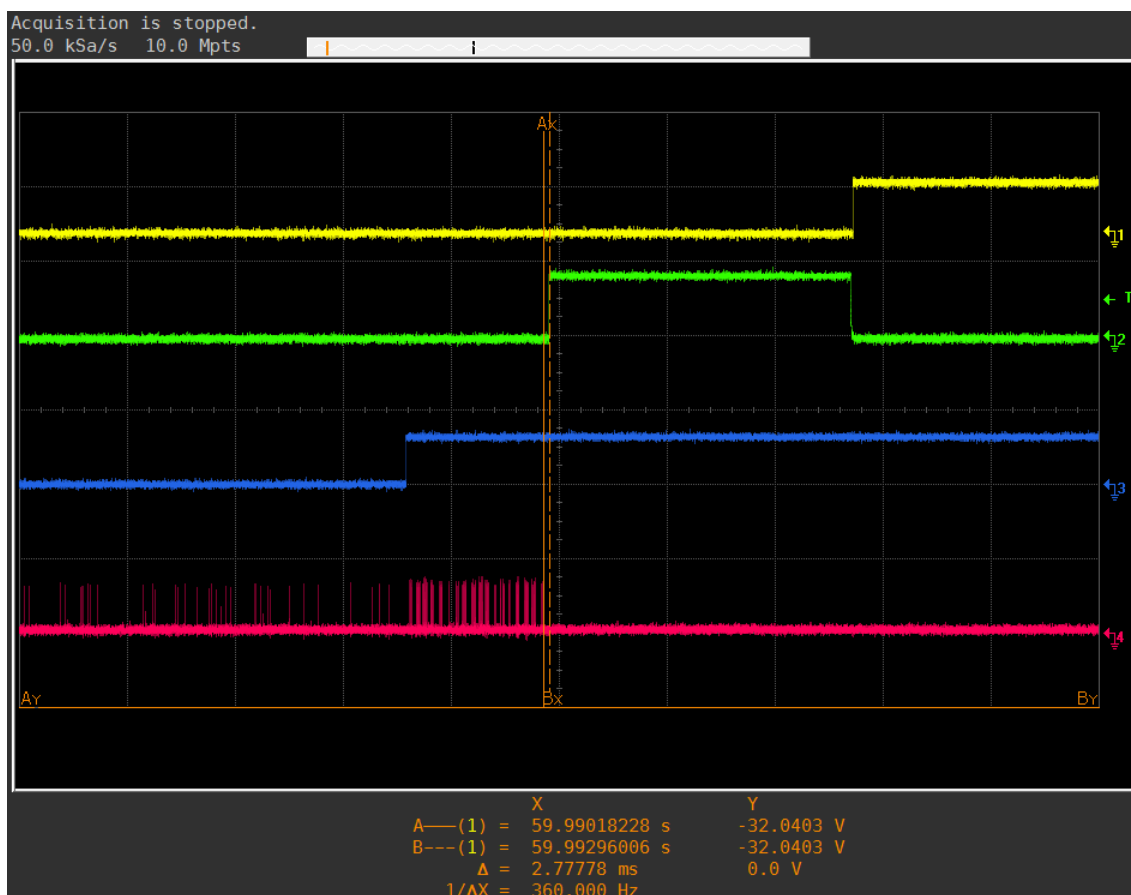
Analisando mais profundamente as amostras obtidas, sendo algumas representadas pelas Figuras 30 e 31, em relação ao sinal do Canal 4, é possível observar que o módulo LoRa RFM95W não realiza sua transmissão instantaneamente quando o nodo coletor entra no *time slot* de *advertisement*, logo, este deve ser considerado no cálculo do *offset*. Comparando as Figuras 30 e 31 nota-se que este tempo de transmissão (ADV_{time}) não é constante. Para mitigar o impacto de picos de valores desses tempos de transmissão, aplica-se um filtro média móvel, apresentado na Seção 3.4, com $M = 5$ sobre estes valores ($ADV_{time_{avgFilt}}$) e se passa a considerar no cálculo do *offset* define-se a Equação 11 a qual passa a ser considerada na Equação 8, desta forma, gerando a Equação 12.

$$ADV_{time_{avgFilt}} = \begin{cases} 0 & \text{if } ADV_{count} = 0 \\ ADV_{elapsed}[t_{time\ slot} - 1] & \text{if } 0 < ADV_{count} < 5 \\ \frac{1}{5} \sum_{j=0}^4 ADV_{elapsed}[t_{time\ slot} - j] & \text{if } ADV_{count} \geq 5 \end{cases} \quad (11)$$

$$\Delta Parse = (ADV_{parse_{time}} - ADV_{slot_{begin}}) - ADV_{time_{avgFilt}} \quad [ms] \quad (12)$$

Onde ADV_{count} é o número de *advertisements* transmitidos pelo nodo coletor, $ADV_{elapsed}$ é o tempo decorrido na transmissão do *advertisement* referente ao seu *time slot* e $t_{time\ slot}$ o *time slot* do *advertisement* mais recente.

Figura 32: Medição do tempo decorrido entre o envio do *advertisement* pelo nodo coletor (Canal 4) e seu *parse* no nodo sensor (Canal 2).

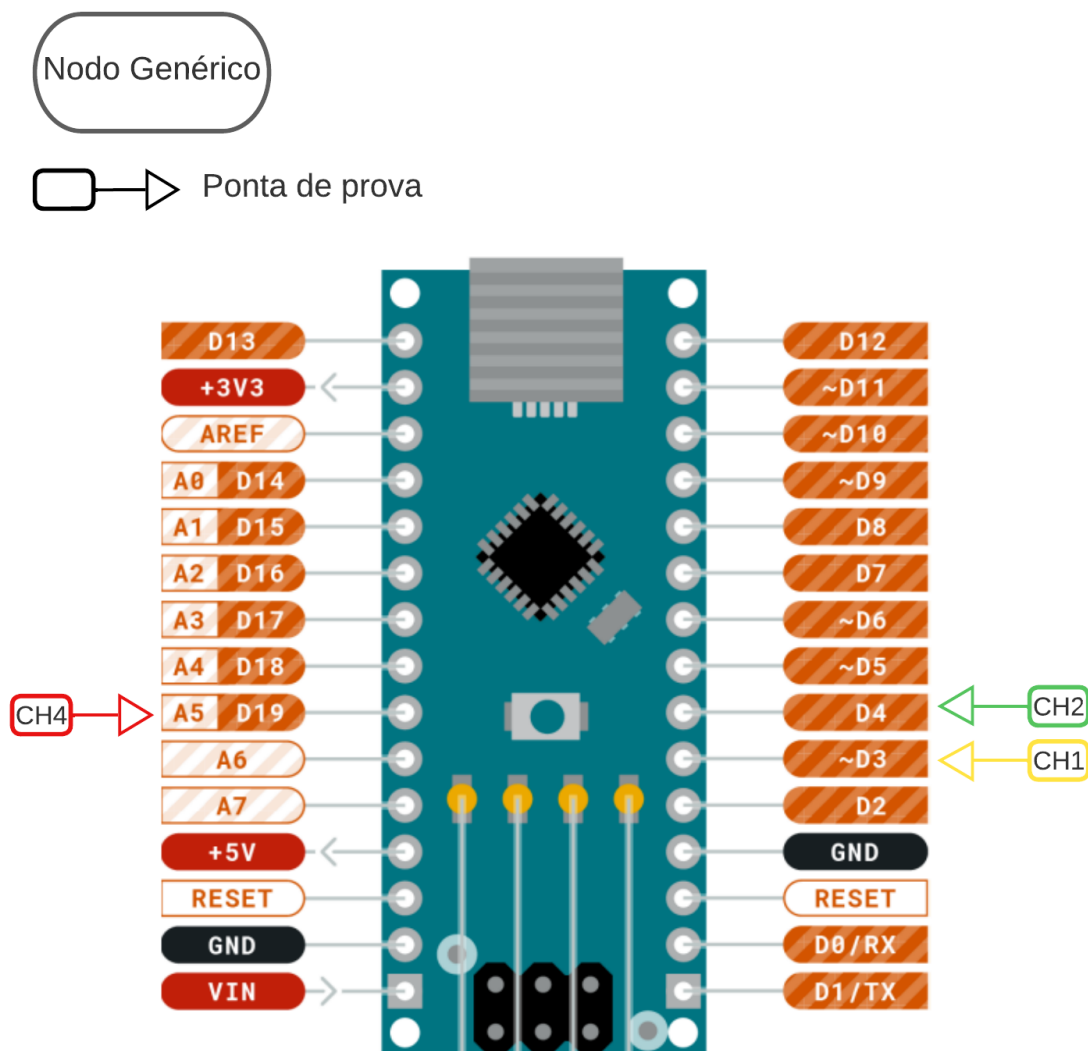


Fonte: Autor.

Outro fator que pode impactar no *offset* seria o tempo entre o envio do *advertisement* pelo nodo coletor e seu *parse* pelo nodo sensor. Todavia, como pode ser visto na Figura 32, este tempo se limita a poucos milissegundos, o qual, quando comparado com a escala do *time slot* em segundos, este tempo é ínfimo e não será considerado para o cálculo de *offset* de correção.

Para investigar os possíveis impactos que o mecanismo de *halt* do RTC utilizado no *offset* possa ter sobre seu cálculo de correção é preciso realizar um novo ensaio.

Figura 33: Diagrama de conexão de pontas de prova em nodo genérico para ensaio verificação de impacto do *halt* do RTC no sincronismo.

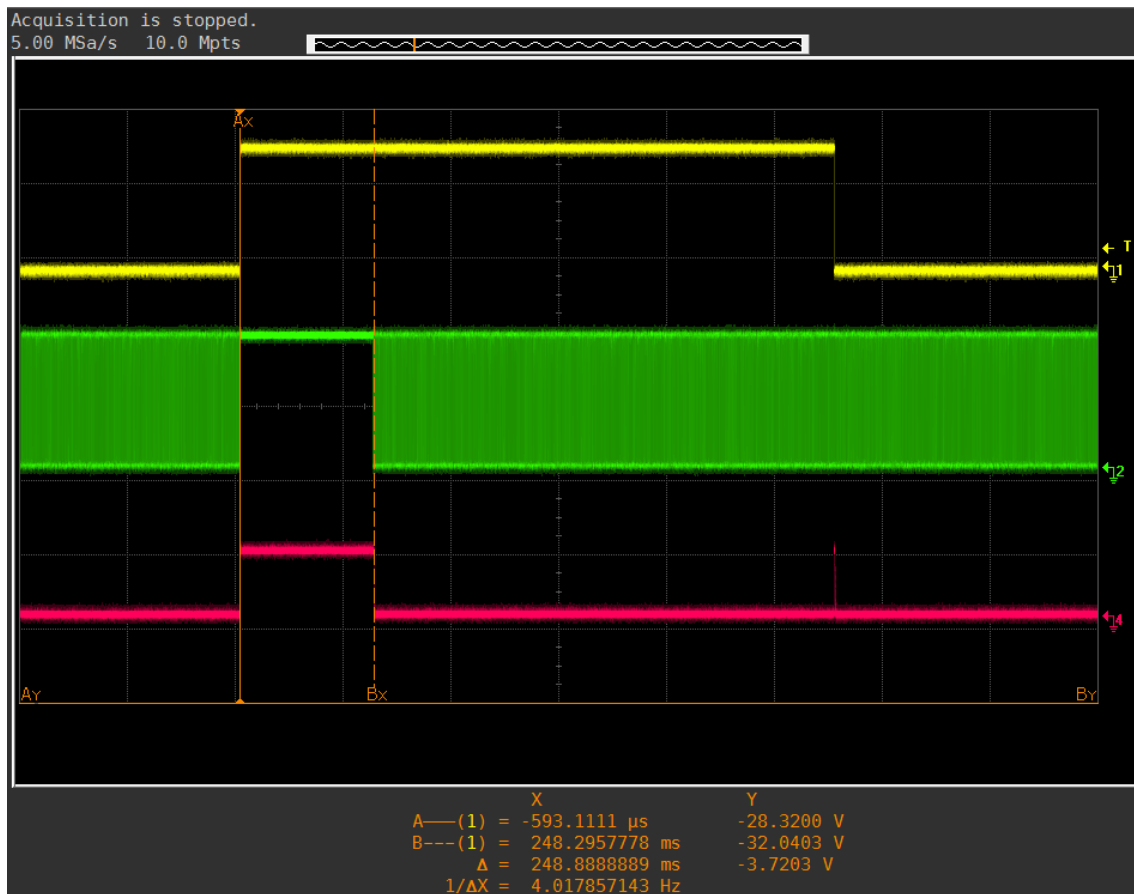


Fonte: Adaptado de (ARDUINO, 2022).

A Figura 33 representa as conexões das pontas de prova para o este segundo ensaio onde, o Canal 1 é conectado no pino D3 o qual tem saída do contador de segundos do RTC do nodo, o Canal 2 é conectado no pino D4, que por sua vez está conectado ao CLK do RTC e, por fim, o Canal 4 é conectado no pino D19 o qual fica em nível lógico alto em quanto o *halt* RTC estiver ativo.

Para executar esse teste foi gravado um *firmware* no Arduino Nano que conta os segundos pelo RTC e quando se igual ao *time slot* definido é ativado o *halt* do RTC, seguido por um atraso de 248 milissegundos e finalizado pela liberação do *halt* do RTC.

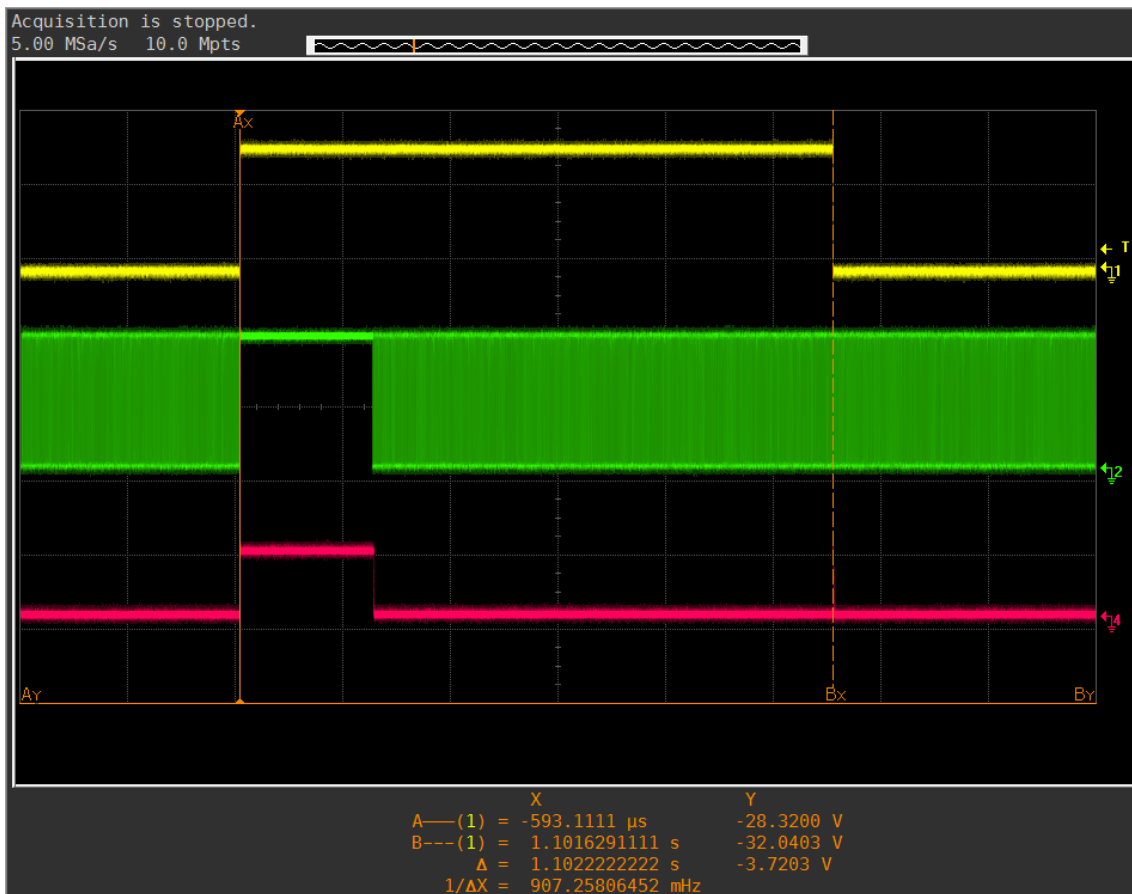
Figura 34: Medida do sinal do *halt* (Canal 4) para o segundo ensaio.



Fonte: Autor.

Como pode ser observado no sinal do Canal 4 na Figura 34, a duração do *halt* foi realizada nos 248 milissegundos esperados. Adicionalmente, pode-se notar que, como apresentado na Seção 4.1.3, a ativação do *halt* desativa o oscilador do RTC, por conseguinte, desativando o CLK como pode ser visto no Canal 2 na Figura 34.

Figura 35: Medida do sinal do contador do RTC (Canal 1) para o segundo ensaio.



Fonte: Autor.

Apesar do tempo do *halt* ter sido o esperado, pode-se observar na Figura 35 que ocorre uma inconsistência em relação ao tempo de permanência no *time slot* em que o evento do *halt* ocorre. O tempo de duração do *time slot* em questão apresentado na Figura 35 é de 1,102 segundos, enquanto o esperado com o *halt* mais o atraso de 248 milissegundos seria de 1,248 segundos.

Com isso, infere-se que o processo de *halt* do RTC DS1307 não retorna diretamente para a contagem dos segundos. Desta forma, um fator de correção relacionado ao processo de *halt* precisa ser considerado no cálculo do *offset* de correção. Para a Figura 35 o Fator de Correção (FC_{halt}) se define pela Equação 13.

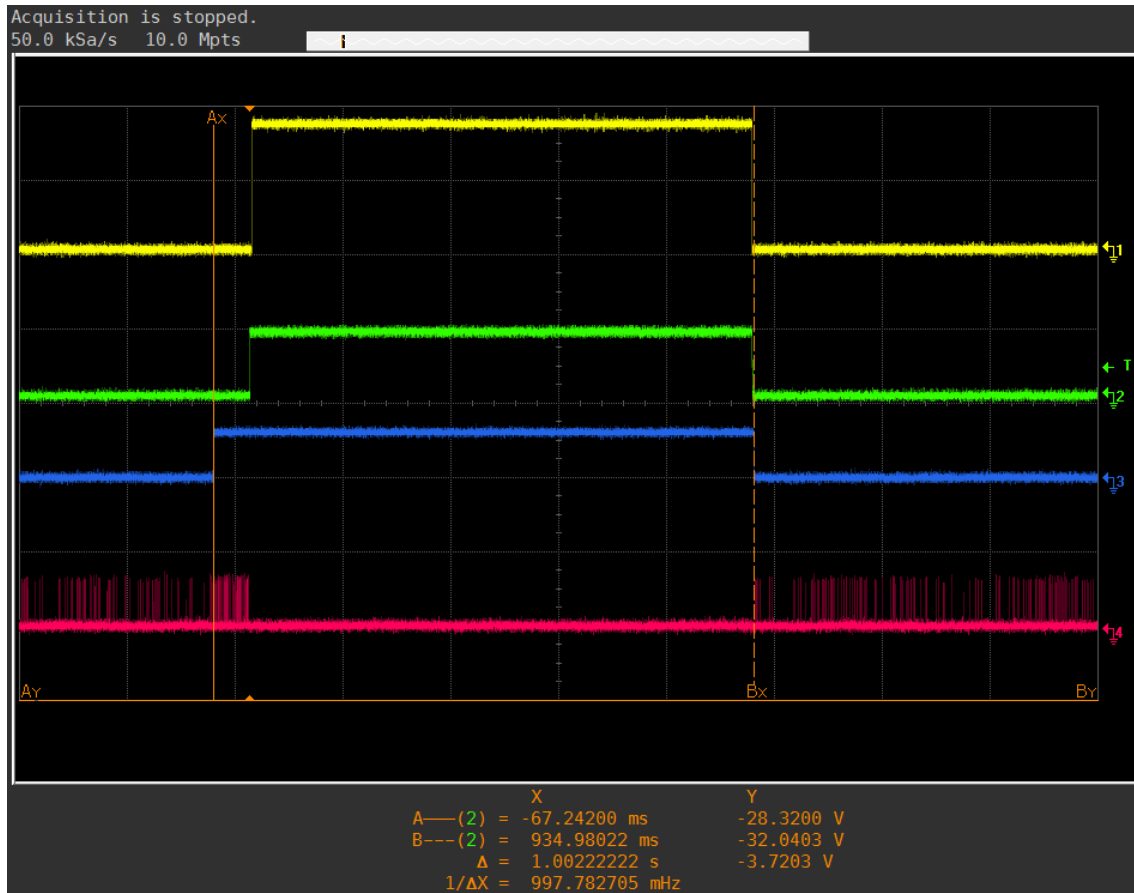
$$FC_{halt} = (\text{tempo esperado} - \text{tempo real})_{time\ slot} = 1,248 - 1,102 = 0,146 [ms] \quad (13)$$

Desta maneira, considerando o Fator de Correção FC_{halt} , definido na Equação 13, na Equação 12 obtém-se a Equação 14

$$\Delta Parse = (ADV_{parse\ time} - ADV_{slot\ begin}) - ADV_{time\ avg\ Filt} + FC_{halt} [ms] \quad (14)$$

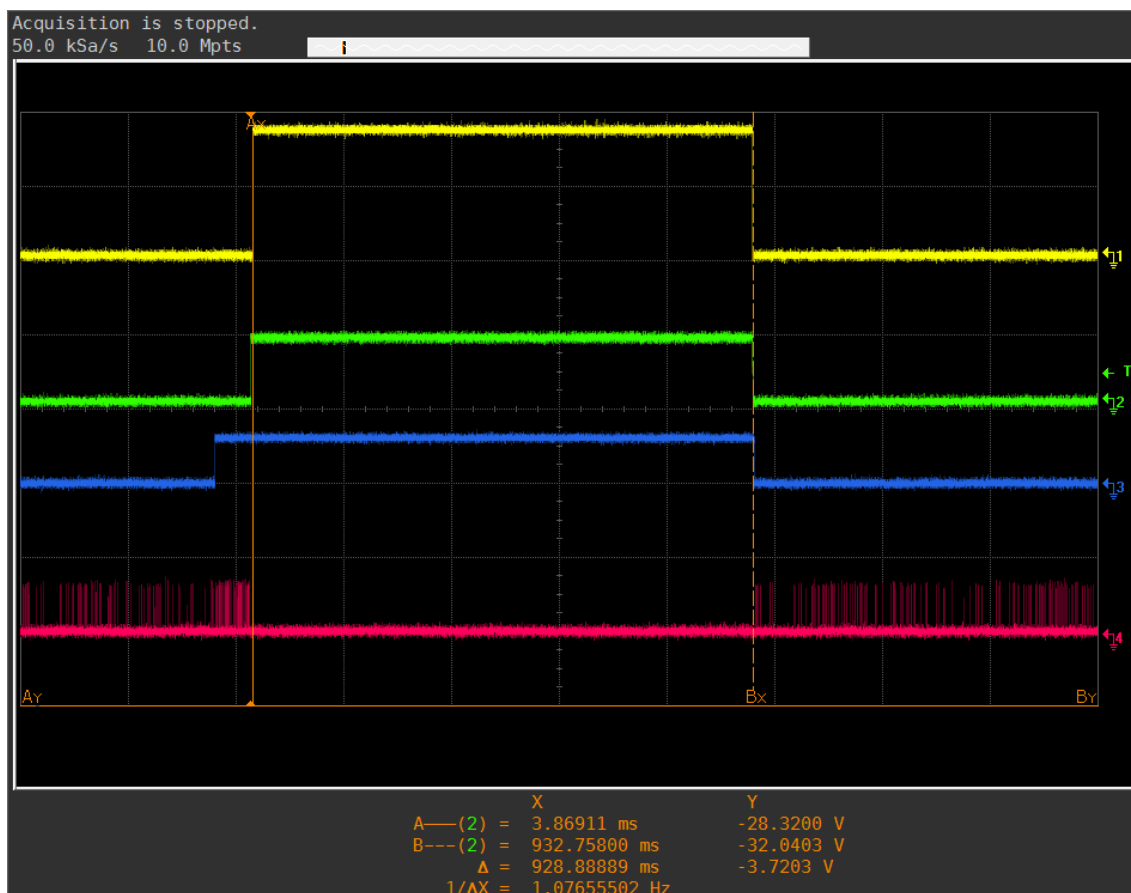
Em seguida é realizado o primeiro ensaio com a pinagem das pontas de prova do osciloscópio representadas pela Figura 24 e atualizando os cálculos de *offset* de correção com $\Delta Parse$ referente à Equação 14.

Figura 36: Medição RTC do nodo coletor (Canal 3) relacionado ao primeiro ADV do primeiro ensaio de verificação da correção de *offset* após ajuste de equações.



Fonte: Autor.

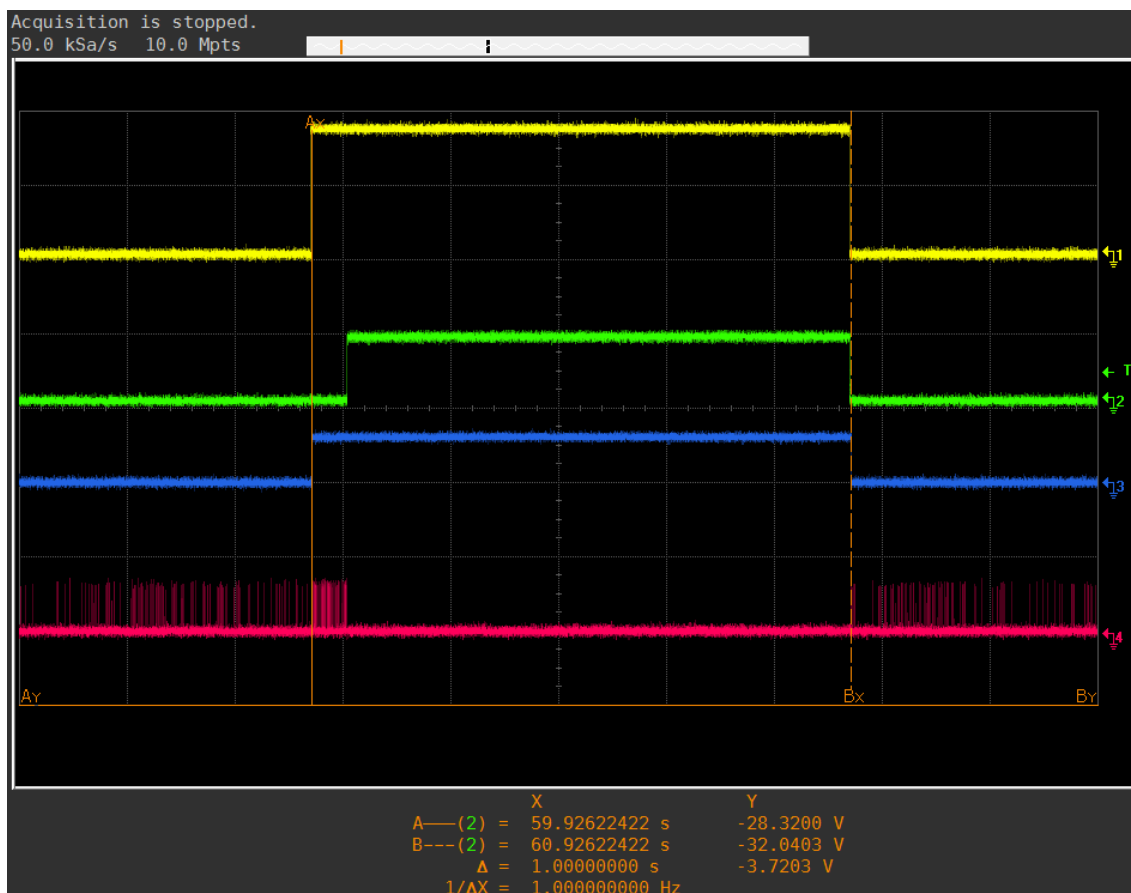
Figura 37: Medição *time slot* de ADV no nodo sensor (Canal 1) relacionado ao primeiro ADV do primeiro ensaio de verificação da correção de *offset* após ajuste de equações.



Fonte: Autor.

Nota-se as Figuras 36 e 37, que nesta versão do ensaio o *offset* de correção foi realizado tal como esperado. O nodo sensor estando adiantado foi corrigido através do atraso de *slot offset*, representado pela Equação 9 de *offset* de correção, desta forma, alinhando a borda de descida dos *time slots* de ADV.

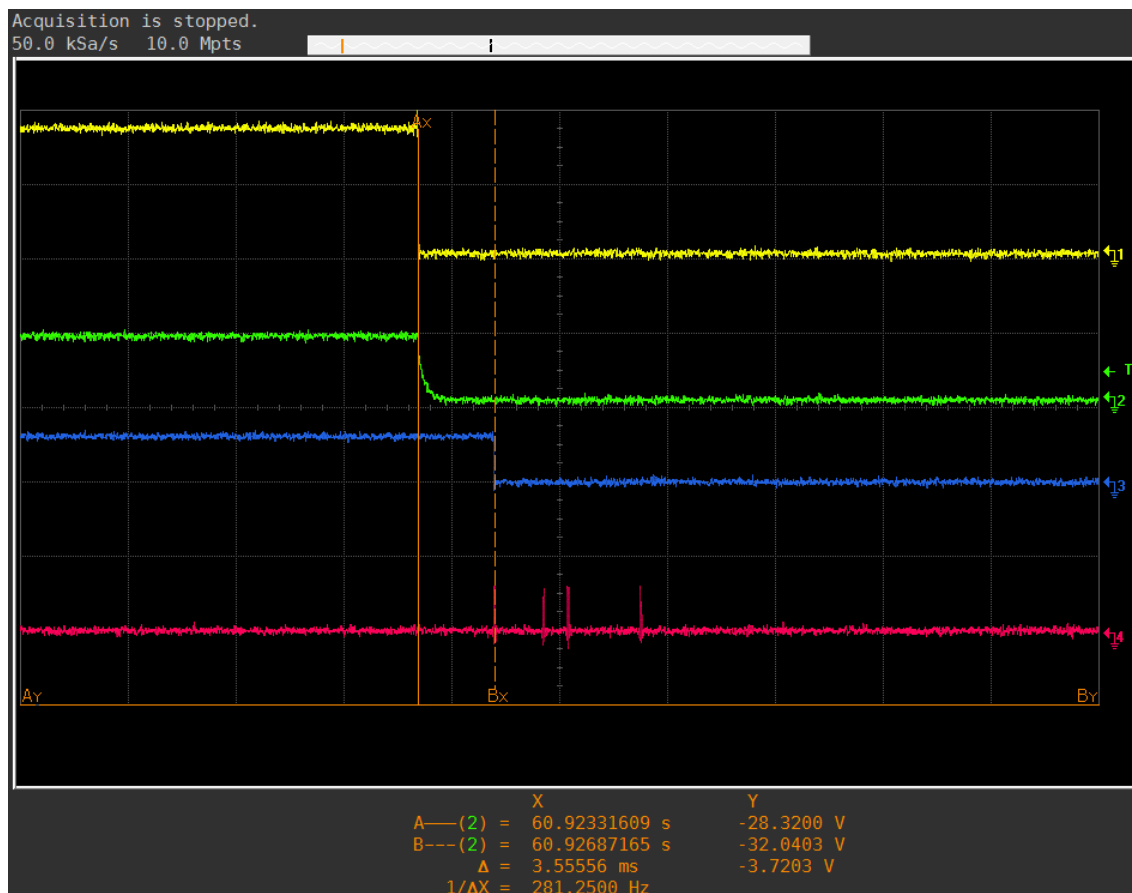
Figura 38: Medição *time slot* de ADV no nodo coletor (Canal 3) relacionado ao segundo ADV do primeiro ensaio de verificação da correção de *offset* após ajuste de equações.



Fonte: Autor.

Observa-se na Figura 38, a qual representa o *offset* de correção aplicado em uma segunda mensagem de *advertisement*, que os *time slots* de ADV de ambos os nodos estão alinhados. Este alinhamento refletirá para toda a cadeia de *time slots*.

Figura 39: Medição da diferença entre *time slot* do nodo coletor (Canal 3) e do nodo sensor (Canal 1) relacionado ao segundo ADV do primeiro ensaio de verificação da correção de *offset* após ajuste de equações.



Fonte: Autor.

Na Figura 39 é possível observar que o sincronismo não é perfeito pois, há uma diferença de alguns milésimos de segundos entre os *time slots* de ADV de ambos os nodos. Entretanto, esta defasagem é ínfima em relação a escala de segundos dos *time slots*. Portanto, para as correções realizadas na equações de *offset*, considera-se que seus resultados são suficientemente sincronizados para esta aplicação de rede de nodos sensores sem fio.

5.2 Validação das conexões na rede por inicialização individual de nodo sensor

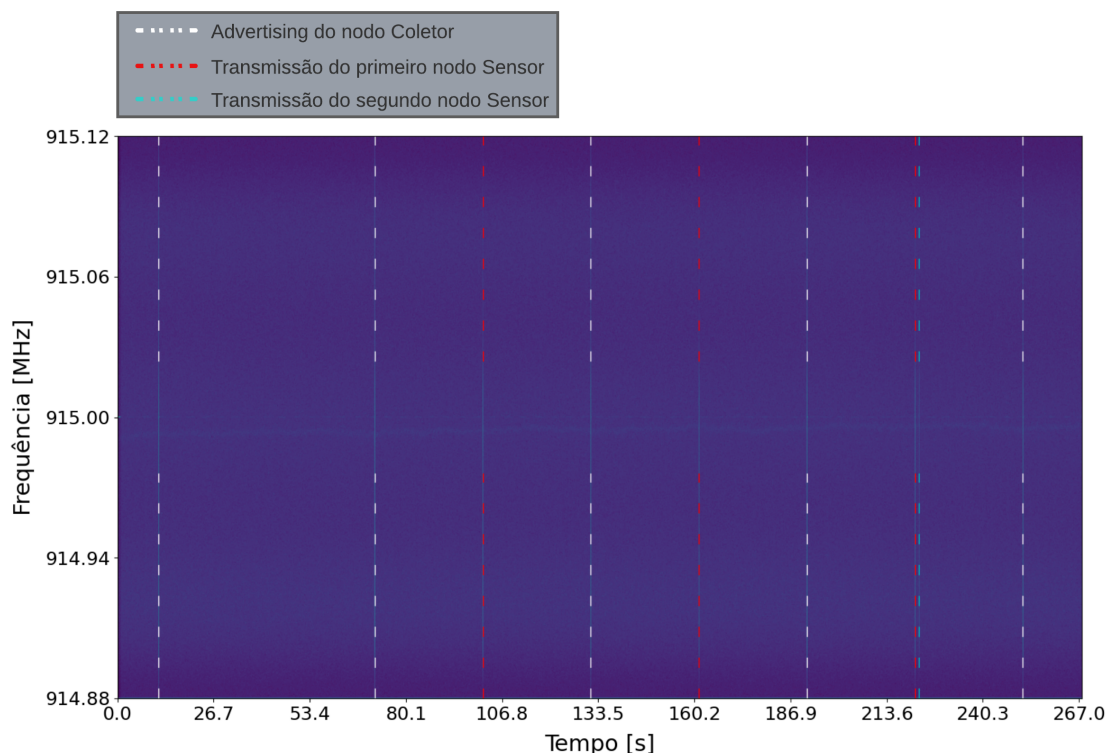
Para validar a funcionalidade de agregação dos nodos à rede e transmissão nos *time slot* alocados, utilizou-se 3 nodos, dois configurados como sensores e um como coletor. Foram coletadas amostras através de um rádio definido por software (RDS), e posteriormente plotadas em um espectrograma para examinar o momento das transmissões dos nodos. Em seguida, com o objetivo de analisar a formação da rede com a conexão dos nodos sensores separadamente, foi realizado o seguinte ensaio:

1. Inicialização do nodo coletor;

2. Inicialização de um temporizador a partir da primeira mensagem de *advertisement* do nodo coletor;
3. Aos 30 segundos do temporizador, inicialização do primeiro nodo sensor;
4. Aos 45 segundos do temporizador, inicialização do analisador de espectro através do *RDS*;
5. Aos 120 segundos do temporizador, inicialização do segundo nodo sensor.

O primeiro *advertisement* do nodo coletor servirá para determinar o temporizador que será levado em conta para o acionamento dos analisador de espectro e dos nodos sensores. É esperado que os nodos tomem 2 *advertisements* para entrar na rede e começar a transmitir, sendo o primeiro para configurar seu RTC e o segundo para realizar as operações de correção de sincronismo exploradas nas Seções 4.2.2.3 e 5.1. Ao final deste ensaio, ambos nodos sensores devem ter se conectado a rede e estar transmitindo no *time slot* anunciado pelo nodo coletor e selecionado pelo gerenciador de *time slots*, explorado na Seção 4.2.1.1.

Figura 40: Gráfico do espectro das transmissões dos nodos em conexão na rede LoRa.



Fonte: Autor.

Como pode ser evidenciado na Figura 40, o *advertisement* está sendo enviado periodicamente pelo nodo coletor a cada 60 segundos como previsto. A transmissão do primeiro nodo sensor só ocorre após a ocorrência de duas mensagens de ADV e passa a ser transmitida periodicamente a cada 60 segundos. Esta observação mostra que o primeiro nodo sensor foi capaz de se inicializar e realizar as correções de sincronização alocando o *time*

slot anunciado pelo *advertisement* do coletor para a sua transmissão, ou seja, o primeiro nodo sensor se conectou na rede com sucesso. Este mesmo comportamento pode ser percebido para o segundo nodo sensor, uma vez que passa a transmitir somente após a passagem de dois *advertisements*, sendo este no caso os que ocorrem após 130 segundos decorridos no analisador de espectro da Figura 40.

Observa-se também, que o gerenciador de *time slots* está realizando o manejo dos *time slot* com sucesso, uma vez que, como visto na Figura 40, ambos os nodos sensores se conectam a rede, passam a transmitir em *time slots* subsequentes. Ou seja, ao ser ocupado um *time slot*, o gerenciador de *time slot* passa a anunciar o próximo na cadeia, o que está de acordo com as definições da Seção 4.2.1.1.

5.3 Validação das conexões na rede por nodo sensor em transmissão concorrente

A validação de transmissões concorrentes de nodos sensores em disputa por um *time slot* anunciado em mensagem de *advertisement* foi realizada com 3 nodos, 1 em modo coletor e 2 em modo sensor. Adicionalmente, a obtenção dos dados para esta validação foi realizada com o auxílio de um analisador de pacotes LoRa. Logo, os seguintes procedimentos foram postos em prática:

1. Inicialização do analisador de pacotes LoRa;
2. Inicialização dos nodos sensores;
3. Inicialização do nodo coletor.

Após a inicialização do analisador de pacotes LoRa, ele se encontrará em modo de espera até que um pacote LoRa transmitido seja recebido por ele. Ao receber um sinal LoRa, o analisador de pacotes, imprimirá em um terminal o conteúdo do pacote recebido na sua ordem de envio assim como o tamanho deste pacote e o horário em que ele foi recebido, como pode ser conferido na Figura 41. Além disso, o último byte que aparece nos pacotes pelo analisador de pacotes não é um dado LoRa enviado, mas sim o RSSI (*Received Signal Strength Indicator*), o qual é uma medida do quão bom está o enlace de RF. É um valor que é útil para determinar se o sinal de rádio é suficiente para obter uma boa conexão.

Já em relação aos nodos sensores, uma vez que o nodo coletor realiza o *advertisement*, ambos entrarão em processo de pleitear o *time slot* da rede que transmitido neste anúncio. Assim, após terem configurado seus RTCs com a data e hora recebida no primeiro *advertisement* e realizado as correções de sincronização, ambos nodos estarão prontos para transmitir suas informações no momento em que o *time slot* pleiteado chegar. É neste momento que o mecanismo de cancelamento de transmissão, descrito na Seção 4.2.2.4, entra em operação.

Figura 41: Sinais recebidos em analisador de pacotes LoRa durante teste conexão concorrente de nodos sensores.

```

-----
- Serial Port = /dev/ttyACM0
-----
Waiting for data...
[19:21:26] Packet Received (Size = 15)
-----
| 0 1 2 3 4 5 6 7 8 9
-----
0 | 0 0 0 0 0 0 0 0 52 10
1 | 5 19 20 30 74
-----
[19:22:26] Packet Received (Size = 15)
-----
| 0 1 2 3 4 5 6 7 8 9
-----
0 | 0 0 0 0 0 0 85 0 52 10
1 | 5 19 21 30 68
-----
[19:22:56] Packet Received (Size = 7)
-----
| 0 1 2 3 4 5 6 7 8 9
-----
0 | 0 0 170 216 61 99 44
-----
[19:23:26] Packet Received (Size = 15)
-----
| 0 1 2 3 4 5 6 7 8 9
-----
0 | 1 0 170 216 61 99 85 0 52 10
1 | 5 19 22 30 62
-----
[19:23:56] Packet Received (Size = 7)
-----
| 0 1 2 3 4 5 6 7 8 9
-----
0 | 0 0 170 216 61 99 47
-----
[19:23:57] Packet Received (Size = 7)
-----
| 0 1 2 3 4 5 6 7 8 9
-----
0 | 1 0 191 216 61 99 58
-----
[19:24:26] Packet Received (Size = 15)
-----
| 0 1 2 3 4 5 6 7 8 9
-----
0 | 2 0 191 216 61 99 85 0 52 10
1 | 5 19 23 30 68
-----
[19:24:56] Packet Received (Size = 7)
-----
| 0 1 2 3 4 5 6 7 8 9
-----
0 | 0 0 170 216 61 99 44
-----
[19:24:57] Packet Received (Size = 7)
-----
| 0 1 2 3 4 5 6 7 8 9
-----
0 | 1 0 191 216 61 99 58
-----
[19:25:26] Packet Received (Size = 15)
-----
| 0 1 2 3 4 5 6 7 8 9
-----
0 | 2 0 191 216 61 99 85 0 52 10
1 | 5 19 24 30 76
-----

```

Fonte: Autor.

Primeiramente, observando a Figura 41 pode-se perceber que os pacotes estão sendo enviados com o conteúdo de acordo com o que foi definido nas Seções 4.2.1.2 e 4.2.2.2, o *advertisement* representado pelos pacotes de tamanho 15 e a transmissão dos nodos sensores os pacotes de tamanho 7. Na Figura 41, os primeiros dois pacotes de ADV apresentam os bytes 2...5 equivalentes ao último nodo que tentou se juntar a rede zerado, pois nenhum nodo sensor realizou transmissão e se juntou a rede até então. Similar pode ser comentado sobre os bytes 6...7 para o primeiro ADV, pois como não houve nenhuma mensagem de

advertisement anteriormente, o valor transmitido nesses bytes, o qual equivale ao tempo de transmissão do *advertisement*, são zero. Nota-se na Figura 41, que duas mensagens de *advertisement* ocorrem e em seguida apenas um nodo sensor transmite, isso significa que um dos nodos sensores transmitiu antes do outro. Este outro nodo recebeu esta transmissão de nodo sensor para o *time slot* que estava pleiteando e cancelou o seu processo de alocação daquele *time slot*, voltando assim ao modo de receptor esperando por um novo *advertisement*. Diante disto, o nodo sensor que transmitiu se conecta a rede após verificar em um próximo *advertisement* que o ID do último nodo conectado é o seu próprio.

Em seguida, observa-se que o nodo coletor incrementa o *time slot* anunciado para 1, pois o *time slot* 0 foi ocupado por um nodo sensor. Paralelo a isso, o nodo sensor que abortou sua transmissão passa a realizar o processo de conexão a rede pleiteando o *time slot* 1. Por fim ambos nodos sensores se conectam na rede e passam a transmitir sincronicamente no *time slot* ao qual se ligaram.

5.4 Validação do mecanismo de *timeout*

5.4.1 Validação do *timeout*: Nodo sensor

Para se testar o *timeout* dos nodos sensores no nodo coletor, se utilizará do analisador de pacote LoRa tal como especificado na Seção 5.3. Para esse teste, foram utilizado dois nodos, um em modo coletor e outro em modo sensor. Assim, o seguinte ensaio foi realizado:

1. Inicialização do analisador de pacotes LoRa;
2. Inicialização dos nodos sensores;
3. Inicialização do nodo coletor;
4. No momento em que o nodo sensor se juntar a rede e realizar sua primeira transmissão, desativá-lo.

Foi definido para este ensaio que o número máximo de transmissões perdidas relacionadas a um *time slot* seria de 4. Após este número de faltas, o *time slot* em questão seria anunciável novamente.

Figura 42: Sinais recebidos em analisador de pacotes LoRa durante teste de *timeout* em nodo coletor.

```

-----
- Serial Port = /dev/ttyACM0
-----

Waiting for data...

[19:53:26] Packet Received (Size = 15)
-----
| 0 1 2 3 4 5 6 7 8 9
-----
0| 0 0 0 0 0 0 0 0 52 10
1| 5 19 52 30 74
-----

[19:54:26] Packet Received (Size = 15)
-----
| 0 1 2 3 4 5 6 7 8 9
-----
0| 0 0 0 0 0 0 84 0 52 10
1| 5 19 53 30 74
-----

[19:54:56] Packet Received (Size = 7)
-----
| 0 1 2 3 4 5 6 7 8 9
-----
0| 0 0 91 224 61 99 60
-----

[19:55:26] Packet Received (Size = 15)
-----
| 0 1 2 3 4 5 6 7 8 9
-----
0| 1 0 91 224 61 99 84 0 52 10
1| 5 19 54 30 74
-----

[19:56:25] Packet Received (Size = 15)
-----
| 0 1 2 3 4 5 6 7 8 9
-----
0| 1 0 91 224 61 99 84 0 52 10
1| 5 19 55 30 74
-----

[19:57:25] Packet Received (Size = 15)
-----
| 0 1 2 3 4 5 6 7 8 9
-----
0| 1 0 91 224 61 99 84 0 52 10
1| 5 19 56 30 74
-----

[19:58:25] Packet Received (Size = 15)
-----
| 0 1 2 3 4 5 6 7 8 9
-----
0| 1 0 91 224 61 99 84 0 52 10
1| 5 19 57 30 73
-----

[19:59:25] Packet Received (Size = 15)
-----
| 0 1 2 3 4 5 6 7 8 9
-----
0| 0 0 0 0 0 0 84 0 52 10
1| 5 19 58 30 74
-----

```

Fonte: Autor.

Como pode ser observado na Figura 42, o mecanismo de *timeout* no nodo coletor está funcionando plenamente, pois logo após o quarto *advertisement* sem receber nenhuma transmissão do nodo sensor o *time slot* (bytes 0...1) do nodo sensor passa a ser disponível para ser anunciado novamente. Isto pode ser verificado quando se olha para o último *advertisement* que ao invés de continuar anunciando o *time slot* 1, passa a anunciar o *time slot* 0 que se tornou livre.

5.4.2 Validação do *timeout*: Nodo sensor

Para se testar o *timeout* do nodo coletor no nodo sensor, se utilizará do analisador de pacote LoRa tal como especificado na Seção 5.3. Para esse teste, foram utilizado três nodos, o primeiro em modo coletor, o segundo em modo sensor e terceiro nodo foi compilado um *firmware* que envia um pacote LoRa de tamanho 1 a cada 60 segundos com os mesmos parâmetros LoRa do nodo sensor e coletor. Este terceiro nodo genérico será utilizado com o intuito de oferecer um parâmetro no analisador de pacotes para observar o momento em que o nodo sensor realizou *timeout* do nodo coletor. Assim, o seguinte ensaio foi realizado:

1. Inicialização do analisador de pacotes LoRa;
2. Inicialização dos nodos sensores;
3. Inicialização do nodo coletor e nodo genérico;
4. No momento em que o nodo sensor se juntar a rede e realizar sua primeira transmissão, desativar o nodo coletor.

Foi definido para este ensaio que o número máximo de *advertisements* perdidos relacionadas a um *time slot* seria de 3. Após este número de faltas, nodo sensor retornaria ao estado de receptor em espera por um novo *advertisement*.

Figura 43: Sinais recebidos em analisador de pacotes LoRa durante teste de *timeout* em nodo sensor.

```

-----
- Serial Port = /dev/ttyACM0
-----

Waiting for data...

[20:02:25] Packet Received (Size = 15)
-----
| 0 1 2 3 4 5 6 7 8 9
-----
0| 0 0 0 0 0 0 84 0 52 10
1| 5 20 1 30 74
-----

[20:03:13] Packet Received (Size = 1)
-----
| 0 1 2 3 4 5 6 7 8 9
-----
0| 0 55
-----

[20:03:25] Packet Received (Size = 15)
-----
| 0 1 2 3 4 5 6 7 8 9
-----
0| 0 0 0 0 0 0 85 0 52 10
1| 5 20 2 30 74
-----

[20:03:56] Packet Received (Size = 7)
-----
| 0 1 2 3 4 5 6 7 8 9
-----
0| 0 0 115 226 61 99 53
-----

[20:04:13] Packet Received (Size = 1)
-----
| 0 1 2 3 4 5 6 7 8 9
-----
0| 0 54
-----

[20:04:56] Packet Received (Size = 7)
-----
| 0 1 2 3 4 5 6 7 8 9
-----
0| 0 0 115 226 61 99 58
-----

[20:05:13] Packet Received (Size = 1)
-----
| 0 1 2 3 4 5 6 7 8 9
-----
0| 0 54
-----

[20:05:56] Packet Received (Size = 7)
-----
| 0 1 2 3 4 5 6 7 8 9
-----
0| 0 0 115 226 61 99 57
-----

[20:06:13] Packet Received (Size = 1)
-----
| 0 1 2 3 4 5 6 7 8 9
-----
0| 0 58
-----

[20:07:13] Packet Received (Size = 1)
-----
| 0 1 2 3 4 5 6 7 8 9
-----
0| 0 56
-----

```

Fonte: Autor.

Observa-se na Figura 43 que logo após o nodo sensor se conectar a rede e transmitir

pela primeira vez, os *advertisements* deixam de ser recebidos. Verifica-se que o mecanismo de *timeout* no nodo sensor está funcionando plenamente, pois após a interrupção na recepção de *advertisements*, o nodo sensor transmite três vezes e deixa de transmitir. É possível verificar que o nodo sensor deixa de transmitir pois o nodo genérico continua enviando suas mensagens periódicas e entre elas não há nenhuma outra transmissão com os mesmos parâmetros LoRa.

6 CONCLUSÃO

Este trabalho tomou como ponto de partida o que foi produzido em (VALADÃO, 2021), o qual propunha uma solução rede de sensores sem fio LoRa para o monitoramento do comportamento reprodutivo de bovinos. O que foi desenvolvido pode se ver como uma extensão do que foi explorado em (VALADÃO, 2021), já que a RSSF proposta para aplicação bovina apresentava grande dependência manual para que a rede fosse inicializada e para que o sincronismo da comunicação entre nodos sensores fosse manejado. Neste trabalho propôs-se uma solução a esta dependência manual, desenvolvendo um *firmware* capaz de inicializar uma rede sensores sem fio, realizar o gerenciamento de seus *time slots* e corrigir assincronismos nas transmissões entre nodos dado os componentes de hardware selecionados.

Para a sincronização, inicialmente se foram estimado cálculos de correção, representados pela Equação 8, de sincronismo *time slot* entre nodo coletor e nodo sensor devido a imprecisão do RTC DS1307 quando se trata de configurar os milissegundos. No entanto, no Capítulo 5, pode-se observar que o calculo de correção de sincronismo precisava ser ajustado pois o resultado não era satisfatório. Após ajustes relacionados ao tempo de transmissão LoRa e de pausa do RTC foi possível chegar em uma Equação 14, cujo erro de sincronismo era suficientemente pequeno, na casa de alguns milésimos de segundos em uma escala de segundos. Logo, o mecanismo de correção de sincronismo foi bem sucedido. Junto a isso também foi implementado um mecanismo de gerenciamento de *time slots* o qual contabiliza quais *time slots* estão recebendo transmissão e quais vão ser anunciados para nodos sensores pleitearem se conectarem na rede. Como foi discutido e pode ser observado na Figura 40 da Seção 5.2 e nas Figuras 42 e 43 da Seção 5.4, o gerenciador de *time slots* desempenha com sucesso a alocação de *time slots* para nodos e suas redistribuições para os *advertisements*.

Em relação ao mecanismo de *timeout*, como pode ser observado na Figura 42 relacionada ao nodo coletor da Seção 5.4.1, após a passagem do número limites de não recebimentos de transmissões de nodos sensores alocados, o nodo coletor é capaz de tornar o *time slot* até então ocupado disponível para ser anunciado mais uma vez. Similar pode ser dito em relação ao *timeout* do nodo sensor, em que na Figura 43 relacionada da Seção 5.4.2, o *timeout* está sendo capaz de identificar que o nodo coletor não está mais presente e de remanejar o nodo sensor em questão para estado de receptor em espera de um novo *advertisement* para pleitear se juntar a rede novamente.

Como ressaltado na Seção 4.2.2.2, a implementação dos sensores não cabe ao escopo do projeto, logo como o nodo LoRa desenvolvido tem o propósito de ser genérico para múltiplas implementações, um dos próximos passos de seu aprimoramento seria atribuir uma cadeia de sensores instrumentados ao seu sistema. Paralelo a isso, a otimização dos parâmetros da camada física LoRa não foram explorados neste trabalho, desta forma o

seu desenvolvimento traria maior eficiência energética e robustez para a rede de sensores sem fio. Por fim, implementação da camada de enlace LoRaWAN poderá agregar valor ao projeto.

REFERÊNCIAS

- ANATEL. Resolução nº 454, de 11 de dezembro de 2006. aprova o regulamento sobre condições de uso de radiofrequências nas faixas de 800 mhz, 900 mhz, 1.800 mhz, 1.900 mhz e 2.100 mhz. *Diário Oficial da União*, 2006.
- ARDUINO. *Arduino® Nano*. [S.l.: s.n.], 2022. <https://docs.arduino.cc/hardware/nano>.
- BOR, M.; ROEDIG, U. LoRa transmission parameter selection. In: IEEE. 2017 13th International Conference on Distributed Computing in Sensor Systems (DCOSS). [S.l.: s.n.], 2017. P. 27–34.
- BOR, M.; VIDLER, J. E.; ROEDIG, U. LoRa for the Internet of Things. Junction Publishing, 2016.
- COMITTEE, L. A. T. *LoRaWAN Regional Parameters V1.0.2b*. [S.l.: s.n.], 2017. https://lora-alliance.org/resource_hub/lorawan-regional-parameters-v1-0-2rb/.
- DEVALAL, S.; KARTHIKEYAN, A. LoRa technology-an overview. In: IEEE. 2018 second international conference on electronics, communication and aerospace technology (ICECA). [S.l.: s.n.], 2018. P. 284–290.
- HOPERF. *RFM95W - LoRa Module*. [S.l.: s.n.], 2020. <https://www.hoperf.com/modules/lora/RFM95.html>.
- LEENS, F. An introduction to I2C and SPI protocols. *IEEE Instrumentation & Measurement Magazine*, IEEE, v. 12, n. 1, p. 8–13, 2009a.
- LEENS, F. An introduction to I2C and SPI protocols. *IEEE Instrumentation & Measurement Magazine*, v. 12, n. 1, p. 8–13, 2009b. DOI: 10.1109/MIM.2009.4762946.
- LINK LABS, I. *A Comprehensive Look at Low Power, Wide Area Networks*. [S.l.: s.n.], 2016. <http://cdn2.hubspot.net/hubfs/427771/LPWAN-Brochure-Interactive.pdf>.
- MICROCHIP. *megaAVR® Data Sheet*. [S.l.: s.n.], 2020. <http://ww1.microchip.com/downloads/en/DeviceDoc/ATmega48A-PA-88A-PA-168A-PA-328-P-DS-DS40002061A.pdf>.
- MIGABO, E. et al. A comparative survey study on LPWA networks: LoRa and NB-IoT. In: PROCEEDINGS of the Future Technologies Conference (FTC), Vancouver, BC, Canada. [S.l.: s.n.], 2017. P. 29–30.
- NARRATIVE. *What is Unix time?* [S.l.: s.n.]. <https://kb.narrative.io/what-is-unix-time>.

- RAZA, U.; KULKARNI, P.; SOORIYABANDARA, M. Low Power Wide Area Networks: An Overview. *IEEE Communications Surveys & Tutorials*, v. 19, n. 2, p. 855–873, 2017. DOI: 10.1109/COMST.2017.2652320.
- SEMICONDUCTOR. *DS1307 datasheet*. [S.l.: s.n.], 2020. <https://www.alldatasheet.com/datasheet-pdf/pdf/123888/DALLAS/DS1307.html>.
- SEMICONDUCTORS, N. S12SPIV4/D SPI Block Guide. *Block Guide*, v. 04.01, 2004.
- SEMICONDUCTORS, N. UM10204 I2C-bus specification and user manual. *User Manual*, v. 4, 2014.
- SEMTECH. *AN1200.22 : LoRa Modulation Basics*. [S.l.: s.n.], 2015. <https://www.frugalprototype.com/wp-content/uploads/2016/08/an1200.22.pdf>.
- SEMTECH. *SX1276/77/78/79 - 137 MHz to 1020 MHz Low Power Long Range Transceiver*. [S.l.: s.n.]. <https://www.semtech.com/products/wireless-rf/lora-connect/sx1276>.
- SIGFOX. [S.l.: s.n.]. <https://www.sigfox.com/en>.
- SINHA, R. S.; WEI, Y.; HWANG, S.-H. A survey on LPWA technology: LoRa and NB-IoT. *Ict Express*, Elsevier, v. 3, n. 1, p. 14–21, 2017.
- SMITH, S. W. et al. *The scientist and engineer's guide to digital signal processing*. [S.l.]: California Technical Pub. San Diego, 1997.
- VALADÃO, Y. D. N. *Sistema de monitoramento do comportamento de bovinos por meio de redes de sensores sem fio*. [S.l.: s.n.], 2021. <https://www.lume.ufrgs.br/handle/10183/223797?show=full>.
- WU, J. Basics of I2C: The I2C Protocol. In: TEXAS INSTRUMENTS. TI Precision Labs – Digital Communications. [S.l.: s.n.], out. 2021.