

Desenvolvimento de sistema SCADA utilizando o protocolo LoRaWan para a transmissão de dados

Victor de Oliveira

Discente do Curso de Engenharia Mecatrônica
Universidade Federal de São João del-Rei
victor-vdo@bol.com.br

Guilherme Gomes da Silva

Departamento das Engenharias de Telecomunicações e
Mecatrônica
Universidade Federal de São João del-Rei
guilhermegomes@ufsj.edu.br

Resumo — A aquisição, supervisão e o controle de dados são parâmetros importantes na construção de um sistema capaz de controlar as variáveis envolvidas em sua manipulação. Neste artigo há o processo de desenvolvimento de um sistema supervisório. Ele utiliza uma tecnologia de transmissão a rádio conhecida como LoRaWan. Ela será utilizada para transmitir os dados dos sensores para o sistema e os comandos de ação para os atuadores e demais dispositivos responsivos.

Palavras-chave — Sistema supervisório, SCADA, LoRaWan, C Sharp, LPWAN.

I. INTRODUÇÃO

Durante a história da comunicação humana houve uma frequente busca por meios que realizam troca de informação mais eficientes. Desta forma, foram criados diversos protocolos que focam numa alta taxa de transmissão de dados por segundo, como as redes wi-fi. Outros protocolos se destacam pelo baixo custo energético ao transmitir dados, como as redes desenvolvidas para a Internet das Coisas (IoT).

Este trabalho possui a função de desenvolver um sistema SCADA capaz de controlar e supervisionar um sistema automatizado. Será utilizado uma destas tecnologias de transmissão de informação a longa distância, o protocolo LoRaWan.

Este protocolo é assim chamado devido ao seu alto alcance de propagação e recepção (Long Range) e ao seu baixo consumo energético para desempenhar essa função. Em contraponto a estas vantagens, ele possui uma baixa taxa de transmissão, pois não é possível enviar ou receber muitos pacotes de dados ao mesmo tempo.

Por este motivo este protocolo será utilizado para a transmissão de informações entre o sistema e o micro controlador. Devido ao baixo número de bytes envolvendo as informações transmitidas dos sensores para o supervisório.

No decorrer do projeto contido neste artigo analisaremos a viabilidade da utilização da LoRa no âmbito industrial ou demais aplicações profissionais.

II. PROJETO DE SISTEMA SUPERVISÓRIO

Após o desenvolvimento tecnológico dos microcontroladores e microprocessadores na década de 70 e 80 [1] foi possível estabelecer sistemas mais complexos de supervisão, controle e aquisição de dados (SCADA).

Estes sistemas apresentam ao usuário, gráficos, tabelas e demais dados referentes ao conjunto de elementos eletrônicos que compõem os sensores do agrupamento automático. Além disso, é possível realizar um controle dos sistemas automáticos através de funções que setam variáveis e/ou parâmetros.

Através destes programas é feito o monitoramento das informações conhecidas como variáveis de processo que são enviadas pelos sensores. Após a recepção destas informações é possível realizar as medidas decisórias que atuam na ação das saídas digitais e/ou analógicas. Estas medidas variam de acordo com as estruturas de seleção presentes nos códigos pré-programados para os microcontroladores e/ou microprocessadores.

O sistema supervisório proposto neste artigo será implementado utilizando o *software* Visual Studio através da linguagem de programação C Sharp (C#). Sua função é mostrar a camada de apresentação ao usuário através de uma interface gráfica dinâmica, intuitiva e interativa. Além disso, também será possível receber e enviar os valores referentes aos sensores através da comunicação serial em tempo real com atualização automática no sistema.

Como dito anteriormente, a conexão entre o microcontrolador ou microprocessador e o sistema será realizada através da utilização de uma porta serial. No C# há uma biblioteca que trabalha diretamente com estas portas, a *System.IO.Ports*.

Ao realizar a ação presente no método *OnClick* contido no pacote padrão de bibliotecas do Visual Studio e acessada pelos botões principais, eles apresentam ao usuário um novo formulário (*form*). Portanto, se a conexão serial for instanciada e aberta no primeiro formulário ela será diferente em cada um deles. Desta forma, seria necessário abrir a conexão toda vez que um novo formulário fosse aberto. Para contornar isso, uma classe de conexão foi criada com o padrão de projeto de *software* chamado *Singleton* [2].

Esse tipo de classe permite que uma única instância seja aberta em todo o projeto. Por esse motivo, não é necessário abrir uma nova instância a cada tela. A classe *Singleton* de conexão está apresentada na Figura 1.

```

1  using System;
2  using System.Collections.Generic;
3  using System.IO.Ports;
4  using System.Linq;
5  using System.Text;
6  using System.Threading.Tasks;
7
8  namespace Apresentacao
9  {
10     public class ConexaoSerial
11     {
12         private ConexaoSerial() { }
13         private static ConexaoSerial instancia;
14         public SerialPort conexao = new SerialPort();
15         public static ConexaoSerial Instancia
16         {
17             get
18             {
19                 if(instancia == null)
20                 {
21                     instancia = new ConexaoSerial();
22                 }
23                 return instancia;
24             }
25         }
26         public SerialPort GetConexao()
27         {
28             return conexao;
29         }
30     }
31 }
32

```

Figura 1 : Classe singleton de conexão serial

Após estabelecer como será a conexão entre o supervisor e o sistema microprocessado, é necessário estabelecer como será o padrão de transmissão de dados entre eles.

No projeto todas as informações serão enviadas e recebidas através de *strings*. Elas são tipos de dados formados por um conjunto de caracteres. Será necessário que o supervisor interprete estas *strings* e retorne para o usuário de forma visual os dados apresentados. Além disso, é preciso que o sistema microprocessado/microcontrolado possa interpretar as *strings* geradas pelos elementos de controle do supervisor. Portanto, ambos os sistemas já devem estar pré-configurados para a recepção e envio destas *strings*.

Para a recepção dos dados, foi criada uma instância do método que analisa cada palavra recebida pela porta serial. Este método é o *ReadExisting()* e está presente na biblioteca *System.IO*. A Figura 2 mostra o método *DataReceivedHandler* que contém a leitura da porta serial através do evento *SerialDataReceivedEventArgs*. Ele seta o parâmetro de leitura no método que realiza a ação de acordo com a *string* recebida.

```

1  private void DataReceivedHandler(object sender, SerialDataReceivedEventArgs e)
2  {
3      string indata = ConexaoSerial.Instancia.GetConexao().ReadExisting();
4      SetTexts(indata);
5  }

```

Figura 2: Método *DataReceivedHandler*.

O supervisor apresenta 5 telas que contêm alguns dos componentes necessários para a aquisição e envio de dados ao sistema microcontrolado contendo os sensores e atuadores.

Cada tela está vinculada a um botão contido num conjunto de elementos chamado menu que está presente na lateral esquerda da tela principal. Ele permanece ativo sem interrupção, contudo, ao iniciar o sistema é necessário autenticar o acesso do usuário informando o login e senha pré-configurados no sistema SCADA.

O sistema possui uma tela principal que não varia seus componentes exceto no painel central. O método *OnClick* acessado através dos botões presentes no menu lateral esquerdo substitui este painel pelos *Windows Forms* referentes às telas do sistema. A Figura 3 demonstra o método de substituição do painel central pelos formulários.

```

1  private void AbrirFormFilho(object formFilho)
2  {
3      // Descobre se existe algum componente no interior do painel e remove caso exista
4      if (pnlCentralInicial.Controls.Count > 0)
5          this.pnlCentralInicial.Controls.RemoveAt(0);
6      Form filho = formFilho as Form;
7      filho.TopLevel = false; // Define que não é um formulário de nível superior
8      filho.Dock = DockStyle.Fill; // Define que o form preencha todo o painel
9      this.pnlCentralInicial.Controls.Add(filho); // Adiciona o form dentro do pnlCentralInicial
10     this.pnlCentralInicial.Tag = filho;
11     filho.Show(); // mostra o novo formulário
12     pnlCentralInicial.Refresh(); // atualiza o painel central
13 }

```

Figura 3: Método que abre um windows forms dentro de um panel.

A tela inicial possui um painel de login que dá acesso ao sistema de modo geral, pode-se verificar sua ilustração na Figura 4. Sem o acesso não é possível ter informações de nenhuma outra tela, nem conectar ao sistema microcontrolado através da conexão serial.



Figura 4: Tela inicial.

Cada tela apresenta uma função diferente do sistema. Na segunda tela é possível analisar graficamente as informações dos sensores. À medida que os sensores enviam os dados para o microcontrolador e estes enviam as *strings* para a porta

serial, o supervisório reconhece o padrão de *string* e adiciona o sinal ao gráfico.

É possível observar na Figura 5 a tela referente ao Dashboard. Esta tela contém os gráficos gerados em tempo real de acordo com as medidas dos sensores enviadas para o supervisório.

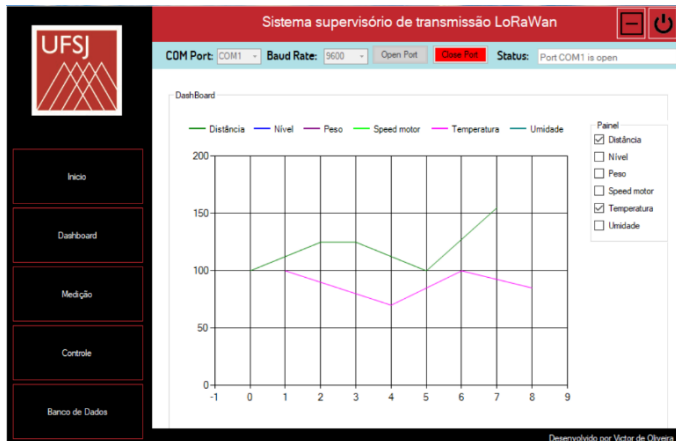


Figura 5: Tela do Dashboard.

A terceira tela refere-se aos módulos de medição. Nele é possível encontrar algumas áreas para a produção e representação de textos resumidos e/ou frases, chamados *text boxes*. Eles são componentes da biblioteca *System.Windows.Forms* capazes de apresentar as informações recebidas pelo sistema de sensoriamento de forma gráfica.

```
1 public void SetTexts(string value)
2 {
3     if (ConexaoSerial.Instancia.GetConexao().IsOpen == true)
4     {
5         if (InvokeRequired)
6             try
7             {
8                 Invoke(new SetTextDelegate(SetTexts), value);
9             }
10            catch { }
11        else
12        {
13            try
14            {
15                if (value.Substring(0, 1).Equals("[")
16                {
17                    // Seta a temperatura
18                    if (value.Substring(1, 1).Equals("T") && value.Substring(2, 1).Equals("P"))
19                    {
20                        txbTemperatura.Text = value.Replace("[TP", "").Replace("]", "");
21                        pnlModuloTemperatura.BackColor = Color.Aquamarine;
22                    }
23                }
24                else
25                { pnlModuloTemperatura.BackColor = Color.White; }
```

Figura 6 : Trecho do método que mostra nos campos de medição, as informações recebidas pela porta serial.

Quando a *string* é recebida pelo sistema SCADA, sua apresentação ao usuário dependerá dos caracteres iniciais que a compõem. Na figura 3 há o método que seta os textos nos *Text Boxes* do formulário Medição. Quando a *string* começa com o colchete “[” o sistema reconhece que as próximas *substrings* serão referentes a este formulário. As próximas 2 *substrings* determinarão qual o sensor será lido. São elas: “TP” para a temperatura, “UM” para a umidade, “DT” para a distância, “PS” para o peso, “SM” para a velocidade do motor e “NV”

para o nível. Os valores referentes aos sensores virão a seguir e a *substring* “]” fecha a *string*.

Por exemplo:

Caso o supervisório receba na porta serial a *string* [DT350], ele vai reconhecer que se trata de um dado de medição, pois começa com o colchete “[”, reconhecerá também que se trata de um dado do tipo Distância, pois as duas *substrings* subsequentes são “DT” e que possui o valor de 350, que no caso da distância, será em centímetros.

A Figura 7 apresenta a tela do supervisório que mostra ao usuário a apresentação gráfica dos valores referentes a cada sensor.

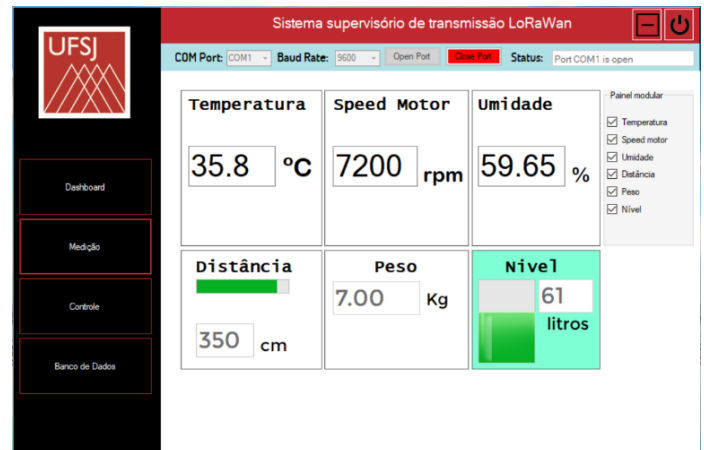


Figura 7: Tela dos elementos de medição.

A quarta tela representada pela Figura 8 contém o sistema de controle do supervisório. Nele é possível realizar ações diretamente nos atuadores conectados ao setor automático.

Nas telas de Dashboard, Medição e Controle há o mesmo padrão de recebimento de *strings* representado na Figura 6. Suas diferenças variam de acordo com os métodos de resposta e as funções de cada componente que compõem a tela.



Figura 8: Tela de controle.

A quinta e última tela mostra uma forma de manter os dados adquiridos pelos sensores em uma planilha do Excel. Nela está

sendo usada a biblioteca *System.IO.File* para gerar o arquivo que armazena os valores que serão salvos para a planilha. É possível implementar no sistema uma conexão direta com um banco de dados mais robusto, contudo, esse não é o foco deste trabalho. Portanto, esta poderá ser uma futura implementação ou complementação de acordo com o banco de dados mais interessante para o usuário. A tela do supervisorio referente a banco está representada pela Figura 9.

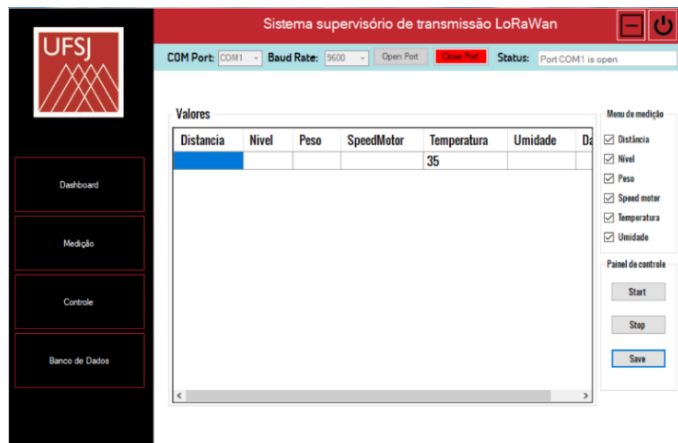


Figura 9: Tela de repositório de banco de dados.

III. PROTOCOLO LORAWAN – TEORIA E APLICAÇÃO

Em um meio industrial é comum que haja a necessidade de transmissão de informações a longa distâncias. Isso se deve ao fato da longa extensão de diversos meios de produção como por exemplo, em um setor metalúrgico. Por esse motivo, foram criadas diversas tecnologias que buscam realizar a comutação dessas informações, porém, todas possuem vantagens e desvantagens que podem inviabilizar ou viabilizar a sua aplicação num determinado setor. Neste tópico há apresentação de algumas dessas tecnologias com o foco no protocolo LoRaWan, que foi utilizado no projeto.

Devido ao conceito que atualmente é discutido mundialmente, conhecido como Internet das Coisas (IoT), algumas empresas estão desenvolvendo aplicativos e serviços que necessitam de diversos dispositivos com baixo consumo de energia para serem interconectados [9]. Por esse motivo, algumas alternativas utilizadas para a comunicação entre usuários e objetos com capacidade de sensoriamento, processamento e comunicação estão em processo de desenvolvimento. Nas últimas décadas houve o interesse de diversas organizações que visam a padronização destas tecnologias. O foco desta padronização está ligado à comunicação em poucos quilômetros e na criação de redes de longa distância e baixa potência (LPWANs) que não exijam a construção e manutenção de topologias complexas *multi-hop* [3].

Uma das características da tecnologia LPWAN é a capacidade de trocar informações através de uma taxa de transferência por faixa, ou seja, é possível ajustar as

configurações da camada física (PHY) para selecionar uma opção mais sensível, portanto, mais lenta. Contudo, ela permite uma comunicação com maior alcance. Entre as tecnologias emergentes estão a SigFox, LoRa, Weightless e NB-IoT.

A Sigfox é uma das primeiras tecnologias *Low Power Wide Area* (LPWAN) propostas para IoT. Criada em 2009 pela *SmartCore*, ela utiliza em sua camada física o método de modulação sem fio *Ultra Narrow Band* (UNB). [4] Não há uma especificação na sua documentação sobre funcionamento de sua camada de rede, pois isto faz parte de seu modelo de negócio. Possui uma topologia estrela e a maior parte do tratamento de dados é efetuado pelo centro de processamento para qual seus dados são enviados. Ela possui uma taxa de transmissão de cerca de 12 bytes.[4]

A *Weightless* foi desenvolvida pela organização global sem fins lucrativos “*WeightlessSpecialInterestGroup*”, que propõe três padrões abertos da LPWA, cada um oferecendo recursos, alcance e consumo de energia diferentes. São esses a *Weightless-W* com modulação 16-QAM e DBPSK, *Weightless-N* com comunicação em apenas um sentido e a *Weightless-P* [5] que utiliza um esquema de modulação de banda estreita, oferecendo um recurso de comunicação bidirecional. Embora possa operar em qualquer banda de frequência, atualmente este padrão de comunicação é definido para operação em bandas de frequência sub-GHz isentas de licença [6] (por exemplo, 138MHz, 433MHz, 470MHz, 780MHz, 868MHz, 915MHz, 923MHz). A Figura 10 apresenta algumas especificações técnicas dos três padrões.

Standard	WEIGHTLESS-W	WEIGHTLESS-SIG WEIGHTLESS-N	WEIGHTLESS-P
Modulation	16-QAM, BPSK, QPSK, DBPSK	UNB DBPSK	GMSK, offset-QPSK
Band	TV white spaces 470-790MHz	ISM SUB-GHZ EU (868MHz), US (915MHz)	SUB-GHZ ISM or licensed
Data rate	1 kbps-10 Mbps	30 kbps-100 kbps	200 bps-100kbps
Range	5 km (URBAN)	3 km (URBAN)	2 km (URBAN)
Num. of channels / orthogonal signals	16 or 24 channels(UL)	multiple 200 Hz channels	multiple 12.5 kHz channels
Forward error correction	✓	✗	✓
MAC	TDMA/FDMA	slotted ALOHA	TDMA/FDMA
Topology	star	star	star
Payload length	> 10B	20B	> 10B
Authentication & encryption	AES 128b	AES 128b	AES 128/256b

Figura 10: Especificações técnicas da rede *Weightless*. [3]

Desenvolvida pela 3GPP, a NB-IoT é uma tecnologia que possui o foco em dispositivos celulares móveis. Ela opera em *downlink* dentro de 180 kHz (um PRB) e um espaço subportadora de 15 kHz e em *uplink*, normalmente com uma transmissão de tom único de 3,75 kHz ou 15 kHz [7]. Entre as possíveis aplicações estão as industriais de rastreamento de ativos, monitoramento de segurança, medição de gás e água, *smart grids*, estacionamentos, máquinas de venda e iluminação pública.

Assim como a Sigfox, a LoRa é uma plataforma de comunicação a longo alcance sem fio que possui uma topologia estrela na qual os *gateways* retransmitem mensagens entre os dispositivos finais e um servidor de rede central. Os *gateways* são conectados ao servidor de rede por meio de conexões IP padrão e agem como uma ponte transparente, simplesmente convertendo pacotes de RF em pacotes de IP e vice-versa [8]. A Figura 11 mostra como a arquitetura de rede da LoRaWan é composta estabelecendo as conexões entre os sensores e os servidores de aplicações.

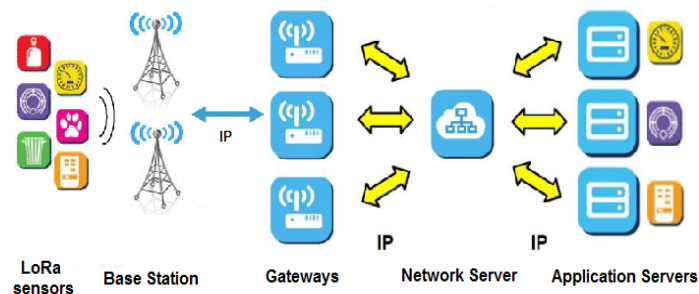


Figura 11: Arquitetura de rede LoRaWan. [11]

Os módulos sensores LoRa, também chamados de *end-devices* são os elementos básicos de rede como os módulos de sensoriamento, tais como, sensores de temperatura, de gás, de iluminação, de corrente elétrica, *switchs on-off*, etc.

As estações de base possuem a função de amplificação do sinal de transmissão das informações dos sensores para os *gateways* e/ou para outros dispositivos LoRa.

Os *gateways* são os receptores dos sinais enviados pelos módulos. Eles são responsáveis por intermediar a comunicação entre os módulos e os servidores de rede. Um *gateway* pode receber informações de diversos módulos sensores [8].

Os servidores de rede gerenciam as informações recebidas pelos *gateways*. Eles são responsáveis pelo encapsulamento e a filtragem das informações, gerando uma maior segurança dos dados.

As aplicações focam no tratamento da apresentação dos dados ao usuário. Ela converte o formato do dado recebido pelos servidores para um formato mais compreensível.

LoRa é a camada física usada no protocolo LoRaWan. Possui uma operação com baixa energia (cerca de 10 anos de duração da bateria), baixa taxa de dados (27 kbps com fator de espalhamento 7 e canal de 500 KHz ou 50 kbps com FSK) e longo alcance de comunicação (2,5 m em áreas urbanas e 15 Km em áreas suburbanas). No artigo realizado anteriormente denominado LoRa: Tecnologia Emergente para Internet das Coisas [10] foi possível constatar uma transmissão com 200 bytes e alcance de aproximadamente 300 metros numa área urbana utilizando dois chips LoRa SX1278 e o microcontrolador Arduino Uno para o envio das strings.

	Sigfox	LoRaWAN	NB-IoT
Modulation	BPSK	CSS	QPSK
Frequency	Unlicensed ISM bands (868MHz in Europe, 915MHz in North America, and 433MHz in Asia)	Unlicensed ISM bands (868MHz in Europe, 915MHz in North America, and 433MHz in Asia)	Licensed LTE frequency bands
Bandwidth	100 Hz	250 kHz and 125 kHz	200 kHz
Maximum data rate	100 bps	50 kbps	200 kbps
Bidirectional	Limited / Half-duplex	Yes / Half-duplex	Yes / Half-duplex
Maximum messages day	140 (UL), 4 (DL)	Unlimited	Unlimited
Maximum payload length	12 bytes (UL), 8 bytes (DL)	243 bytes	1600 bytes
Range	10 km (urban), 40 km (rural)	5 km (urban), 20 km (rural)	1 km (urban), 10 km (rural)
Interference immunity	Very high	Very high	Low
Authentication & encryption	Not supported	Yes (AES 128b)	Yes (LTE encryption)
Adaptive data rate	No	Yes	No
Handover	End-devices do not join a single base station	End-devices do not join a single base station	End-devices join a single base station
Localization	Yes (RSSI)	Yes (TDOA)	No (under specification)
Allow private network	No	Yes	No
Standardization	Sigfox company is collaborating with ETSI on the standardization of Sigfox-based network	LoRa-Alliance	3GPP

Figura 12: Resumo das principais tecnologias LPWAN. [11]

A comunicação sem fio tira proveito das características de longo alcance da camada física LoRa, permitindo um *link* de salto único entre o dispositivo final e um ou vários *gateways* [9]. A Figura 12 mostra um breve resumo das três tecnologias LPWAN que atualmente são o maior foco do mercado tecnológico deste segmento.

O protocolo LoRaWAN é dividido em três classes A, B e C. A classe A suporta a funcionalidade básica LoRaWAN, obrigatória para todos os dispositivos. Ela permite a comunicação bidirecional. Para a transmissão de *uplink*, os dispositivos usam acesso aleatório sem traços, semelhante ao ALOHA. A transmissão de *downlink* pode ser feita apenas durante intervalos de tempo dedicados chamados janelas de recebimento, que seguem as transmissões de *uplink* bem-sucedidas. Se a carga da rede for baixa, a Classe A fornece o menor consumo de energia para os motores, mas mesmo nesse caso, há longos atrasos no *downlink* [9].

A Classe B implementa comunicação bidirecional com *slots* de recebimento agendados para *downlink*. A disseminação das informações do cronograma é realizada por meio de *beacons* enviados pelo *gateway*. Os dispositivos de classe C “ouvem” o canal continuamente, fornecendo a menor latência de *downlink*, mas exigindo um consumo de energia extremamente alto.

Na Classe A, as comunicações descendentes (ou seja, de um *gateway* para um dispositivo final) são sempre iniciadas pelos dispositivos finais. Após cada transmissão, um dispositivo final abre duas janelas de recepção - RX1 e RX2. Ambos são abertos cerca de 20 micro segundos após o final do *slot* de transmissão. Por padrão, o RX1 usa o mesmo canal de frequência e taxa de dados que o *slot* de transmissão. Ao contrário do RX1, o canal de frequência e a taxa de dados usados no RX2 podem ser negociados entre o *gateway* e o dispositivo final. Se um quadro for recebido pelo dispositivo final durante o RX1, o RX2 não será aberto. Eles são longos o suficiente para detectar uma informação com eficiência. Se uma informação for detectada durante o RX1 (ou RX2), o dispositivo final manterá seu receptor de rádio ativo para receber todo o quadro de informação. [9] Na Figura 13 há a

representação gráfica da transmissão dos *slots* e da recepção pelo RX1 e RX2.

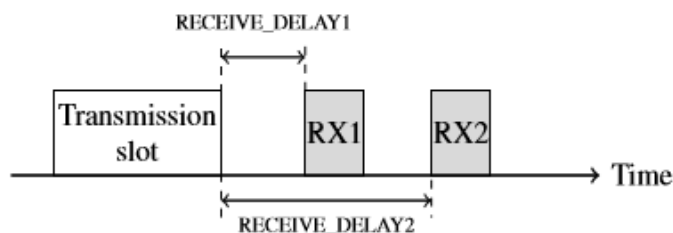


Figura 13: Transmissão e recepção de pacotes da Classe A.[9]

Dispositivos LoRaWan se comunicam com gateways usando a tecnologia LoRa. Ela é baseada em um esquema de modulação que é uma adaptação da modulação *Chirp-Spread Spectrum* (CSS). Inicialmente projetado na década de 1940 para radares, o CSS é amplamente usado em comunicações militares e de segurança devido, pois possuem requisitos de energia de transmissão relativamente baixos e robustez inerente aos efeitos de degradação de canais, como interferências de *multipath*, *fading*, *Doppler* e *in-band jamming* [9].

IV. PROJETO FINAL

No desenvolvimento desse projeto foram utilizadas duas placas de circuitos integrado contendo os chips LoRa SX1278 e uma pequena antena para cada deles. A Figura 14 contém a imagem real do circuito LoRa.



Figura 14: Chip LoRa SX1278.

Estes chips foram integrados ao microcontrolador Arduino UNO para o tratamento das informações recebidas ou enviadas através da LoRa. A Figura 15 mostra o esquemático eletrônico do circuito que contém o micro controlador e o *chip* LoRa SX1278.

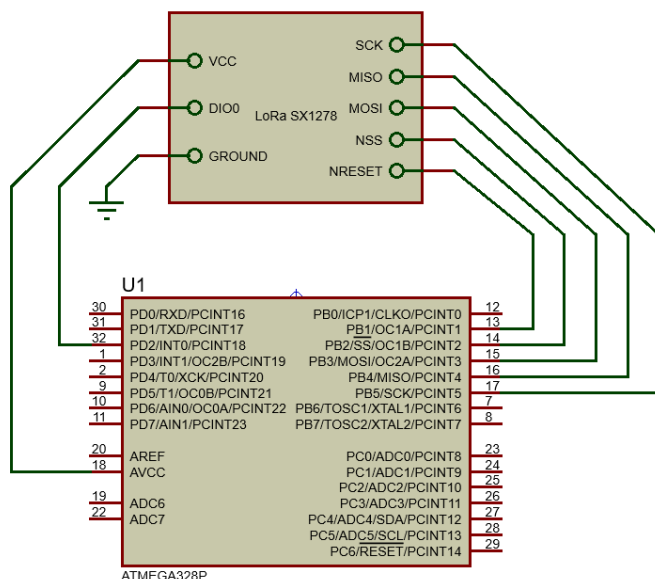


Figura 15: Esquemático eletrônico do protótipo de recepção e emissão LoRaWAN.

O esquemático geral do sistema desenvolvido está representado na Figura 16. Ele demonstra todo o percurso da informação transmitida dos sensores ao supervisor e a resposta do supervisor para o sistema microprocessado.

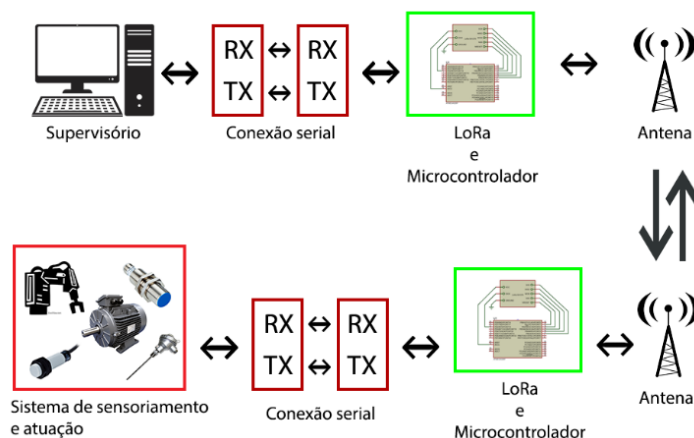


Figura 16: Esquemático geral do sistema desenvolvido.

O sistema microcontrolado deve estar previamente programado para receber as strings de informação e realizar as devidas ações de controle e monitoramento. Na Tabela 1 há uma breve descrição das strings que são enviadas ou recebidas pelo sistema supervisorio através da comunicação serial.

Tabela 1: Strings recebidas e enviadas pelo sistema supervisorio.

String	Descrição
[Indica que o dados posteriores são de medição.
[XX00.00]	Dado de medição onde XX é substituído pela sigla do tipo de informação.
highLedRed/lowLedRed	Acende ou apaga o led vermelho.
highLedGreen/lowLedGreen	Acende ou apaga o led verde.
highLedYellow/lowLedYellow	Acende ou apaga o led amarelo.
highBuzzer/lowBuzzer	Liga ou desliga o alarme.
turnLeft/turnRight	Gira a esteira pra esquerda ou direita.
speedLeft/speedRight	Aumenta a velocidade da esteira pra esquerda ou pra direita
CC : GGG*	Canal (CC) e valor em graus (GGG) enviados para o controle do arm.
sendMessage	Envia strings manualmente para o microcontrolador

V. CONCLUSÃO

A aplicação do protocolo LoRaWan no âmbito industrial apresenta algumas vantagens, a mais importante é o baixo custo energético para atingir um alto alcance, desta forma, ele pode ser utilizado em sistemas modulares sem que haja a necessidade deste sistema estar conectado diretamente na rede de distribuição de energia elétrica.

Portanto, este protocolo aliado a um sistema supervisorio, torna viável sua implantação visto que é possível unir diversas informações de setores distintos de forma rápida e num padrão de segurança customizado pelo próprio conjunto de sistemas.

VI. AGRADECIMENTOS

Agradeço à UFSJ pela estrutura que me deu a oportunidade de desenvolver um perfil profissional. Agradeço a cada professor que contribuiu no meu processo de aprendizado do ciclo básico ao ciclo superior. Além do conhecimento, eles me proporcionaram a oportunidade de me espelhar em profissionais com um alto nível de excelência. Não poderia deixar de agradecer também à minha família, pois sem eles eu não poderia concluir meu curso superior numa instituição tão renomada. Por fim, agradeço à todos os pesquisadores e demais contribuintes nos trabalhos científicos que me ajudaram a construir este artigo através de suas contribuições individuais.

VII. REFERÊNCIAS

- [1] Great moments in microprocessor history. Disponível em: <https://www.ibm.com/developerworks/library/pa-microhist/index.html>, acesso em 07 de outubro de 2019.
- [2] C# - Simple serial port. Disponível em: <http://blog.arwsoft.id/2016/csharp-simple-serialport-singleton-class>, acesso em 12 de outubro de 2019.
- [3] RAZA, Usman. KULKARNI, Parag. SOORIYABANDARA, Mahesh. Low Power Wide Area Networks: An Overview. 2017.
- [4] Uma visão técnica da Rede Sigfox e seus princípios. Disponível em: <https://www.embarcados.com.br/uma-visao-tecnica-da-rede-sigfox/>, acesso em 12 de outubro de 2019.
- [5] Weightless Specification – Weightless Management Ltd. Disponível em: <http://www.weightless.org/about/weightless-specification>, acesso em 12 de outubro de 2019.
- [6] What is Weightless-P ?. Disponível em: <https://www.ubiik.com/post/2016/09/21/what-is-weightless-p>, acesso em 15 de outubro de 2019.
- [7] C3GPP. LTE Progress Leading to the 5G Massive Internet of Things. Dezembro de 2017.
- [8] Conheça a tecnologia LoRa e o protocolo LoRaWan. Disponível em: <https://www.embarcados.com.br/conheca-tecnologia-lora-e-o-protocolo-lorawan/>, acesso em 23 de outubro de 2019.
- [9] BANKOV, Dmitry. KHOROV, Evgeny. LYAKHOVON, Andrey. On the Limits of LoRaWAN Channel Access. Outubro de 2017.
- [10] LoRa: Tecnologia Emergente para Internet das Coisas. Disponível em: <https://github.com/victor-vdo/LORA>. Acesso em 16 de dezembro de 2019.
- [11] MEKKI, Kais; BAJIC, Eddy; CHAXEL, Frederic; MEYER, Fernand. A comparative study of LPWAN technologies for large-scale IoT deployment. 2018.