



MATHEUS APARECIDO MELO FREIRE

**PROJETO IOT PARA CONTROLE REMOTO DE
ILUMINAÇÃO E AR CONDICIONADO E
MONITORAMENTO DO CONSUMO ENERGÉTICO DE
SALAS DE AULA**

**LAVRAS - MG
2019**

MATHEUS APARECIDO MELO FREIRE

**PROJETO IOT PARA CONTROLE REMOTO DE ILUMINAÇÃO E AR
CONDICIONADO E MONITORAMENTO DO CONSUMO ENERGÉTICO DE SALAS
DE AULA**

Relatório técnico apresentado à
Universidade Federal de Lavras, como
parte das exigências do Curso de
Engenharia de controle e automação,
para a obtenção do título de Bacharel.

Prof. Dr. Wilian Soares Lacerda
Orientador

**LAVRAS - MG
2019**

MATHEUS APARECIDO MELO FREIRE

**PROJETO IOT PARA CONTROLE REMOTO DE ILUMINAÇÃO E AR
CONDICIONADO E MONITORAMENTO DO CONSUMO ENERGÉTICO DE SALAS
DE AULA:**

**IOT PROJECT FOR REMOTE CONTROL OF LIGHTING AND AIR
CONDITIONING AND ELECTRICAL CONSUMPTION MONITORING OF
CLASSROOMS**

Relatório técnico apresentado à
Universidade Federal de Lavras, como
parte das exigências do Curso de
Engenharia de controle e automação,
para a obtenção do título de Bacharel.

Prof. Dr. Wilian Soares Lacerda
Orientador

**LAVRAS - MG
2019**

RESUMO

A necessidade de se automatizar prédios e ambientes que possuem ar condicionado e iluminação vem crescendo cada vez mais. As vantagens desta automação englobam mais praticidade, mais conforto, possibilidade de monitoramento do ambiente, economia energética dentre várias outras. Este tipo de automação já existe, mas um novo conceito conhecido como IoT (*Internet of thing*) vem ganhando espaço na indústria da automação, levando estes projetos a um outro nível. Com a internet das coisas tudo está conectado nesta grande rede de comunicação que hoje muitos não vivem sem. Este trabalho apresenta o desenvolvimento de um sistema de controle para acionamento dos circuitos de iluminação e ar condicionado de uma sala de aula, bem como o monitoramento em tempo real do consumo elétrico do ambiente. O projeto foi realizado utilizando um Gateway inteligente da Siemens (Simatic IoT2000) para fazer a conexão a internet, um sensor de corrente (SCT-013) para fazer a medição de energia, um módulo de interface (Ircom) para o controle do ar condicionado e um relé para acionamento da iluminação. Este trabalho apresenta cada etapa do processo de criação, bem como os resultados finais do projeto montado.

Palavras chave: Internet das coisas. Automação. Acionamento remoto. Monitoramento energético.

ABSTRACT

The need to automate buildings and environments that have air conditioning and lighting is growing. The advantages of this automation include more convenience, possibility of environmental monitoring, economic savings and many others. This type of automation already exists, but a new concept known as IoT (Internet of thing) has been gaining ground in the automation industry, taking these projects to another level. With the internet of things, everything is connected in this great communication network that today we don't live without. This paper presents the development of a control system for the activation of lighting circuits and air conditioning in a classroom, as well as real time monitoring of the electrical consumption of the environment. The project was developed using a Siemens Smart Gateway (Simatic IoT2000) for the Internet connection, a current sensor (SCT-013) for the energy measurement, an interface module (Ircom) for the air conditioning control and a relay for lighting activation. This paper presents each step of the creation process as well as the final results of the installed project.

Keywords: Internet of things. Automation. Remote control. Energy monitoring.

LISTA DE FIGURAS

Figura 1 - Simatic IoT2040.....	11
Figura 2 - Simatic IoT2040 IO-Shield.....	11
Figura 3 - Fonte Wipos P1 24-2.5.....	12
Figura 4 - Retificador monofásico de onda completa.....	13
Figura 5 - Esquema em blocos de uma fonte chaveada.....	14
Figura 6 - Esquema com pinos de um relé.....	15
Figura 7 - Funcionamento de um relé.....	15
Figura 8 - Relé Finder.....	16
Figura 9 - Ircom e seus acessórios.....	17
Figura 10 - Diagrama de correntes do TC.....	19
Figura 11 - Transformador de corrente STC013-000.....	20
Figura 12 - Modem Huawei Vivo.....	20
Figura 13 - Exemplo de fluxo do Node-Red	22
Figura 14 - Esquema funcionamento cliente e <i>broker</i>	24
Figura 15 - Sistema operacional acessado pelo Putty.....	27
Figura 16 - Plataforma de desenvolvimento do Node-Red.....	29
Figura 17 - Fluxos do Node-Red para Iluminação.....	30
Figura 18 - Código em JavaScript do nó <i>function</i>	31
Figura 19 - Controlador Conflex Light AGST.....	33
Figura 20 - Fluxos para alteras registros do Ircom.....	34
Figura 21 - Tela geral do aplicativo.....	35
Figura 22 - Tela gestão de consumo do <i>Dashboard</i>	36
Figura 23 - Lâmpada de LED.....	37
Figura 24 - Botão normalmente aberto.....	37
Figura 25 - Projeto montado em bancada.....	38
Figura 26 - Borne simples.....	39
Figura 27 – Quadro de comandos.....	40
Figura 28 - Últimos eventos no aplicativo da Moka Mind.....	42
Figura 29 - <i>Insights</i> gerados.....	43
Figura 30 - Informações do consumo elétrico.....	43

LISTA DE TABELAS

Tabela 1 – Funções no protocolo Modbus	23
--	----

SUMÁRIO

1 INTRODUÇÃO.....	8
1.1 Contextualização e Motivação.....	8
1.1 Objetivo.....	8
1.2 Organização do texto.....	9
2 REFERENCIAL TEÓRICO.....	10
2.1 Hardwares.....	10
2.1.1 Gateway IoT Inteligente.....	10
2.1.2 Fonte 24V.....	12
2.1.3 Relê.....	14
2.1.4 Ircom.....	16
2.1.5 Sensor de corrente.....	18
2.1.6 Modem 4G.....	20
2.2 Softwares.....	21
2.2.1 Linux Yocto.....	21
2.2.2 Node-Red.....	21
2.3 Protocolos.....	22
2.3.1 Modbus.....	22
2.3.2 MQTT.....	23
3 DESENVOLVIMENTO DO PROJETO.....	25
3.1 Desenho elétrico.....	25
3.2 Lista de Sinais.....	26
3.3 Preparação do Sistema operacional.....	26
3.4 Broker.....	28
3.5 Programação do Node-Red.....	28
3.6 Configuração do Ircom.....	32
3.6.1 Aprendizado do Ircom.....	32
3.6.2 Endereçando os escravos da rede.....	33
3.6.3 Modificando configurações padrão.....	34
3.7 Criar Aplicativo e <i>Dashboard</i>	35
3.8 Montagem em bancada para teste.....	36
3.9 Montagem do quadro elétrico.....	39
3.10 Instalação.....	40
4 RESULTADOS E DISCUSSÕES.....	42
5 CONCLUSÃO.....	44
REFERENCIAS.....	46

1 INTRODUÇÃO

1.1 Contextualização e Motivação

Vivemos hoje uma nova fase de uma revolução tecnológica. A Indústria 4.0, que é a quarta revolução industrial, vem avançando e ganhando espaço no mercado da engenharia. Ela parte do princípio de novas tecnologias para automação e troca de dados. A previsão é que ela esteja cada vez mais presente no dia a dia da população a partir de uma nova tecnologia chamada Internet das Coisas, ou *IoT* (*internet of things*), que é um dos conceitos principais da indústria 4.0. Este fenômeno vem mudando a forma com que interagimos com o mundo, e como o mundo interage conosco.

A Internet das Coisas é nada mais nada menos do que a conexão de objetos do cotidiano a internet, formando assim, uma rede de comunicação possibilitando uma interação e troca de dados dos usuários com os objetos e dos objetos entre si. Com este simples conceito, pode-se criar aplicações que resultam em mais praticidade, conforto, economia, informação e uma infinidade de melhorias ainda inimagináveis. (VILARINO, 2018)

Embora seja uma tecnologia recente e que vem se desenvolvendo a cada dia, a Internet das coisas já é uma realidade. Hoje já existem mais objetos conectados do que pessoas no planeta. Mais de 15 bilhões de “coisas” já estão conectadas a internet. Segundo a consultoria Gartner, esse número deve subir para 26 bilhões até 2020, e por volta de 2025 poderá chegar a marca de 100 bilhões de objetos. (GARTNER, 2019)

A princípio, essa tecnologia surgiu a partir do conceito de automação residencial inteligente, o *smart home*, onde os objetos conectados eram luminárias, termostatos, sistemas de segurança doméstica e câmeras de vigilância. Hoje essa tecnologia vai muito além desta aplicação que deu a sua origem. Este relatório abrangerá um projeto em que o intuito se aproxima muito com a ideia inicial do IoT do *smart home*.

1.2 Objetivo

Este relatório tem como objetivo apresentar o desenvolvimento de um projeto para controle remoto de acionamento de iluminação e controle de ar condicionado de uma sala de aula. Este texto mostra como é feita a coleta de dados dos sensores e como os Hardwares do projeto atuam modificando o mundo real. Para isso, todos os passos de criação do projeto são detalhados. Por mais que não seja o foco deste relatório, vale ressaltar que o projeto também

oferece uma inteligência artificial para os usuários, utilizando *bots* com a geração de *insights*. Toda a coleta de dados juntamente com um reconhecimento de padrão transforma o produto em um sistema inteligente onde o usuário, na medida do tempo, não precise se preocupar e controlar o ambiente manualmente. Tudo isso visando conforto, praticidade e principalmente economia energética.

1.3 Organização do texto

Este trabalho é dividido em 5 capítulos. No capítulo 1 são apresentados a contextualização, a motivação e o objetivo do trabalho. No capítulo 2 é apresentado todo o referencial teórico para a sua execução. Nesta seção, é feito o detalhamento de cada Hardware e Software utilizados no projeto. No capítulo 3, é apresentado o passo a passo do desenvolvimento do projeto, desde o desenho do escopo até a instalação do mesmo, detalhando-se cada tarefa executada até atingir o produto final. O capítulo 4 mostra os resultados obtidos com o projeto executado. Finalmente, o capítulo 5 apresenta a conclusão e uma discussão das principais dificuldades e possíveis melhoramentos do projeto.

2 REFERENCIAL TEÓRICO

Este tópico abordará toda a literatura utilizada que serviu de base para o projeto. Ela será dividida em três subtópicos. Num primeiro momento será apresentado os Hardware utilizados, assim como suas propriedades. Em seguida inicia-se uma apresentação dos softwares utilizados, discorrendo tanto sobre o sistema operacional quanto a linguagem de programação utilizada. Outra importante parte dos projetos de automação e IoT se concentra nos protocolos de comunicação que estes utilizam, por isso, finalmente este tópico citará e descreverá os protocolos utilizados.

2.1 Hardwares

2.1.1 Gateway IoT Inteligente

Um gateway inteligente é um dispositivo que serve para receber, traduzir e enviar dados de e para todos os objetos IoT de um ambiente. Ele é o equipamento principal por trás de toda a aplicação.

O hardware utilizado neste projeto é o Simatic IoT2040 da Siemens, onde todos os dados dos sensores são enviados, processados e ele é o responsável por enviar comandos para os atuadores sendo o equipamento central de todo o processo, para onde todos os sensores e atuadores se voltam. Neste projeto, toda lógica do processo está no gateway determinando o comportamento dos equipamentos, tanto dizendo como eles devem agir quanto fazendo a requisição dos dados.

O Simatic IoT2040, mostrado na Figura 1, possui um processador Quark Intel com 1GB de RAM e diversas interfaces como portas Ethernet, portas de comunicação serial RS232 e RS485 e porta USB. Ele suporta o Linux Yocto que é colocado no Gateway por meio de um cartão de memória SD inserido no IoT2040. (ALVES, 2017)

Figura 1 – Simatic IoT2040.



Fonte: (SIEMENS, 2010).

O fabricante do Simatic IoT2040 oferece um módulo de expansão de Arduino que pode ser acoplado ao gateway para que ele possua entradas e saídas digitais e entradas analógicas. Este módulo oferece cinco entradas digitais de 24V, duas saídas digitais de 24V e duas entradas analógicas de 0-10V ou de 0-20mA.

O módulo de expansão possibilita que o gateway atue diretamente em diversos controladores e receba dados de sensores através das suas próprias portas de entrada e saída. A grande vantagem da utilização deste módulo é que o Simatic IoT2040 participa do projeto como o único equipamento conectado a internet. Tudo se comunica por meio das interfaces de comunicação serial e por meio das entradas e saídas, ou seja, pode-se criar um ambiente que será remotamente controlado utilizando sensores e atuadores comuns, que não precisam se conectar a internet, uma vez que estes se comunicam com o Simatic IoT2040 com fios, e este sim faz o papel de conectá-los a internet.

Figura 2 – Simatic IoT2040 IO-Shield.



Fonte: (SIEMENS, 2010).

2.1.2 Fonte de 24V

O Simatic IoT2040, precisa ser alimentado com uma fonte de energia entre 9 e 36 Volts. No caso deste projeto ele foi conectado com uma fonte que o alimenta com 24V. Além desta tensão de alimentação para o seu funcionamento, o módulo de expansão Arduino precisa que seus pinos das saídas digitais sejam conectados em 0 e 24V, e um pino de referência das entradas digitais deve ser conectado ao 0V da fonte.

Para este projeto, utilizou-se uma fonte monofásica Wipos P1 24-2.5 da Wieland mostrada na Figura 3. Esta fonte possui uma tensão entrada entre 115V e 230V VCA (tensão alternada) e uma tensão de saída de 24 VCC (tensão contínua). A corrente máxima da fonte é de 2,5A, fazendo com que ela tenha uma potência de 60 VA.

A escolha desta fonte foi devido a necessidade de utilizar um equipamento mais robusto, confiável e que não apresentasse falhas e defeitos com o tempo.

Figura 3 – Fonte Wipos P1 24-2.5.



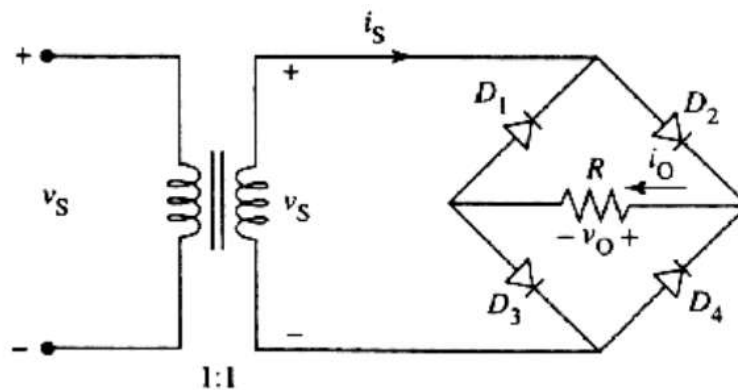
Fonte: (WIELAND, 2014).

A Wipos P1 24-2.5 se trata de uma fonte chaveada, que se difere das tradicionais fontes lineares. A grande vantagem de utilizar fontes chaveadas é que elas apresentam um tamanho e um peso menor e menos potência dissipada.

As fontes lineares funcionam através da utilização de um transformador. Este transformador, esquematizado na Figura 4, e que é o responsável por diminuir o valor da tensão que vai para os retificadores. Transformadores basicamente modificam os níveis de tensão mantendo a potência elétrica praticamente constante. Este transformador é um dos principais responsáveis pelo grande peso e tamanho destas fontes, e um dos principais motivos da substituição das fontes lineares por fontes chaveadas como é o caso da fonte Wipos P1 24-2.5 utilizada no projeto.

Um dos principais componentes de uma fonte é o retificador. A ponte retificadora monofásica de onda completa, utiliza quatro diodos. O diodo é um elemento básico da eletrônica, e sua principal característica é que ele é um semicondutor que faz com que a corrente flua em um sentido, mas impede que ela flua em outro. Este bloqueio da corrente em uma direção e a permissão da passagem em outra é o princípio básico utilizado pelos retificadores. Para isso, quatro diodos são conectados em ponte como mostra a Figura 4, onde D1, D2, D3 e D4 representam os quatro diodos.

Figura 4 – Retificador monofásico de onda completa.



Fonte: (URBANETZ JUNIOR; MAIA, 2014).

A tensão que alimenta este circuito é VCA e possui semiciclos positivos e negativos. Os diodos D2 e D3 conduzem no semiciclo positivo pois estão diretamente polarizados. Já no semiciclo negativo, os diodos D1 e D4 são os que ficam polarizados, e, portanto, conduzindo. Essa propriedade faz com que a tensão de entrada que possui semiciclos positivos e negativos, fique totalmente positiva, e usando um capacitor como filtro, o resultado é uma tensão de saída VCC do retificador.

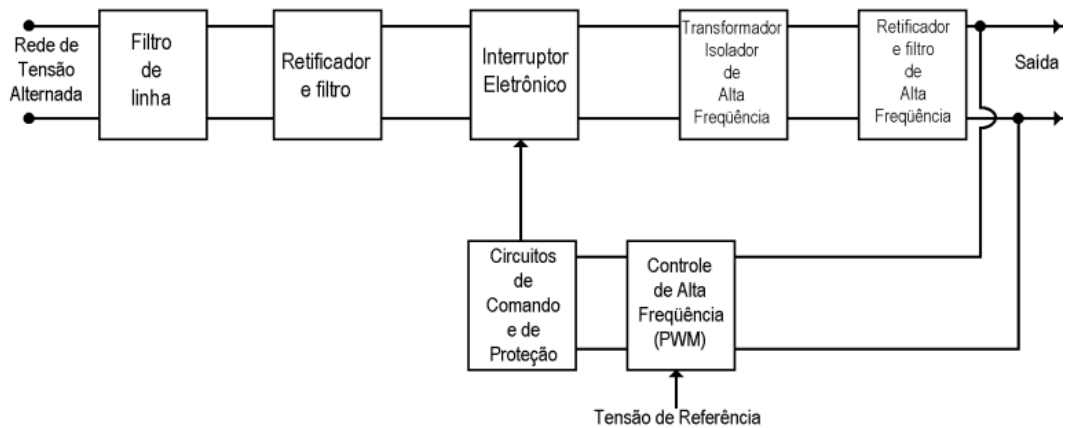
As fontes chaveadas, assim como as fontes lineares tem o objetivo de converter corrente alternada em corrente contínua, porém sem a necessidade de um transformador. Uma fonte chaveada exerce essa função trabalhando a frequência do circuito. Ela basicamente abre e fecha o circuito o que faz com que a tensão média seja diminuída para o valor desejado. Após o rebaixamento da tensão, utilizando filtros de capacitores e indutores, o sinal de saída fica contínuo.

O grande responsável pela abertura e fechamento do circuito, conhecido como chaveamento, é um circuito que controla a largura de um pulso chamado PWM. Ou seja, o PWM controla quanto tempo o circuito fica fornecendo energia a fim de se abaixar a tensão de

entrada para o valor desejado. Ou seja, o valor da tensão na saída esta diretamente ligada com a largura do pulso do PMW.

Um diagrama do funcionamento de uma fonte chaveada é mostrado na Figura 5.

Figura 5 – Esquema em blocos de uma fonte chaveada.



Fonte: (ATHOS ELECTRONICS, 2019).

2.1.3 Relê

O gateway inteligente é capaz enviar sinais para as suas saídas digitais e com isso controlar vários tipos de equipamentos. Como discutido na seção 2.1.1, as saídas digitais do módulo Arduino acoplado ao gateway fornecem 0V (nível lógico 0) ou 24V (nível lógico 1).

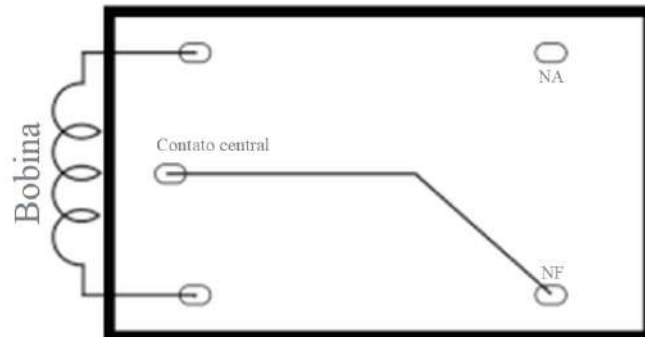
Quando é necessário ligar ou desligar algum equipamento que é alimentado com uma tensão diferente da fornecida pelas saídas digitais, faz-se necessário o uso de um relê.

Um exemplo de relê é o que funciona utilizando um sinal contínuo para controlar equipamentos que funcionam com corrente alternada, como é o caso de circuitos de iluminação, que geralmente funcionam com uma tensão de 127V ou 220V. Ou seja, um relê é um componente que age como um interruptor, sendo capaz de comutar uma chave a partir de um sinal. Existem vários modelos de relês com diferentes aspectos físicos, porém o princípio de funcionamento deles é sempre o mesmo (FABIO, 2019).

O elemento principal do relê que permite que ele exerça a sua função é um eletroímã. Quando uma corrente elétrica passa por uma bobina que envolve um núcleo, geralmente de ferro, cria-se um campo magnético que orienta os ímãs do ferro fazendo com que ele atraia outros materiais ferromagnéticos. Além deste circuito com a bobina, um relê apresenta um outro

circuito com duas chaves, uma normalmente aberta (NA) e uma normalmente fechada (NF) como mostra a Figura 6.

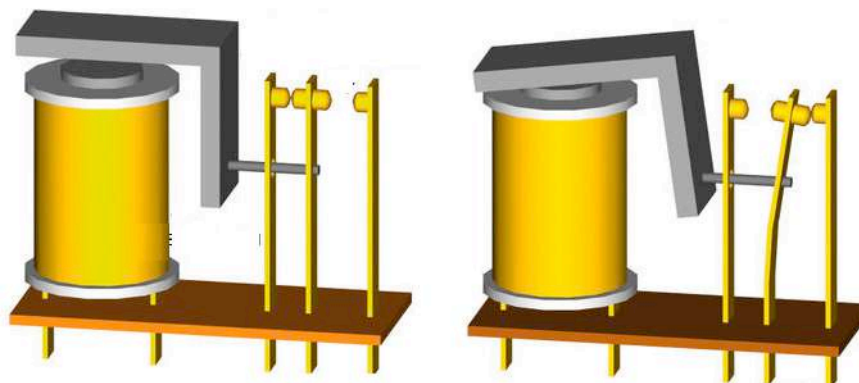
Figura 6– Esquema com pinos de um relé.



Fonte: (FABIO, 2019).

Existe um componente metálico que é o contato central do relé que sempre vai estar conectada com um dos pinos. Como mostra na Figura 7, quando a bobina está sem uma corrente passando por ela, há um contato com o pino normalmente aberto. Quando a bobina é energizada, o componente central é atraído como descrito anteriormente e ele deixa de fazer contato com o pino normalmente aberto para fazer contato com o pino normalmente fechado, fazendo assim a comutação. Ou seja, há a comutação de terminais quando a bobina está energizada. Portanto, como o circuito do acionamento do interruptor fica isolado, é possível fazer o acionamento de cargas que utilizam corrente alternada utilizando pequenos sinais de corrente contínua.

Figura 7 – Funcionamento de um relé.



Fonte: Adaptado de (FABIO, 2019).

Existem vários modelos e tipos de relés. O escolhido para este projeto foi o relé *Finder Master Plus* (Figura 8). A tensão de comutação é de 24V e é capaz de acionar circuitos de até 230VCA que tenham uma corrente de até 6A. O dispositivo possui um fusível de proteção oferecendo toda a segurança e confiabilidade demandada pelo projeto.

Figura 8 – Relé Finder.



Fonte: (Do autor, 2019).

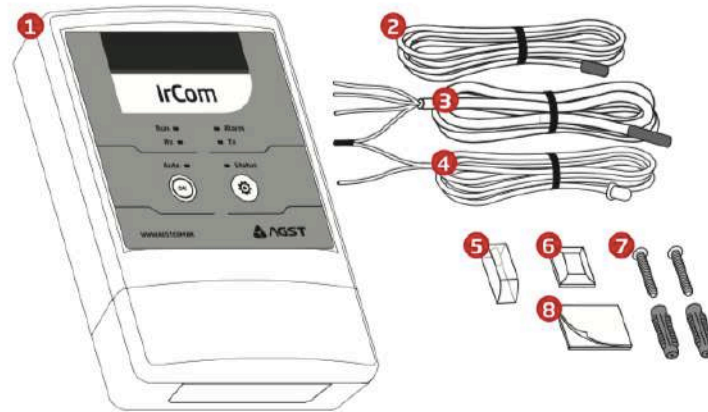
2.1.4 Ircom

O Ircom é um aparelho responsável por ligar, desligar e mudar o *setpoint* de temperatura de ares condicionados. Funciona como um módulo de interface que trabalha junto com um controlador AGST que faz o papel de mestre, e ele faz o papel de escravo na rede. No caso específico deste projeto, é utilizado apenas o Ircom, sem a controladora AGST. Assim, o Simatic IoT2040 faz o papel de mestre e o Ircom é conectado diretamente a ele através de cabos de comunicação serial.

O Ircom funciona como um controle remoto sem fio. Ele pode ser alimentado na rede elétrica com uma tensão alternada de 110V a 220V e é geralmente conectado em paralelo com os ares condicionados que fornecem esta tensão por meio de bornes no interior dos *Splits* (nome

dado ao modelo de ar condicionado mais conhecido no mercado, que possui condensadora e evaporadora geralmente controlado por controles remotos infravermelhos).

Figura 9 – Ircom e seus acessórios.



- | | |
|---------------------------------------|--|
| 1. <i>Módulo de Interface IrCom</i> | 6. <i>Suportes para fixação do Emissor IR e sondas</i> |
| 2. <i>Sonda de retorno IrCom</i> | 7. <i>Conjunto de parafusos e buchas para fixação</i> |
| 3. <i>Sonda de insuflamento IrCom</i> | 8. <i>Fita dupla face para fixação</i> |
| 4. <i>Emissor IR</i> | |
| 5. <i>Guia de luz</i> | |

Fonte: (AGST, 2017).

A Figura 9 apresenta um desenho do Ircom e os seus componentes. Além dos equipamentos para a sua fixação e instalação, é válido fazer uma breve análise dos elementos principais que permitem o seu funcionamento: o emissor infravermelho o sensor de temperatura de retorno e o sensor de temperatura de insuflamento.

A principal função do Ircom é receber um comando via modbus e fazer a tradução desta informação, enviando-a por meio da comunicação sem fio infravermelha. Ou seja, como um controle remoto, ele envia um sinal IR que é reconhecido pelos receptores infravermelhos dos ares-condicionados. Para isso o Ircom possui um cabo com um LED emissor infravermelho de 5mm. Esse sinal é então enviado através de uma luz que é uma radiação eletromagnética que fica numa faixa que não é perceptível ao olho humano. (SILVA JÚNIOR, 2015).

Embora a principal função do Ircom seja fazer o envio de comandos para os *Splits*, eles também possuem sensores de temperatura. Como descrito anteriormente, o Ircom possui dois sensores de temperatura, um que coleta os dados da temperatura de retorno, que é a temperatura que entra no ar condicionado (temperatura ambiente da sala) e um que coleta a temperatura do ar que sai do ar condicionado conhecida como temperatura de insuflamento. Coletar os dados

da temperatura de insuflamento é importante quando se deseja fazer o monitoramento da eficiência do ar condicionado. Ambos os sensores são conectados ao módulo Icom, e os dados de temperatura coletados são enviados via cabos serial com a comunicação modbus RTU para o Simatic IoT2040. O módulo possui um micro controlador interno, desta maneira, os dados de temperatura, que são valores analógicos, podem ser convertidos, fazendo com que seja possível requisitar e utilizar os dados de temperatura já em graus célsius por exemplo.

2.1.5 Sensor de corrente

Como discutido na seção 2.1.1, o Simatic IoT200 possui duas entradas analógicas, que podem ler de 0-10V ou de 0-20mA. Estas entradas são muito importantes para se fazer a leitura de corrente utilizando sensores para que se possa monitorar o consumo de energia de aparelhos elétricos. Esta medição pode ser feita utilizando sensores de corrente.

Neste projeto, uma corrente alternada é consumida devido ao uso do ar condicionado. Essa corrente é medida utilizando um sensor de corrente alternada não invasivo SCT-013. A partir desta medição é possível estipular a potência instantânea que o aparelho está usando.

A utilização deste tipo de sensor é muito comum devido à sua principal vantagem que é o fato de ele ser não invasivo, ou seja, ele consegue medir a corrente alternada que passa por um fio sem a necessidade de um contato físico. Desta forma não é necessário que o circuito seja aberto e, portanto, nenhuma alteração na instalação elétrica no equipamento precisa ser feita. O que precisasse fazer é apenas utilizar o sensor para envolver um dos fios que alimenta o equipamento. Este funcionamento que faz com que o uso deste sensor seja tão prático se deve as propriedades da corrente alternada, lei de Ampere e a lei de Faraday.

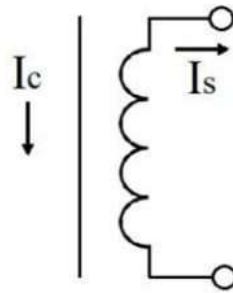
A corrente alternada é uma senoide, ou seja, ela varia de um valor negativo a zero e de zero a um valor positivo ao longo do tempo. Essa variação faz com que a corrente seja uma forma de onda com uma frequência de 60Hz, ou seja, variando do negativo ao positivo 60 vezes por segundo.

A lei de Ampere nos prova que quando uma corrente passa por um condutor, como um fio elétrico por exemplo, ela cria um campo magnético em torno deste condutor que é proporcional ao seu valor. Como esse valor do campo magnético é proporcional a corrente, quando uma corrente alternada, que varia ao longo do tempo, passa por um condutor, o campo criado por ela também varia. (TIPLER; MOSCA 2009)

A Lei de Faraday, nos mostra que quando há uma variação de campo magnético atravessando uma espira, esta cria uma corrente elétrica que é proporcional a intensidade do campo magnético que a atravessa. (TIPLER; MOSCA 2009)

Com estes três princípios conseguimos explicar o funcionamento de um transformador de corrente, que é o que se trata o sensor utilizado: SCT-013. Este sensor, mais conhecido como TC (transformador de corrente) funciona basicamente através de um conjunto de espiras que geram uma corrente elétrica devido a presença de uma variação de um campo magnético. A Figura 10 exemplifica este funcionamento. O TC cria uma corrente I_s induzida devido a presença de uma corrente I_c que passa pelo fio. A corrente I_c é proporcional a corrente I_s , e sabendo-se a proporcionalidade entre elas é possível então medir a corrente de um equipamento utilizando o TC.

Figura 10 – Diagrama de correntes do TC.



Fonte: (DEMETRAS, 2019).

O SCT-013 possui vários modelos que variam principalmente em relação a corrente de entrada (corrente que deseja medir), e o seu sinal de saída, que pode ser em amperes ou em volts. No caso deste projeto, o sensor utilizado foi o SCT-013-000 mostrado na Figura 11. Este modelo faz uma leitura de uma corrente alternada entre 0 e 100 amperes, e o seu sinal de saída é uma corrente que varia de 0 a 50mA.

Figura 11 – Transformador de corrente STC013-000.



Fonte: (YHDC, 2019).

2.1.6 Modem 4G

Para se conectar a internet, faz-se necessário o uso de um modem para uma conexão com um sinal 4G. Para isso, um modem Huawei E173 da Vivo como o da Figura 12 é usado. Com ele, é possível se obter um sinal 4G através de um chip como o utilizado em celulares. Ele consegue se conectar com aparelhos que possuem portas USB, como é o caso do Simatic IoT2040.

Figura 12 – Modem Huawei Vivo.



Fonte: (Do autor, 2019).

2.2 Softwares

2.2.1 Linux Yocto

O sistema operacional utilizado pelo Simatic IoT2040 é o Linux Yocto que é um sistema operacional que vem sendo utilizado cada vez mais em equipamentos. Ele é um sistema de *built* que é *open source*, com ferramentas para criação e distribuição Linux. (SUEIRO, 2013).

O conceito do Linux embarcado não se difere do Linux utilizado em computadores. Sua diferença é que o Linux embarcado é customizado e adaptado para que ele possa ser “acoplado” ao Hardware específico. Para isso é necessário agregar a ele artefatos de software como o *Bootloader*, o Linux Kernel, Bibliotecas, Serviços e Aplicações que serão executados no Hardware alvo. (SUEIRO, 2013).

2.2.2 Node-Red

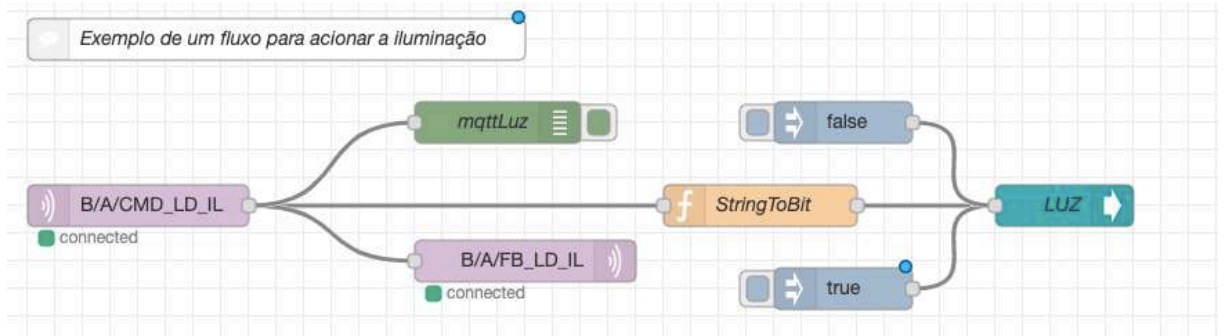
Em 2013 a IBM criou uma plataforma de programação de código aberto feita especialmente para desenvolvimento de projetos IoT chamada Node-Red.

O Node-Red é uma ferramenta visual que possui vários APIs e é uma plataforma de desenvolvimento que facilita a visualização geral da lógica por meio dos fluxos que são criados. Estes fluxos são formados por blocos, como mostra a Figura 13, conhecidos como nós. Os nós podem ser funções de entrada, processamento ou saída. A interligação destes nós formam os fluxos, que por sua vez, determinam a lógica do funcionamento da aplicação.

O Node-Red funciona em qualquer sistema operacional, e uma vez instalado, é possível fazer todo o desenvolvimento utilizando o próprio navegador como editor. Ele possui aproximadamente 40 nós com as mais diversas funções e muitos outros podem ser adicionados a partir de bibliotecas existentes.

Os fluxos fazem trocas de mensagens que são objetos *JavaScript* conhecidos como *msg*. Em cada nó, as mensagens são geradas, consumidas e processadas. Além disso, pode-se anexar outras propriedades nestas mensagens que serão usadas nos próximos fluxos. Todos estes nós são conectados por fios, que ligam as extremidades de saídas dos nós nas entradas de outros.

Figura 13 – Exemplo de fluxo do Node-Red.



Fonte: (Do autor, 2019).

2.3 Protocolos

2.3.1 Modbus

Hoje em dia, um dos protocolos de comunicação mais utilizados é o Modbus. Ele foi desenvolvido na década de 1970 pela Modcon, e devido a sua simplicidade e facilidade, é utilizado até hoje. Este protocolo é baseado na comunicação de um mestre com os seus escravos dependendo do padrão utilizado. Estes padrões variam entre RS-232, RS-485 e o Modbus TCP (FREITAS, 2014).

O padrão RS-232 serve para redes que possuem apenas um mestre e um escravo, ou seja, é uma comunicação de ponto a ponto, que pode ter uma distância máxima entre eles de 30 metros e se comunicam com uma velocidade por volta de 114Kbps. Já padrão RS-485 pode ter um número máximo de 32 dispositivos, uma velocidade de comunicação que pode chegar a 12Mbps, e a distancia máxima permitida na rede é de aproximadamente 1200 metros. Ele é o padrão mais utilizado hoje pelo protocolo Modbus. Finalmente, o Padrão Modbus TCP, também conhecido como Ethernet TCP/IP, tem uma conexão que pode chegar a 100Mgps ou até mesmo a 10Gbps COM uma distância máxima de 200 metros.

Para se fazer a comunicação, o protocolo Modbus envia uma série de bits (um ou zero), permitindo que o mestre envie comandando ou solicite que os escravos enviem dados a ele. Os escravos recebem estas informações e respondem transmitindo os dados solicitados ou modificando o seu estado como ordenado pelo mestre da rede.

Para se fazer esta comunicação, existem dois diferentes tipos de transmissão, o ASCII e o RTU. São eles quem determinarão como os bytes das mensagens serão transmitidos, dizendo como eles se compactarão e descompactarão.

O protocolo Modbus possui 256 endereços. O mestre não possui um endereço. O endereço 0 é o endereço que envia mensagem para todos os escravos quando o mestre se comunica com eles, conhecido como endereço de *Broadcast*. Os endereços de 1 a 247 são destinados para os escravos e o restante são endereços reservados. Desta maneira o mestre pode comunicar-se individualmente com cada escravo através do endereço a ele alocado, e para determinar o tipo de função ou serviço que o mestre deseja solicitar, basta utilizar o padrão de códigos para as funções desejadas como a Tabela 1.

Tabela 1 – Funções no protocolo Modbus.

Código	Descrição
1	Leitura de bloco de bits do tipo coil (saída discreta).
2	Leitura de bloco de bits do tipo entradas discretas.
3	Leitura de bloco de registradores do tipo holding.
4	Leitura de bloco de registradores do tipo input.
5	Escrita em um único bit do tipo coil (saída discreta).
6	Escrita em um único registrador do tipo holding.
7	Ler o conteúdo de 8 estados de exceção.
8	Prover uma série de testes para verificação da comunicação e erros internos.
11	Obter o contador de eventos.
12	Obter um relatório de eventos.
15	Escrita em bloco de bits do tipo coil(saída discreta).
16	Escrita em bloco de registradores do tipo holding.
17	Ler algumas informações do dispositivo.
20	Ler informações de um arquivo.
21	Escrever informações em um arquivo.
22	Modificar o conteúdo de registradores de espera através de operações lógicas.
23	Combina ler e escrever em registradores numa única transação.
24	Ler o conteúdo da fila FIFO de registradores.
43	Identificação do modelo do dispositivo.

Fonte: (FREITAS, 2014).

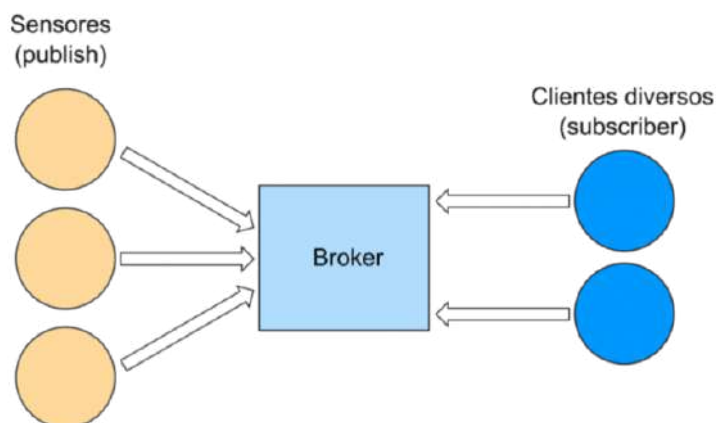
2.3.2 MQTT

O MQTT é um protocolo de troca de mensagens criado pela IBM no final da década de 1990 que é usado para aplicações IoT. Sua utilização nos projetos IoT se deve ao fato de que os dispositivos precisam se comunicar entre si e também se comunicar com os serviços de

backend. Embora seja um protocolo de simples, o MQTT é um protocolo seguro, de qualidade e de fácil implantação (YUAM, 2017).

O MQTT se baseia na forma de troca de mensagens *public/subscriber* (publicador/subscritor). Isso é possível utilizando um *broker* que é o elemento central intermediário da aplicação. Quando deseja-se receber uma informação, um elemento da rede faz uma requisição para o *broker* (subscrive). Os elementos da rede que quiserem publicar algo, conseguem fazê-la também a partir do *broker*, enviando-lhe as informações (publica). Ou seja, o broker é um servidor para onde todas as mensagens dos clientes são enviadas e ele roteia todas estas informações para os clientes. O esquema mostrado pela Figura 14 mostra como este protocolo funciona (BARROS, 2015).

Figura 14 – Esquema funcionamento cliente e *broker*.



Fonte: (BARROS, 2015).

As mensagens enviadas pelo protocolo MQTT precisam ser identificadas para que se consiga saber qual sensor está publicando uma determinada mensagem ou para que se possa publicar uma mensagem para determinado dispositivo. Esta identificação é possível devido ao uso de tópicos. Os tópicos são parecidos com o conceito URL, e possui níveis separados pelo caractere barra: (“/”). Desta maneira é possível identificar um sensor pelo seu tópico como por exemplo: “casa1/sala/sensor/temperatura1”. Neste tópico sabemos que a mensagem que está chegando é do sensor de temperatura 1, instalado na sala da casa 1. O padrão de classificação fica por conta do desenvolvedor de modo a facilitar ao máximo a identificação de cada dispositivo, sempre usando “/” para separar as categorias. (BARROS, 2015).

No MQTT não existem restrições quanto ao formato que a mensagem será enviada, ou seja, o formato dependerá da aplicação. Desta forma é possível codificar a mensagem em json, em um valor binário de vários bits, dentre outras formas de codificação de mensagem.

3 DESENVOLVIMENTO DO PROJETO

Esta seção detalhará cada etapa do processo de execução deste projeto. Para facilitar a organização da criação do projeto, o mesmo foi dividido em diferentes etapas: criação do desenho elétrico, lista de sinais, preparação do sistema operacional, criação do broker, programação do Node-Red, configuração do Ircom, criação do aplicativo e *dashboard*, teste em bancada, montagem do quadro e instalação

3.1 Desenho elétrico

Uma vez que o projeto foi definido e está pronto para começar a ser executado, um desenho dos diagramas elétricos e de comunicação de todos os equipamentos que farão parte do projeto foi feito. Utilizando o software EPLAN, o desenho foi criado e ele esquematiza a posição de todos os Hardwares dentro do painel elétrico, as ligações elétricas que serão feitas, as portas de comunicação que serão utilizadas, bem como quais equipamentos serão conectados a elas. Além disso ele também determina em quais entradas e saídas cada sensor, botão e atuador serão conectados ao Simatic IoT 2000.

O desenho começa mostrando a alimentação do painel. A tensão alternada proveniente da rede elétrica do local alimenta a fonte. A tensão que sai da fonte alimenta o Simatic Iot2040, alimenta suas portas de saída digital e fornece a referência negativa para as portas de entrada digital. O desenho também mostra que o Ircom deve ser alimentado diretamente pela tensão alternada do local.

Depois de o desenho do projeto determinar como as ligações de alimentação devem ser, o mesmo esquematiza as portas lógicas do Simatic IoT2040. Primeiramente o desenho mostra em quais portas analógicas o sensor de corrente é conectado, em quais saídas digitais os reles são conectados, e em quais entradas digitais o botão e o sinal de retorno de estado da iluminação devem ser conectados.

Além das portas de entrada e saída, o Simatic se comunica com o Ircom por meio da sua porta de comunicação serial via modbus. Os pinos desta porta que são usados e a esquematização desta comunicação também são informadas pelo desenho do projeto.

No Ircom são conectados dois sensores de temperatura, um emissor infravermelho, os cabos de comunicação modbus e um resistor para fechar o varal da rede. O posicionamento de conexão de todos estes componentes na entrada do Ircom também é detalhado pelo desenho.

Além de toda a esquematização do projeto, este desenho elétrico também apresenta uma lista com todos os materiais que são necessários para a sua construção.

3.2 Lista de sinais

A lista de sinais, é um documento que é basicamente uma tradução do desenho elétrico para uma planilha no Excel contendo todos os dispositivos que fazem parte do projeto. Este documento tem duas utilidades. A primeira utilidade é facilitar o desenvolvimento do projeto em geral e principalmente facilitar a programação dos fluxos do Node-Red, pois ela contém todas as informações necessárias de cada dispositivo. A segunda utilidade da lista de sinais é que ela apresenta colunas com as informações necessárias para a criação do aplicativo. Os responsáveis pela criação do aplicativo, desenvolveram um software que importa esta lista de sinais, e baseado nas informações nela contida, um aplicativo customizado contendo os ambientes, o número de ares condicionados e de circuitos de iluminação é gerado.

A lista de sinais conta com 86 colunas, cada uma com a sua função específica. Estas colunas apresentam informações de cada dispositivo como, por exemplo, em qual porta o dispositivo está ligado, o tópico MQTT, o título do dispositivo que aparecerá no aplicativo, o ambiente em que ele está, dentre outras especificações.

Além das colunas, cada dispositivo tem as suas linhas que os representa. Um circuito de iluminação por exemplo apresenta três linhas nesta tabela, uma linha para o comando de ligar ou desligar, uma linha para o feedback do comando, e uma linha para retornar o estado da iluminação.

3.3 Sistema operacional

Depois destas etapas de pré-projeto, inicializa-se a execução do mesmo. Para isso, o primeiro passo é a preparação de um *SD card* contendo o sistema operacional que será executado pelo Simatic IoT2040. O software utilizado para projetos como este é o Yocto 2.1. O sistema operacional, contendo todos os *scripts* de inicialização e softwares como o Node-Red instalados é gravado no *SD card* precisando apenas fazer pequenas mudanças nele.

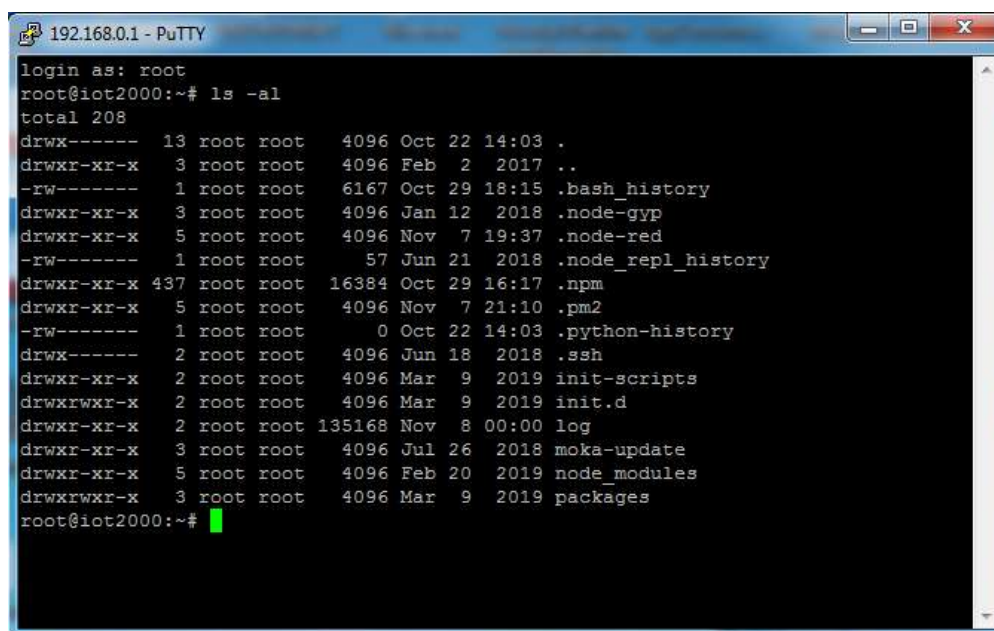
O primeiro passo para gravar o cartão é descompactar o arquivo do sistema operacional, e gravá-lo em no *SD card* de 32GB. O sistema operacional possui 31GB, e a gravação dele em um cartão de memória leva entre 30 e 40 minutos.

Depois de gravar o sistema operacional no cartão de memória, deve-se de inseri-lo no Simatic IoT2040 e liga-lo. Uma vez ligado, é necessário conectar uma ponta de um cabo de rede na primeira porta Ethernet do Gateway inteligente (X1 P1 LAN) e a outra ponta em um computador. A primeira porta Ethernet do IoT2040 é a sua porta de rede com IP fixo. A forma descrita de como é conectado um cabo de rede nele e em um computador, se classifica como uma conexão de ponta a ponta e sabendo-se o IP fixo é possível acessar o Gateway através de um computador utilizando o *software* Putty nele instalado.

Para utilizar o Putty, basta fazer o *download* deste *software* e indicar o IP que desejasse acessar. Uma vez conectado, uma tela como o Figura 15 aparece e ela mostra exatamente o sistema operacional rodando no Hardware da Siemens. A partir daí é possível dar comandos, executar *scripts* dentre outras funções.

As únicas modificações que devem ser feitas no Yocto 2.1, é trocar os tópicos de broker local que ficam na pasta moka-update. Isto é necessário pois cada projeto tem o seu prefixo definido de tópico MQTT. São estes tópicos que definirão qual projeto é esse, ou seja, funcionam como os identificadores dos comandos que são destinados a este projeto. Estes tópicos, assim como todos os outros, estão presentes na lista de sinais.

Figura 15 – Sistema operacional acessado pelo Putty



```
192.168.0.1 - PuTTY
login as: root
root@iot2000:~# ls -al
total 208
drwx----- 13 root root  4096 Oct 22 14:03 .
drwxr-xr-x   3 root root  4096 Feb  2  2017 ..
-rw-----   1 root root  6167 Oct 29 18:15 .bash_history
drwxr-xr-x   3 root root  4096 Jan 12  2018 .node-gyp
drwxr-xr-x   5 root root  4096 Nov  7  19:37 .node-red
-rw-----   1 root root    57 Jun 21  2018 .node_repl_history
drwxr-xr-x  437 root root 16384 Oct 29 16:17 .npm
drwxr-xr-x   5 root root  4096 Nov  7  21:10 .pm2
-rw-----   1 root root    0 Oct 22 14:03 .python-history
drwx-----   2 root root  4096 Jun 18  2018 .ssh
drwxr-xr-x   2 root root  4096 Mar  9  2019 init-scripts
drwxrwxr-x   2 root root  4096 Mar  9  2019 init.d
drwxr-xr-x   2 root root 135168 Nov  8 00:00 log
drwxr-xr-x   3 root root  4096 Jul 26  2018 moka-update
drwxr-xr-x   5 root root  4096 Feb 20  2019 node_modules
drwxrwxr-x   3 root root  4096 Mar  9  2019 packages
root@iot2000:~#
```

Fonte: (Do autor, 2019).

3.4 Broker

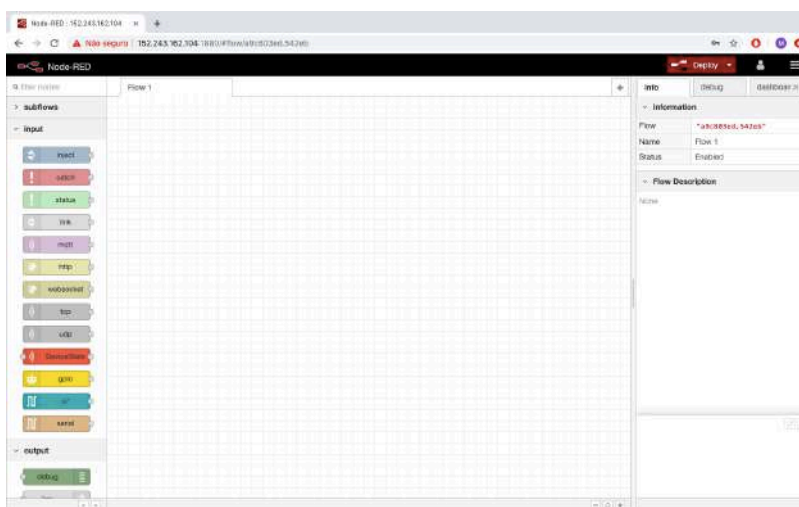
Antes de começar a programação embarcada no Hardware do projeto, é necessário criar um servidor, usuário e senha para o broker, ou seja, criar o mqtt-broker. Para este projeto, foi utilizado o broker *mosquitto*. Todos os projetos como este utilizam o broker *mosquitto*, e em cada projeto o broker é criado com um nome diferente para o servidor, um nome diferente para o usuário e uma senha diferente. É nas configurações dos nós responsáveis pela comunicação MQTT que o broker específico é colocado, informando-se o endereço do servidor, a porta, o usuário e a senha criados.

3.5 Programação do Node-Red

Instalado no sistema operacional, o Node-Red inicializa e começa a funcionar automaticamente sempre que o Simatic IOT2040 é ligado e é possível acessá-lo utilizando um navegador. Para isso é preciso descobrir qual o IP o Simatic IOT2040 está utilizando. Existem três tipos de IPs que podemos acessar. O primeiro é conectando o Gateway com um cabo de rede de ponta a ponta utilizando a primeira porta Ethernet (X1 P1 LAN) do Simatic e uma porta Ethernet de um computador. Esta porta utiliza o IP fixo do Simatic, que neste caso foi escolhido o IP “192.168.0.1”. Outra maneira é utilizar a segunda porta Ethernet (X2 P1 LAN), responsável por conectá-lo a internet. Esta porta atribui um IP DHCP para o Gateway, e utilizando um software que encontra os IPs que estão sendo utilizados na rede, como o “nmap” por exemplo, é possível acessar o Simatic através do IP encontrado. A terceira e última forma de acessar o hardware da Siemens é utilizando o IP de um chip 4G a ele atribuído, caso ele esteja conectado a um modem.

Uma vez encontrado o IP que representa o equipamento, é possível acessar a plataforma de desenvolvimento do Node-Red pelo navegador utilizando a porta 1880. Se o IP fixo for utilizado, o endereço que deve ser acessado pelo navegador é “192.168.0.1:1880” por exemplo. Uma página em branco pronta para a programação como a da Figura 16 é mostrada.

Figura 16 – Plataforma de desenvolvimento do Node-Red.



Fonte: (Do autor, 2019).

A lógica programada no Node-Red é que determina como o Simatic IOT2040 se comporta, ou seja, é através dele que é possível mudar o nível lógico de uma saída digital, coletar dados das entradas analógicas e digitais. Por meio dela também é possível se comunicar, coletar informações e enviar comandos para dispositivos modbus, como o caso do Ircom. E é no Node-Red que é possível enviar e receber informações da nuvem.

A maioria dos nós utilizados para compor a lógica já vem instalados no Node-Red e muitos outros nós podem ser instalados, uma vez que existem várias bibliotecas com os mais variados tipos de nós, cada um fazendo uma função específica.

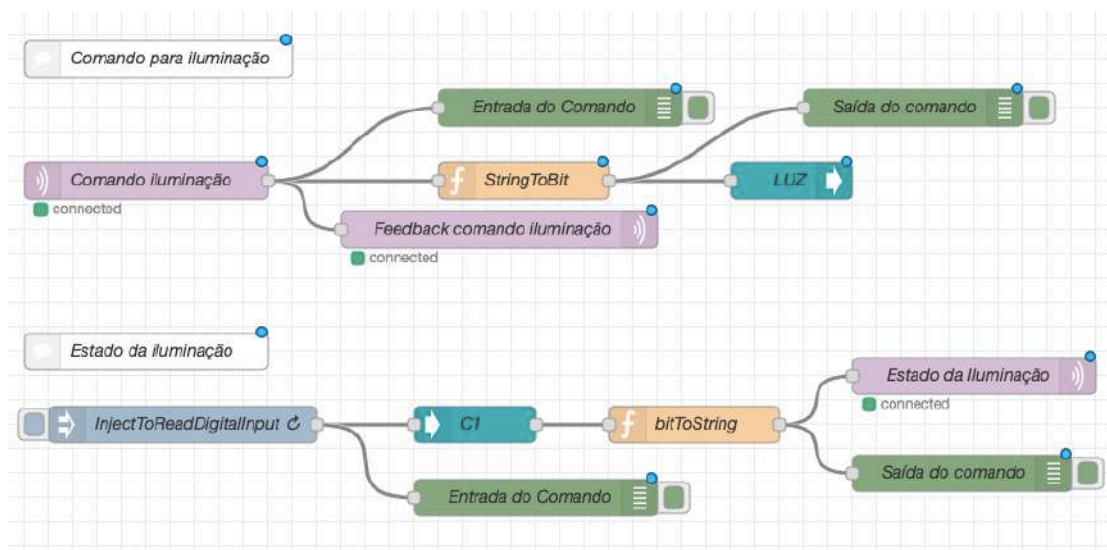
Embora exista tipos variados de nós, para algumas funções específicas do projeto há a necessidade de programar alguns deles. Para isto há a possibilidade de usar um bloco de programação livre chamado “*function*”. Nestes blocos, utilizando *JavaScript*, o programador é livre para programa-lo para que ele execute a tarefa por ele desejada.

Para fazer a lógica do processo deste projeto, existem basicamente dois tipos de fluxo: os que vão enviar comandos para atuadores e os que vão coletar dados. Os que controlam os equipamentos, começam com um nó de comunicação MQTT recebendo comandos da nuvem. Essas informações recebidas passam por alguns nós que vão processar os dados recebidos, modifica-los, e enviá-los para os nós que acionam os equipamentos que transformam o mundo real, como uma saída digital conecta uma lâmpada por exemplo. Já os fluxos que vão coletar dados, fazem o papel inverso, eles começam com um nó responsável por ler informações dos equipamentos que estão no mundo real, como sensores de corrente conectados as entradas analógicas por exemplo. Estas informações lidas passam por fluxo que interpretam essas

informações, as modifica, e as envia para os nós de comunicação MQTT, para enviá-las a nuvem.

Para exemplificar os fluxos que compõem a lógica, pode-se utilizar a simplificação de dois fluxos que compõem o funcionamento da iluminação da sala de aula como mostra a Figura 17. Para acionar a iluminação remotamente são necessários os dois tipos de fluxo, um para receber dados da nuvem e comandar a iluminação e outro para ler uma entrada digital que recebe um sinal que confirma que a iluminação foi ligada. A Figura 17 mostra estes dois fluxos.

Figura 17 – Fluxos do Node-Red para Iluminação.



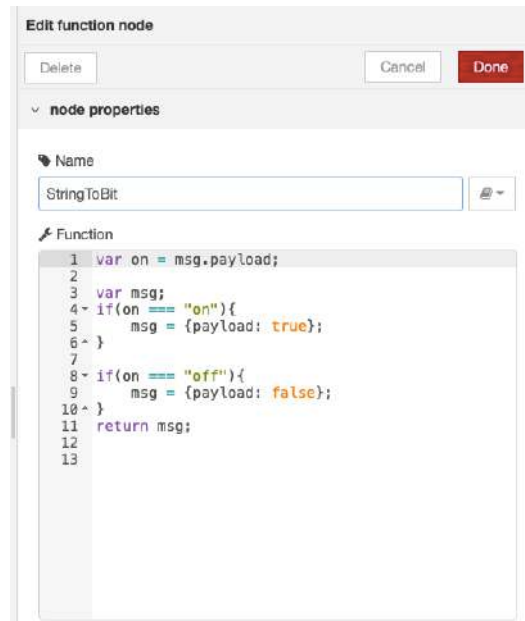
Fonte: (Do autor, 2019).

Os nós de cor roxa são os nós MQTT, neles chegam as informações que estão no broker. Quando um usuário do aplicativo envia um comando para ligar a iluminação, ele manda uma mensagem “on” ou “off” para o broker e o broker envia essa mensagem para o tópico específico de acionamento de iluminação, representado pelo nó nomeado “Comando iluminação”. O nó “Feedback comando iluminação”, também MQTT, recebe o comando enviado pelo nó “Comando iluminação” e o envia para o broker. Este feedback serve para que seja enviado um sinal para o broker de que o comando para controle do circuito de iluminação foi recebido.

A string “on” ou “off” vinda pelo nó MQTT entra no nó de função, e este contém um código programado, como o mostrado na Figura 18, que basicamente recebe o comando vindo pelo aplicativo e traduz “on” para o sinal booleano “True” e “off” para o sinal booleano “False”. Este sinal lógico é então enviado para o nó “LUZ”, que é um nó de uma biblioteca específica para o *Hardware* do Simatic IOT2040. Esta biblioteca basicamente envia sinais

lógicos para as saídas digitais, e coleta os sinais que estão entrando nas entradas digitais e analógicas, bastando apenas selecionar a entrada/saída que o programador quer comandar/ler.

Figura 18 – Código em JavaScript do nó *function*.



```

1 var on = msg.payload;
2
3 var msg;
4 if(on === "on"){
5   msg = {payload: true};
6 }
7
8 if(on === "off"){
9   msg = {payload: false};
10 }
11 return msg;
12
13

```

Fonte: (Do autor, 2019).

Os nós verdes, como o “Entrada do Comando” por exemplo, são nós de debug. Eles exibem os valores que chegam neles no canto direito da plataforma de desenvolvimento. Servem para facilitar a programação do fluxo, permitindo que o programador enxergue as informações que estão trafegando nos fluxos.

Finalmente, o nó azul claro, como é o caso do “InjectToReadDigitalInput”, se trata de um nó que gera e injeta sinais nos fluxos. No caso do exemplo de acionamento de iluminação, ele está configurado para injetar um sinal “True” a cada um segundo. Este sinal entra no nó “C1”, fazendo a leitura do que está na entrada digital D0 do Simatic IOT2040. Esta entrada digital recebe um sinal de 24V quando o circuito de iluminação está ligado. Após fazer a leitura, o sinal é traduzido no bloco “bitToString” e envia para o broker o estado atual do circuito de iluminação.

Embora o caso de acionamento de iluminação resuma bem a ideia de como os fluxos do Node-Red funcionam, os outros fluxos seguem os mesmos princípios, porém envolvem mais nós e programas em *JavaScript* mais complexos nos blocos “*function*”.

3.6 Configuração do Ircom

Para controlar os ares condicionados, o projeto utiliza o equipamento Ircom fabricado pela AGST. Como discutido na seção 2.1.4, o Ircom controla o ar condicionado emitindo um sinal infravermelho para o receptor infravermelho do Split. Antes de ser conectado com o IOT2040 pela porta serial, existem uma série de configurações que precisam ser feitas em cada aparelho.

3.6.1 Aprendizado do Ircom

Primeiramente é preciso fazer com que o Ircom se comunique com o *Split* do ambiente que o projeto será instalado. O Ircom basicamente recebe comandos através do protocolo modbus e envia sinais para o ar condicionado por meio de um emissor infravermelho que vem anexado a ele. Cada marca e modelo de *Split* de ar condicionado, trabalha com uma frequência diferente na comunicação sem fio infravermelha. Desta forma, para se comunicar com os aparelhos de ar condicionado, é necessário fazer um pequeno passo a passo de aprendizado do Ircom, onde, utilizando o controle remoto do ar condicionado e o Ircom, vários comandos são ensinados e salvados nas configurações do módulo para que ele possa se comunicar e enviar os sinais corretos para a marca e modelo específica do *Split* utilizado.

O aprendizado mencionado funciona da seguinte maneira: o controle remoto do ar condicionado envia o sinal específico que o *Split* precisa receber, por esse motivo, quando o Ircom está no modo aprendizado, basta apontar o comando do controle para o receptor do Ircom que ele replica o sinal para o ar condicionado. Se o ar condicionado executar a ação que deveria fazer se o controle remoto estivesse apontado diretamente para ele, basta apertar um botão de confirmação do Ircom que o comando será aprendido e guardado na memória do aparelho. Quando todos os comandos forem ensinados o Ircom está pronto para o próximo passo.

Como os Ircoms são configurados e testados em bancada, antes de irem para a instalação eles são primeiramente configurados para responder os Splits instalados no local do desenvolvimento. Quando são levados para obra no local de instalação eles precisam passar por este processo novamente.

3.6.2 Endereçando os escravos da rede

A rede modbus funciona com um mestre e um ou mais escravos. Cada escravo na rede modbus deve ter o seu endereço, ou seja, o primeiro equipamento do varal é o escravo número um, o segundo é o escravo número dois e assim por diante. Cada equipamento que será um escravo na rede deve ser configurado para que ele possa registrar qual é o seu endereço.

A maioria dos equipamentos modbus podem ser configurados utilizando um software para computador. O Ircom, por sua vez, deve ser configurado utilizando o seu controlador *Conflex Light* mostrado na Figura 19. Embora para este projeto utilizemos o Simatic IOT2040 como o mestre que comanda o Ircom, ele tem o seu controlador próprio que faz o papel de mestre na rede controlando até 8 Ircoms. É por meio deste controlador que é possível configurar o endereço do escravo de cada Ircom. Para fazer esta configuração, basta conectar, com um cabo de duas vias blindado, um Ircom por vez com o controlador, e, utilizando as teclas e o display do *Conflex Light*, consegue-se endereçar o Ircom a ele conectado. Portanto, se o ambiente possui dois *Splits* de ar condicionado por exemplo, há a necessidade de utilizar 2 Ircons, sendo um o escravo um e o outro o escravo dois, e ambos salvam o seu endereço nas suas memórias através desta configuração com o *Conflex Light*.

Figura 19 – Controlador Conflex Light AGST.



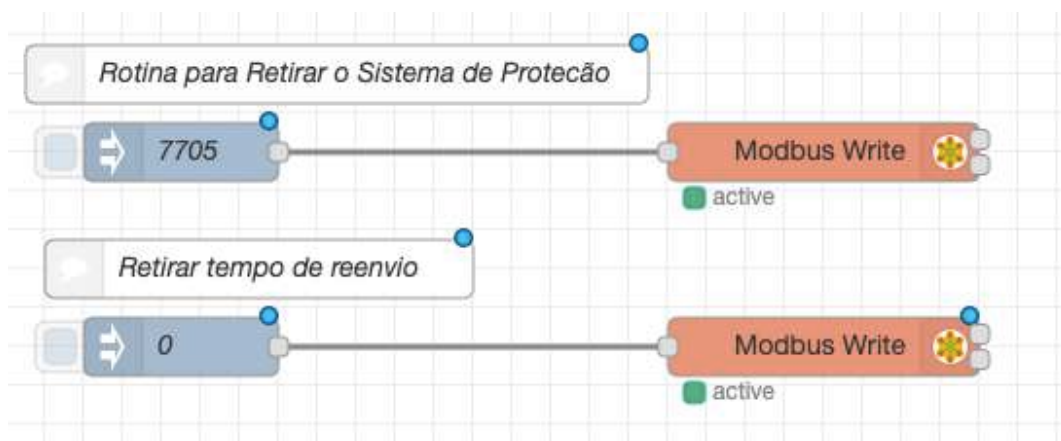
Fonte: (AGST, 2017).

3.6.3 Modificando configurações padrão

Durante a execução do projeto, percebeu-se duas peculiaridades do Ircom. Primeiramente, ele enviava um sinal infravermelho de dez em dez minutos para o ar condicionado. Este sinal não era nenhum comando, apenas um sinal para o ar condicionado emitir um sinal sonoro a fim de ver se a comunicação estava funcionando. Além deste sinal para “acordar a máquina”, o Ircom, depois de meia hora sem um mestre na rede enviava um sinal para ligar o ar condicionado num *setpoint* de 24 °C. Ambos os casos descritos precisavam ser mudados. A AGST deu o suporte necessário e explicou que para mudar estes sinais de envio, primeiramente precisaríamos fazer uma atualização do Firmware do Ircom. Eles enviaram o arquivo de atualização e utilizando o Software Uploader 2.0.1.015 e um conversor serial USB, foi possível atualizar o firmware.

No endereço 7 do mapa modbus do Ircom fica armazenado o tempo em que ele envia o sinal para que o ar condicionado emita um sinal sonoro. Com a atualização do firmware foi possível mudar o valor salvo neste endereço para zero. Desta maneira o Ircom não envia mais este sinal. Assim como o passo anterior, também com o firmware atualizado, para retirar o Ircom do sistema de proteção bastou escrever o valor “7705” no endereço 11. Ambas as escritas destes valores nos endereços puderam ser feitas pelo Node-Red como mostra a Figura 20. Uma vez modificado os valores nos registros, não é necessário escrevê-los novamente.

Figura 20 – Fluxos para alterar registros do Ircom.



Fonte: (Do autor, 2019).

3.6 Criar Aplicativo e *Dashboard*

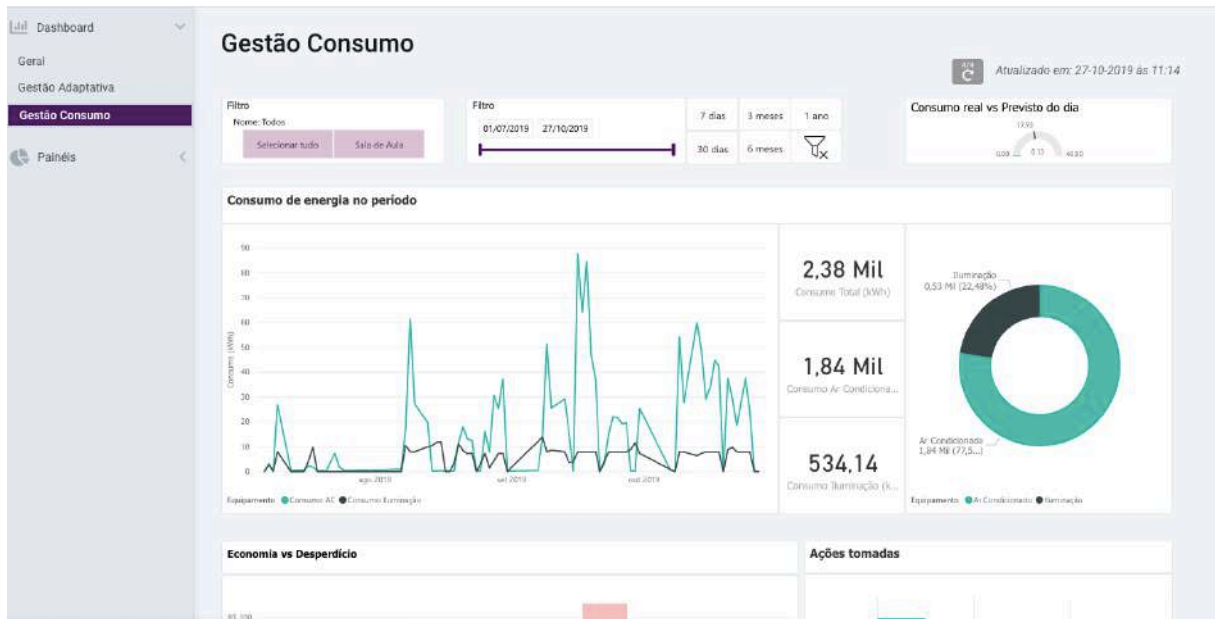
Para comunicar com o projeto, a empresa onde o projeto foi desenvolvido (Moka Mind) oferece um aplicativo, disponível para IOS e Androide, e um *Dashboard* que pode ser acessado tanto por um computador quanto pelo navegador do celular. É por meio do aplicativo que todos os comandos para controlar os ares condicionados e a iluminação são feitos. Além disso o aplicativo fornece informações de consumo energético do ambiente e possui uma inteligência artificial para gerar insights sobre o ambiente, dizendo os fatores que influenciaram no consumo e avisando o usuário de possíveis situações incomuns como um ar condicionado ligado em um horário não previsto por exemplo. Já o *Dashboard*, é uma plataforma que recebe todos os dados e *insights* e tem como função principal mostrar para os usuários a situação do ambiente ao longo do tempo, utilizando vários gráficos e textos informativos sobre o ambiente a partir do banco de dados do local.

O *backend e frontend* padrão do aplicativo e do *Dashboard* já estão criados e não fazem parte da abordagem deste trabalho. Quando o ambiente do projeto aqui discutido chega nesta etapa de produção, a lista de sinais é enviada para os responsáveis pela geração do aplicativo e *Dashboard*. A lista de sinais contém as informações necessárias para se criar o ambiente desejado a partir do aplicativo e *Dashboard* padrão. A Figura 21, mostra a tela geral do aplicativo e a Figura 22 mostra a tela de gestão de consumo do *Dashboard*.

Figura 21 – Tela geral do aplicativo.



Fonte: (Do autor, 2019).

Figura 22 – Tela gestão de consumo do *Dashboard*.

Fonte: (Do autor, 2019).

3.6 Montagem em bancada para teste

Após todos os procedimentos anteriores, o projeto está pronto para funcionar. A fim de se testar os equipamentos, a funcionalidade da lógica do Node-Red, a conexão com a internet, o broker e o aplicativo, os equipamentos são montados na bancada simulando o ambiente em que o projeto será instalado.

No caso deste projeto, o ambiente consiste em uma sala com um circuito de iluminação e um ar condicionado. Para isso, foi utilizado uma lâmpada de LED como mostrada na Figura 23 para simular a circuito de iluminação. Na instalação ao invés da lâmpada de LED um relé será utilizado, mas com o intuito de testar a lógica da programação, utilizar a lâmpada como indicador é suficiente. Um botão normalmente aberto como o da Figura 24, também é utilizado para o teste. Assim como a lâmpada de LED, este botão serve apenas para testar a lógica do projeto. A botoeira real e o relê são testados pelos montadores do painel. Os outros matérias utilizados no teste como o Ircom, o sensor de corrente, a fonte, o Simatic IO2000 com o cartão de memória pronto e o modem com o chip, são os materiais que serão utilizados projeto.

Figura 23 – Lâmpada de LED.



Fonte: (Do autor, 2019).

Figura 24 – Botão normalmente aberto.



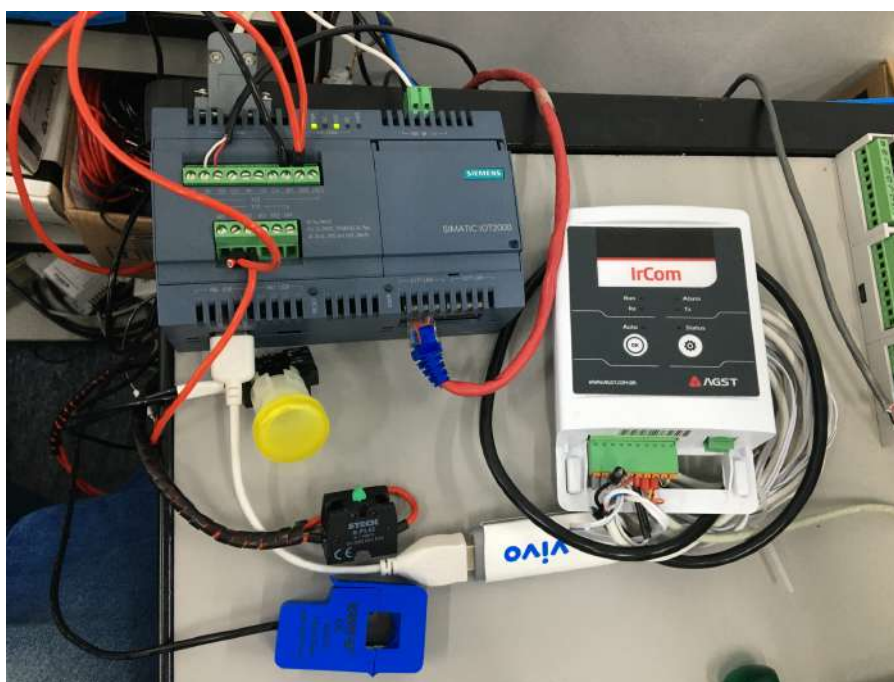
Fonte: (Do autor, 2019).

Com todos os materiais prontos para o teste, o produto é montado. A lâmpada é conectada em uma saída digital, o sensor é conectado em uma entrada analógica, o botão é conectado em uma entrada digital e um fio faz um *jumper* entre a saída digital que aciona a lâmpada e outra entrada digital. Este *jumper* serve para saber se a saída digital foi realmente acionada. Esta lógica de segurança está no *backend* do aplicativo. Se a saída digital é acionada

e a entrada digital a ela relacionada não muda o seu valor para o mesmo estado da saída digital, o aplicativo aponta um erro. Por isso, colocar este jumper é de extrema importância para o teste.

Além das conexões dos IOs, o Ircom é conectado a porta de comunicação serial do Simatic IOT2040. Seus dois sensores ficam sobre a bancada e o emissor infravermelho apontado para o ar condicionado da sala em que se está fazendo os testes. A Figura 25 mostra os equipamentos conectados e prontos para o teste.

Figura 25 – Projeto montado em bancada.



Fonte: (Do autor, 2019).

Depois de tudo montado, a fonte é alimentada e o Simatic inicia. Após aproximadamente 15 minutos ele já está no ar e conectado a internet. A fim de se testar tudo e comprovar a funcionalidade do projeto em geral, primeiramente todos os comandos são testados com os sinais sendo enviados diretamente dos fluxos do node-red. Para isso utiliza-se uma biblioteca chamada “*inject*” e ela é conectada aos fluxos e os valores que desejasse enviar para os equipamentos são forçados a serem enviados. A biblioteca “*debug*” que imprime os valores na plataforma do node-red é utilizada para conferir o que está sendo lido pelos sensores de temperatura do Ircom e pelo sensor de corrente. Para validar se os valores estão corretos, um multímetro e um termômetro são utilizados.

Após o teste enviando os valores pelo Node-Red, deve-se testar os comandos e a leitura dos sensores pelo aplicativo. Utilizando um celular com o aplicativo instalado, os comandos

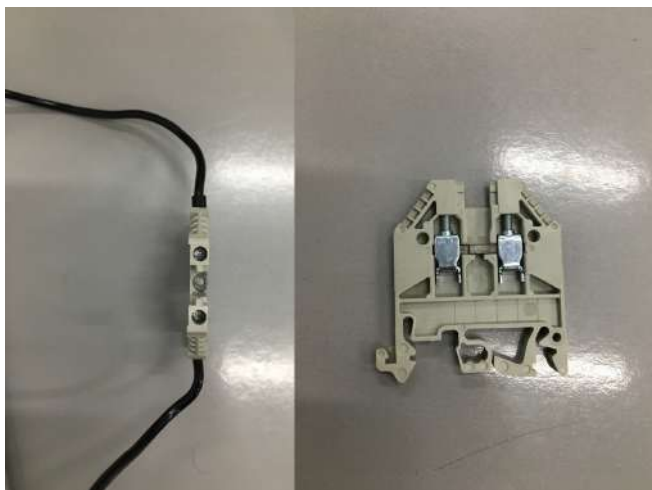
para testar todos os equipamentos são dados. Além dos comandos, os valores lidos pelos sensores que chegam no aplicativo também são validados. Como no caso anterior os comandos são confirmados observando se a luz ligou ou se o ar condicionado ligou ou mudou o *setpoint*. Já as leituras dos sensores são validadas utilizando um termômetro e um multímetro.

A medida em que se vai testando o projeto, um arquivo do padrão de qualidade da empresa chamado IDT, é preenchido. Ele determina todos os testes que devem ser feitos e nele é anotado todos os resultados. Quando todos os testes funcionam, o arquivo é assinado por um responsável e o quadro do projeto está liberado para ser montado para que ele vá para a instalação.

3.7 Montagem do quadro elétrico

Após o funcionamento de todos os equipamentos e da lógica do projeto, o quadro de comando com todos os *Hardwares* é montado. Os equipamentos, junto com o desenho elétrico do projeto, são enviados a equipe responsável pela montagem dos quadros. Eles colocam o disjuntor, a fonte, o Simatic IOT2040, o modem, os relés e os bornes nos trilhos do quadro. Os bornes como o da Figura 26 servem para facilitar a organização das ligações elétricas das fiações que vão entrar no quadro de comando. Além dos bornes para organizar as ligações, bornes com fusíveis também são colocados para as fiações que alimentam os equipamentos do quadro. Estes bornes fusíveis servem para proteger os Hardwares contra possíveis excesso de tensão na alimentação dos equipamentos prevenindo que eles queimem. Além de fixar os equipamentos nos trilhos, a botoeira é colocada na porta do quadro.

Figura 26 – Borne simples



Fonte: (Do autor, 2019).

Após fixar os equipamentos nos trilhos, as conexões utilizando fios vermelhos para o positivo e preto para o negativo, fazem as ligações elétricas entre os equipamentos. Estas ligações são feitas como manda o desenho elétrico feito no início do projeto.

A Figura 27 mostra uma foto do quadro de comandos montado.

Figura 27 – Quadro de comandos



Fonte: (Do autor, 2019).

3.8 Instalação

Com o projeto testado e funcionando, o quadro de comando pode ser instalado no ambiente do projeto. Na obra, o quadro é parafusado na parede num local que facilite a conexão dos cabos do ambiente que precisam ser conectados a ele. No caso da escola em que foi instalado, o quadro foi colocado na sala das condensadoras, pois nesta sala fica o quadro de energia do local, com os disjuntores do circuito de iluminação e do ar condicionado.

Os primeiros passos na instalação foram em relação ao Ircm. Primeiramente foi feito o modo de aprendizado dele para que ele fosse configurado para poder comandar o modelo do ar condicionado do local. Uma vez configurado, ele foi alimentado com 220V, conectado em

paralelo com a evaporadora, que se encontrava sob o forro da sala. O seu emissor infravermelho foi colado sobre o receptor infravermelho que fica anexado ao teto da sala. O seu sensor de temperatura de insuflamento foi colocado na saída de ar do ar condicionado. Já o seu sensor de temperatura de retorno foi colocado no teto da sala de aula, uma vez que ele precisa medir a temperatura ambiente do local. Um cabo blindado de comunicação serial foi conectado nas entradas responsáveis por fazer a conexão modbus e esse cabo foi levado até o local onde o quadro de comando foi instalado.

Na sala das condensadoras, o quadro foi parafusado, alimentado com 220V e aterrado. Para isso um disjuntor duplo foi colocado no quadro elétrico da escola para poder fornecer essa tensão.

Depois de conectar os cabos responsáveis por energizar o quadro, com ele ainda desligado, os fios responsáveis por acionar a iluminação foram passados sobre o forro, descendo por um eletroduto embutido na parede e entrando no quadro de comando, tudo isso para que fosse possível conectá-los aos bornes que os conectam ao relê.

O sensor de corrente foi colocado em torno do fio que passa a corrente do ar condicionado, que fica no quadro elétrico da escola, e o seu cabo foi passado por um eletroduto dentro da parede, que ia até o forro, e do forro ele desceu, por outro eletroduto, para o quadro de comandos. Dentro do quadro, os fios foram conectados aos bornes que os conectam a entrada analógica do Simatic IoT2040.

Finalmente, o cabo de comunicação que vem do Ircom sobre o forro, foi passado por um eletroduto descendo até o quadro de comando, e assim como os outros fios que entram no quadro, ele foi conectado aos bornes que os conectam a entrada de comunicação serial do Simatic IoT2040.

Todas as outras conexões internas do quadro já vêm prontas da montagem que foi feita anteriormente. O cartão de memória com o sistema operacional instalado, configurado e programado é colocado no Simatic e o Modem com o chip é conectado ao gateway.

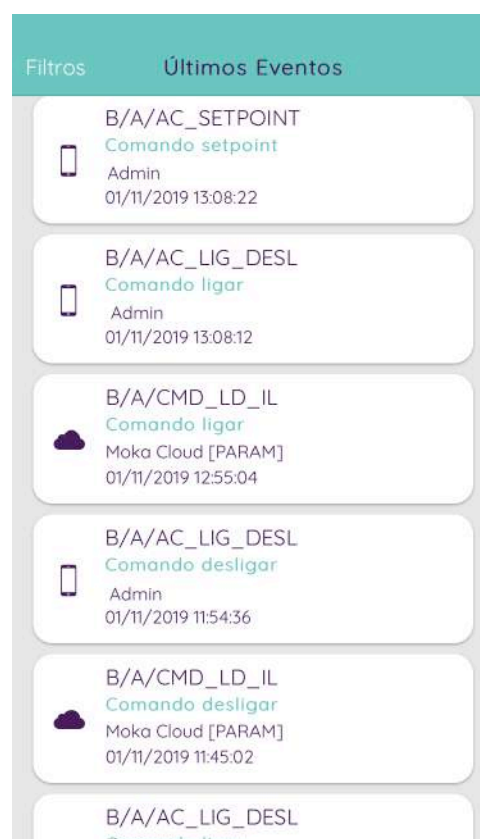
Após todos os procedimentos anteriores, o disjuntor do quadro foi ligado e após alguns minutos o projeto já estava online e pronto para o uso. Os testes de comandos foram feitos no local e após confirmar o funcionamento da automação do ambiente, o produto desenvolvido estava concluído e os usuários e senhas de acesso foram passados ao cliente.

4 RESULTADOS E DISCUSSÕES

Um tempo após a instalação do projeto, os resultados esperados foram obtidos. Hoje, os responsáveis pelo local onde o projeto foi instalado já utilizam todas as funcionalidades do mesmo.

A Figura 28 mostra um pequeno histórico de alguns comandos dados pelo usuário. Neste histórico pode-se ver que o cliente modificou o *setpoint*, ligou e desligou o ar condicionado e ligou e desligou a iluminação. Além disso, pelos ícones identificados por uma nuvem, percebe-se que o usuário programou comandos horários para que os equipamentos da sala de aula fossem controlados sem a necessidade de dar comandos pelo aplicativo. Este pequeno histórico comprova que os comandos funcionaram devido ao fato de nenhum evento de erro ter aparecido.

Figura 28– Últimos eventos no aplicativo da Moka Mind



Fonte: (Do autor, 2019).

Com tudo instalado e funcionando, os eventos foram analisados e *insights* foram gerados. Alguns exemplos destes *insights* gerados são apresentados na Figura 29. Eles detectaram eventos anormais, e utilizando o banco de dados do ambiente foi possível apontar os causadores do desperdício de energia, bem como calcular o gasto gerado por esses

acontecimentos anormais. Tudo isso para poder fornecer para o cliente um mapeamento do ambiente que está sendo analisado para evitar que estes eventos se repitam.

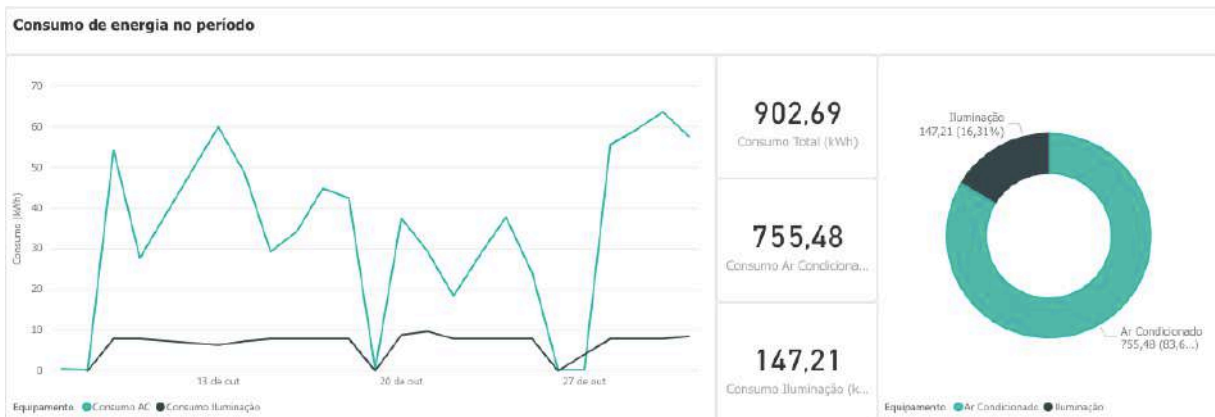
Figura 29 – Insights gerados.



Fonte: (Do autor, 2019).

A Figura 30 mostra as informações de consumo de energia elétrica do ambiente dos últimos trinta dias. As informações coletadas foram analisadas e comparadas com contas de consumo de eletricidade do ambiente, e com esta comparação comprovou-se que além dos comandos, a coleta de dados de consumo elétrico do ambiente também estava funcionando.

Figura 30 – Informações do consumo elétrico.



Fonte: (Do autor, 2019).

5 CONCLUSÃO

Como objetivo, o projeto abordado neste trabalho teve como ideal comandar iluminação e ar condicionado de uma sala de aula bem como coletar o gasto de energia elétrica de ambos. Esta tarefa foi concluída com êxito.

Este projeto facilitou o funcionamento e gerenciamento da sala de aula, pois com os comandos remotos e monitoramento, foi possível fazer com que o cliente programasse horários para ligar e desligar iluminação e ar condicionado através do seu celular. Além de poder ter definido os horários de funcionamento da sala de aula, caso alguém solicite um ajuste de temperatura do ambiente, o responsável por esta tarefa não tem mais a necessidade de ir até o ambiente para fazer o ajuste com o controle remoto.

Com o projeto instalado, a necessidade de um funcionário do local fazer uma ronda para ter que ir até a sala para conferir se o ar ou a luz foram desligados, passou a ser desnecessária, já que agora é possível monitorar como o ambiente está utilizando o celular. Caso a iluminação ou o ar condicionado precisem ser desligados não há a necessidade de ir até a sala por exemplo.

Outro fator que torna este projeto extremamente útil é o fato de que agora, o cliente pode monitorar o consumo energético do ambiente, e com a geração de *insights* ele pode entender porque o ambiente esta consumindo aquela quantidade de energia elétrica, apontando os fatores influenciadores que podem ser a temperatura externa, horas de iluminação ligada, horas de ar ligado,... Tudo isso contribui para a economia em dinheiro, beneficiando ainda mais o cliente.

Durante a execução do projeto, algumas dificuldades foram encontradas, e a maioria delas foi em relação ao Simatic IoT2040 e o seu sistema operacional Yocto. A dificuldade deste Hardware e o seu sistema operacional está na falta de conteúdo online sobre o produto. Por esse motivo, a maioria dos problemas encontrados em relação Gateway da Siemens demandaram um grande tempo de trabalho para solucioná-los.

Outro problema em relação a estes projetos IoT é em relação a conexão com a internet. Para o caso do projeto abordado, o ambiente de instalação era um lugar com um bom sinal 4G o que permitiu a boa funcionalidade do produto. Existem alguns clientes, em que os seus ambientes são desprovidos de um sinal de internet de qualidade e modems mais robustos e caros que o *Huaway* precisam ser usados, encarecendo o produto final. Como melhoria para estes casos, um estudo e testes da utilização da nova rede de comunicação *Narrowband* já está sendo feita. O *Narrowband* pode solucionar este problema, uma vez que ele é um novo tipo de banda específica para projetos IoT que promete uma boa melhoria quanto a conexão e o sinal.

Outro empecilho na criação deste projeto é em relação ao sensor de corrente utilizado. O SCT-013 utilizado é um transformador de corrente e por este motivo o seu sinal de saída é alternado. A fim de se coletar o valor eficaz da corrente, um fluxo de amostragem para se pegar o pico da corrente lida foi desenvolvido no Node-Red. O grande problema deste método é que ele gasta uma quantia significativa de processamento do Gateway (15%) e devido a amostragem, a leitura do sensor apresenta um tempo de resposta de aproximadamente cinquenta segundos para atualizar o valor coletado. A fim de se melhorar este problema, está sendo estudado e testado a possibilidade de utilizar outros sensores que possuem micro controladores integrados, que fazem esta amostragem. Apesar de serem mais caros, com esses modelos é possível receber um dado contínuo do sensor, sendo desnecessário a utilização da amostragem no Simatic IoT2040.

Outra dificuldade encontrada na criação deste produto é a limitação da quantidade de entradas e saídas analógicas e digitais do Simatic IoT2040. Devido a esta limitação, não é possível controlar muitos circuitos de iluminação e fazer a medição de muitos aparelhos de ar condicionado, o que limita o projeto a ambientes pequenos. Se há a necessidade de controlar ambientes maiores ou mais de um ambiente, é preciso utilizar mais de um Simatic IoT2040, que é o equipamento mais caro do projeto. Para resolver isso, estudos e testes de expansores de IOs por modbus estão sendo feitos. Hoje, dois produtos estão sendo testados. Utilizando estes equipamentos na rede modbus junto ao Ircom, a quantidade de IOs que o projeto poderá possuir, possibilitará que um único gateway, em conjunto com estes expansores, controle vários circuitos de iluminação e possa medir vários aparelhos de ar condicionado.

SILVA JÚNIOS, J, S **Bluetooth e infravermelho**. Mundo Educação, 2015. Disponível em <<https://mundoeducacao.bol.uol.com.br/fisica/bluetooth-infravermelho.htm>>. Acessado em 04/11/2019.

DEMETRAS, E. **SCT-013 – Sensor de Corrente Alternada com Arduino**. Vida de Silício, 2017 Disponível em <<https://portal.vidadesilicio.com.br/sct-013-sensor-de-corrente-alternada/>>. Acessado em 04/11/2019.

<https://www.makers-hut.com/product/sct-013-000-non-invasive-split-core-current-transformer-ac-current-sensor-100a/>

SUEIRO, D. **Anatomia de um Sistema Linux embarcado**. Embarcados, 2013. Disponível em <<https://www.embarcados.com.br/anatomia-de-um-sistema-linux-embarcado/>>. Acessado em 04/11/2019.

FREITAS, C, M. **Protocolo Modbus: Fundamentos e Aplicações**. Embarcados, 2014. Disponível em <<https://www.embarcados.com.br/protocolo-modbus/>>. Acessado em 04/11/2019.

BARROS, M. **MQTT - Protocolos para IoT**. Disponível em. Embarcados, 2015 <<https://www.embarcados.com.br/mqtt-protocolos-para-iot/>>. Acessado em 04/11/2019.

YUAM, M. **Conhecendo o MQTT**. IBM, 2017. Disponível em <<https://www.ibm.com/developerworks/br/library/iot-mqtt-why-good-for-iot/index.html>>. Acessado em 04/11/2019.

AGST. **Guia de instalação rápido módulo Ircom**. 2017

AGST. **Guia de instalação rápido controlador conflex light**. 2017

Ficha catalográfica elaborada pelo Sistema de Geração de Ficha Catalográfica da Biblioteca
Universitária da UFLA, com dados informados pelo(a) próprio(a) autor(a).

Freire, Matheus Aparecido Melo.

Projeto IoT para controle remoto de iluminação e ar
condicionado e monitoramento do consumo energético de salas de
aula / Matheus Aparecido Melo Freire. - 2019.

47 p. : il.

Orientador(a): Wilian Soares Lacerda.

TCC (graduação) - Universidade Federal de Lavras, 2019.
Bibliografia.

1. Internet das coisas. 2. Automação de salas de aula. 3.
Monitoramento energético. I. Lacerda, Wilian Soares. II. Título.