UNIVERSIDADE FEDERAL DO ESPIRITO SANTO
CENTRO DE CIÊNCIAS EXATAS, NATURAIS E DA SAÚDE
BACHARELADO EM CIÊNCIA DA COMPUTAÇÃO

JOSÉ PAULO DE AMORIM JÚNIOR

ILHA DE MEDIÇÃO DE GRANDEZAS ELÉTRICAS USANDO INTERNET DAS COISAS

ALEGRE
2018

JOSÉ PAULO DE AMORIM JÚNIOR

ILHA DE MEDIÇÃO DE GRANDEZAS ELÉTRICAS USANDO INTERNET DAS COISAS

Trabalho de Conclusão de Curso apresentado ao Departamento de Computação do Centro de Ciências Exatas, Naturais e da Saúde da Universidade Federal do Espírito Santo, como requisito parcial para obtenção do grau de Bacharel em Ciência da Computação.

Orientador: Prof. Dr. Claudio Moises Ribeiro

ALEGRE

2018

JOSÉ PAULO DE AMORIM JÚNIOR

ILHA DE MEDIÇÃO DE GRANDEZAS ELÉTRICAS USANDO INTERNET DAS COISAS

Trabalho de Conclusão de Curso apresentado ao Departamento de Computação do Centro de Ciências Exatas, Naturais e da Saúde da Universidade Federal do Espírito Santo, como requisito parcial para obtenção do grau de Bacharel em Ciência da Computação.

Aprovado em 20 Novembro de 2018.

COMISSÃO EXAMINADORA

Prof. Dr. Claudio Moises Ribeiro
Universidade Federal do Espírito Santo
Orientador

Prof. Dr. Helder de Amorim Mendes
Universidade Federal do Espírito Santo

Prof. MSc. Jacson Rodrigues C. da Silva
Universidade Federal do Espírito Santo

Dedico esse trabalho ao meu pai José Paulo de Amorim, a minha mãe Carmem Lúcia Rosa de Amorim e as minhas irmãs Paloma Rosa de Amorim e Paula Rosa de Amorim que, com muito carinho e apoio, não mediram esforços para que eu chegasse até essa etapa da minha vida.

AGRADECIMENTOS

A minha família por todo apoio e pela capacidade de acreditarem e investirem em mim. Mãe, seu cuidado e dedicação foi que me deram, em alguns momentos, a esperança para seguir. Pai, sua presença significou segurança e certeza de que não estou sozinho nessa caminhada.

Ao meu orientador Prof. Dr. Claudio Moises Ribeiro, pelo empenho dedicado a elaboração do presente trabalho que vem acontecendo desde 2016.

Ao Prof. Dr. Roberto Colistete Júnior, pelo apoio, confiança e por todo material que foi cedido para a elaboração do presente trabalho.

Ao meu coorientador Prof. Dr. Helder de Amorim Mendes, pelo total apoio oferecido para elaboração do presente trabalho.

Ao Prof. Ms. Jacson Rodrigues C. da Silva, por aceitar fazer parte da minha banca examinadora.

Agradeço a todos os professores por me proporcionarem o conhecimento não apenas racional, mas a manifestação do caráter e afetividade da educação no processo de formação profissional, por tanto que se dedicaram a mim, não somente por terem me ensinado, mas por terem me feito aprender. A palavra mestre, nunca fará justiça aos professores dedicados aos quais sem nominar terão os meus eternos agradecimentos.

Enquanto o conceito de vencedores existir, precisa também existir o de perdedores. O desejo egoísta de proteger a paz causa guerras e o ódio nasce para proteger o amor. Quando um homem aprende a amar, ele tem que suportar o risco de odiar.

Uchiha Madara (2012)

RESUMO

Este trabalho apresenta uma ilha de medição de grandezas elétricas e a aplicação do conceito de Internet das Coisas para monitorar e controlar, em tempo real, qualquer dispositivo conectado à ilha. A ilha de medição conta, essencialmente, com um circuito de condicionamento de sinais e um microcontrolador Wemos Lolin ESP32. O microcontrolador realiza cálculos que geram informações elétricas de consumo como potência média, fator de potência, valor eficaz de tensão e valor eficaz de corrente. As informações geradas pelo circuito são enviadas via protocolo MQTT do microcontrolador para um microcomputador Raspberry Pi. O microcomputador tem a função de monitorar a ilha recebendo as informações de consumo geradas, apresentando-as ao usuário, além de controlar o dispositivo conectado à ilha, enviando informações de acionamento para o microcontrolador. A ilha de medição e o microcomputador compõem o sistema integrado que pode ser usado para automação residencial e/ou industrial, para realizar o controle de consumo de qualquer aparelho elétrico conectado à ilha e ainda pode ser ampliado para ser acessado via Internet por algum servidor *online* podendo controlar vários dispositivos simultaneamente.

Palavras-chave: Internet das Coisas, medição de grandezas elétricas, MQTT, automação.

ABSTRACT

This paper presents an island of measurement of electrical quantities and the application of the Internet of Things concept to monitor and control in real time, any device connected to the island. The measuring island relies essentially on a signal conditioning circuit and a Wemos Lolin ESP32 microcontroller. The microcontroller performs calculations that generate electrical consumption information such as average power, power factor, effective voltage value and effective current value. The information generated by the circuit is sent via MQTT protocol of the microcontroller to a microcomputer Raspberry Pi. The microcomputer has the function of monitoring the island receiving the consumption information generated, presenting them to the user, besides controlling the device connected to the island, sending drive information to the microcontroller. The measuring island and the microcomputer make up the integrated system that can be used for residential and/or industrial automation to control the consumption of any electrical appliance connected to the island and can also be expanded to be accessed via the Internet by an online server being able to control several devices simultaneously.

Keywords: Internet of Things, measurement of electrical quantities, MQTT, automation.

SUMÁRIO

1 INTRODUÇÃO 10

2 OBJETIVOS 13

2.1 OBJETIVO GERAL 13

2.2 OBJETIVOS ESPECÍFICOS 13

3 REFERENCIAL TEÓRICO 14

3.1 MICROCONTROLADORES 14

3.2 MICROCOMPUTADORES 15

3.3 PROTOCOLO DE COMUNICAÇÃO MQTT 15

3.4 AUTOMAÇÃO 16

3.5 COMPUTAÇÃO EMBARCADA 17

3.6 IOT CLOUD 18

3.7 USO DE MICROCONTROLADORES E IOT PARA MEDIÇÃO DE GRANDEZAS ELÉTRICAS 19

4 METODOLOGIA 21

4.1 HARDWARE 21

4.1.1 CIRCUITO DE CONDICIONAMENTO DE SINAIS 21

4.1.2 MICROCONTROLADOR WEMOS LOLIN ESP32 22

4.1.3 MICROCOMPUTADOR RASPBERRY PI 23

4.2 SOFTWARE 25

4.2.1 BIBLIOTECAS 25

4.3 SISTEMA 26

5 DESENVOLVIMENTO 28

6 RESULTADOS 37

7 CONCLUSÃO 47

REFERÊNCIAS 48

ANEXOS 52

ANEXO A – FIGURA DO MICROCONTROLADOR WEMOS ESP8266 D1 MINI 53

ANEXO B – QUADRO DE ESPECIFICAÇÕES WEMOS ESP8266 D1 MINI 54

ANEXO C – FIGURA DO MICROCONTROLADOR WEMOS LOLIN ESP32 55

ANEXO D – QUADRO DE ESPECIFICAÇÕES DO MICROCONTROLADOR WEMOS LOLIN ESP32 56

ANEXO E – FIGURA DO MICROCOMPUTADOR RASPBERRY PI 2B 57

ANEXO F – QUADRO DE ESPECIFICAÇÕES DO MICROCOMPUTADOR RASPBERRY PI 2B 58

1 INTRODUÇÃO

A Internet das Coisas (*Internet of Things* ou IoT) surgiu da necessidade de conectar “coisas” (Objetos) do dia a dia à Internet, manifestou-se em meio as milhares de conexões de dispositivos e veio para facilitar a vida das pessoas. Como novo conceito de uso da Internet, ela passa a embarcar não só computadores, como também objetos do dia a dia (MATTERN; FLOERKEMEIER, 2010). É um assunto que vem se difundindo em todo o mundo e é a base para se criar projetos de soluções em diversas áreas como planejamento urbano, produção agrícola, grandes produções industriais, logísticas, preservação do meio ambiente etc.

O cientista da computação Mark Weiser, criador do conceito Computação Ubíqua, defendia o oposto da Realidade Virtual em que as pessoas são colocadas dentro do mundo gerado pelo computador. Na Computação Ubíqua, é o computador que se integra a vida das pessoas de modo que elas não percebam, mas o utilizem (WEISER, 1993).

Segundo Weiser (1999, p. 3), “As tecnologias mais profundas e duradouras são aquelas que desaparecem. Elas dissipam-se nas coisas do dia-a-dia até tornarem-se indistinguíveis”. Os computadores estão se tornando tão pequenos, presente em tantos lugares que nem percebemos. Isso é resultado da evolução e da necessidade de usar dispositivos em lugares de tal forma que eles se tornem imperceptíveis. A IoT conceitua a ideia de objetos conectados à Internet e entre si, através da rede, munido de sensores, controladores, circuitos eletrônicos e softwares capazes de coletar, processar e trocar dados e informações.

Existem diversas aplicações onde a IoT é usada para o fortalecimento de sistemas urbanos em geral, criando assim um grande sistema distribuído autônomo e cooperativo. Essas redes permitem verificar uma variedade de fenômenos que são descritos por algumas grandezas físicas como temperatura, pressão e humidade. Além das grandezas físicas, podem ser capturadas também grandezas “abstratas” como captura de faces, íris ou digitais, lugares visitados e registrados, localização de veículos em uma estrada e até pessoas em uma multidão (AQUINO, 2015).

Um objeto no qual se aplica o conceito de IoT pode ser qualificado como inteligente, devido a característica de poder se comunicar com outros objetos e de poder fazer isso de forma autônoma. De acordo com os dados da Tabela 1 segundo

a Gartner (EGHAM, 2017), a previsão é que a partir de 2020 mais de 20,4 bilhões de objetos inteligentes estarão conectados à Internet.

Tabela 1 – Unidades de IoT instaladas separadas por categoria (unidade em milhões)

Categoria	2016	2017	2018	2020
Consumidores	3.963,0	5.244,3	7.036,3	12.863,0
Empresas	1.102,1	1.501,0	2.132,6	4.381,4
Indústrias	1.316,6	1.635,4	2,027,7	3.171,0
Total	6.381,8	8.380,6	11.196,6	20.415,4

Fonte: Gartner (Janeiro, 2017).

Os gastos das empresas em IoT somaram 57% dos gastos gerais de IoT em 2017. Conforme a Tabela 2, os gastos com itens conectados entre as empresas e as indústrias chegaram a US$ 964 bilhões e os gastos com aplicações para consumidores chegaram a US$ 725.7 bilhões em 2017. Até 2020, os gastos de ambos os segmentos atingirão quase US$ 3 trilhões (EGHAM, 2017).

Tabela 2 - Ponto de extremidade dos gastos em IoT por categoria (milhões de dólares)

Categoria	2016	2017	2018	2020
Consumidores	532.515	725.696	985.348	1.494.466
Empresas	212.069	280.059	372.989	567.659
Indústrias	634.921	683.817	736.543	863.662
Total	1.379.505	1.689.572	2.094.881	2.925.787

Fonte: Gartner (Janeiro, 2017).

Em 2017 houve um salto de 31% em relação a 2016, com 8,4 bilhões de coisas conectadas, onde esses serviços renderam em torno de $ 1.7 trilhões em

2017. A China, a América do Norte e a Europa Ocidental juntas representaram 67% do total de Internet das Coisas instaladas no mundo em 2017 (EGHAM, 2017).

Os gastos das empresas com itens conectados podem ser considerados gastos envolvendo dispositivos como microcontroladores e microcomputadores, produzidos especificamente para aplicações envolvendo IoT. Já o gasto com aplicações para consumidores podem ser considerados como gastos com softwares e serviços que complementam o uso da IoT. Existem alguns serviços disponíveis na Internet que foram produzidos exatamente com o propósito de armazenar dados de microcontroladores e dispositivos que coletam dados e geram informações ligadas à Internet das Coisas. Alguns desses serviços que complementam os gastos gerais com uso de IoT são chamados de IoT *Cloud*.

O uso de serviços em geral, voltados a IoT, juntamente o conceito de automação, seja ela industrial ou residencial, é capaz de entregar ao usuário deste conjunto de serviços e aplicações algumas vantagens, como: maior conforto devido ao uso de microcontroladores; melhor usabilidade, podendo ser monitorado e controlado a distância; aumento de produtividade, quando se troca aplicações manuais por aplicações automáticas; maior controle de qualidade, usando variáveis programadas podendo atingir melhores resultados e melhor custo benefício graças aos baixos custos usando microcontroladores e microcomputadores com propósito de uso específico em Internet das Coisas, alcançando com essas vantagens uma gama de possibilidades práticas e econômicas.

2 OBJETIVOS

Nesta seção, serão descritos os objetivos estabelecidos para este projeto.

2.1 OBJETIVO GERAL

Este projeto visou à construção de uma ilha de medição de grandezas elétricas, capaz de captar dados de um dispositivo elétrico ligado à ilha e gerar informações de consumo que são de fácil entendimento para o usuário. Além disso, a ilha de medição tem a capacidade de controlar um dispositivo conectado a ela, ou seja, receber os dados de controle enviados pelo usuário e repassá-los ao dispositivo.

Por meio de algum serviço online ou até um microcomputador, o usuário é capaz de acessar as informações de consumo produzidas pela ilha de medição e controlar um dispositivo conectado a ela. Desta forma, o usuário consegue ter o controle e o acesso de todas as informações geradas pelo circuito. A comunicação entre a ilha de medição e o usuário é feita por meio de uma conexão local via WiFi ou pela Internet.

2.2 OBJETIVOS ESPECÍFICOS

- Construção de um circuito de condicionamento de sinais.
- Adaptação de um microcontrolador ao circuito construído.
- Programação de um microcontrolador para realizar a leitura dos dados do circuito construído.
- Exposição dos dados processados para o usuário através de algum serviço *offline.*
- Comunicação do microcontrolador com um microcomputador para realizar o envio dos dados processados.
- Comunicação do microcomputador com o microcontrolador para realizar o controle do dispositivo conectado a ilha.

3 REFERENCIAL TEÓRICO

Nesta seção constam os referenciais teóricos que serviram de base para a pesquisa e elaboração do presente trabalho.

3.1 MICROCONTROLADORES

Microcontroladores são elementos essenciais para se aplicar o conceito de IoT, podendo receber informações de sensores, realizar o controle de motores básicos e outros dispositivos que criam mudanças físicas, ser capaz de se conectar à Internet, enviar informações para computadores, microcomputadores, outros microcontroladores, aplicativos de celular etc. Eles agem como *gateways* entre o mundo físico e o mundo da computação e podem ser encontrados em muitos dispositivos eletrônicos.

Um exemplo de microcontrolador clássico e pioneiro no mundo da IoT, que se tornou famoso por suas dimensões, peso, custo e funcionalidades é a ESP8266EX (ESPRESSIF, 2018). Com a sua versão ESP8266 D1 mini (WEMOS, 2018) medindo apenas 34,2 milímetros de comprimento e 25,6 milímetros de largura, pesando 3g, custando em média de R$ 10,00 a R$ 20,00, a ESP8266 D1 mini é capaz de se conectar à Internet, captar dados e gerar informações das coisas conectadas a ela através de seus canais digitais.

A D1 mini se tornou uma referência no mundo da IoT. Por ser pequena e barata, se torna facilmente adaptável, permitindo explorar projetos de localização específica nos lugares mais improváveis ou desenvolver projetos que se podem acionar vários dispositivos simples, que resultam em um grande sistema integrado graças aos seus canais analógicos e seu canal digital.

Com a evolução da IoT e a evolução das coisas que são conectadas, o dispositivo ESP8266 passou a se tornar obsoleto, dando espaço para suas versões mais novas como a ESP32, especificamente a versão Wemos Lolin ESP32 (WEMOS, 2018) que passou a ter além da expansão dos canais ADC (Analogic Digital Conversor) que antes era apenas 1 na ESP8266 para 12 canais, a expansão dos canais digitais de 11 para 26, aumentando ainda mais a capacidade de controlar dispositivos conectados a ela. A ESP32 também conta com a tecnologia de

comunicação *Bluetooth*, aumentando também a sua capacidade de comunicação sem fio que antes só contava com Wi-Fi (WEMOS, 2018).

Com as novas tecnologias no mercado é possível adquirir microcontroladores para uma grande quantidade de funções, seja ela para medir temperatura, pressão, humidade, tensão, corrente etc.; deste modo, captando dados e gerando informações que podem ser disponibilizadas pra que o usuário possa interpretar mudanças físicas. Combinando o uso de microcontroladores com o uso de microcomputadores é possível construir grandes sistemas de automação, no qual os microcontroladores usando uma comunicação Wi-Fi podem se comunicar com o microcomputador que, por sua vez, podem atuar como uma central inteligente, monitorando todos os microcontroladores envolvidos nesses sistemas.

3.2 MICROCOMPUTADORES

Microcomputadores são essenciais para o processamento de dados complexos em sistemas que aplicam o conceito de IoT e se tornam ferramentas indispensáveis por serem capazes de processar informações de vários dispositivos simultaneamente, ter um baixo consumo de energia, além de servirem como uma central de monitoramento em sistemas autônomos. Dentre os vários microcomputadores no mercado temos a Raspberry Pi (RASPBERRY PI, 2015), PocketBeagle (BEAGLEBOARD, 2018) e o Arduino Uno (ARDUINO, 2018).

Microcomputadores são geralmente usados para gerenciar os dados recebidos e enviados por algum microcontrolador usando um protocolo de comunicação específico, como por exemplo, o protocolo de comunicação MQTT, muito usado em IoT. Além de enviar e receber dados de um serviço qualquer como IoT *Cloud* ou alguma base de dados *online* por meio da Internet, são muito usados em sistemas de automação devido a sua capacidade de se comunicar com vários microcontroladores simultaneamente.

3.3 PROTOCOLO DE COMUNICAÇÃO MQTT

O protocolos de comunicação são maneiras que os dispositivos eletrônicos usam para se comunicar entre si ou com softwares e outros dispositivos. Existem

vários protocolos de comunicação usados no mundo da IoT como NFC (NFC, acesso em 09 dez. 2018), o famoso *Bluetooth* (BLUETOOTH, acesso em 09 dez. 2018) e o MQTT (MQTT, acesso em 09 dez. 2018).

Para realizar a comunicação entre o microcontrolador e o microcomputador, o protocolo MQTT se mostra capaz de atender a expectativas devido a sua topologia Publishe/Subscribe, versatilidade e compatibilidade com a maioria dos microcomputadores e microcontroladores disponíveis no mercado. O princípio de funcionamento protocolo MQTT parte da utilização de um Servidor MQTT que é implementado geralmente nos microcomputadores, também conhecido como Broker MQTT. Esse servidor implementa estruturas de armazenamento de dados, referenciadas como tópicos. Cada tópico refere-se a uma estrutura de publicação (Publishe) de dados, mas também possibilita a leitura (Subscribe) dos dados (FIREWARE LAB, 2016).

A estrutura do MQTT torna o envio e a segurança dos dados mais prático, conectando microcontroladores a tópicos específicos que são controlados por usuários específicos daquele tópico. Isso faz com que as informações geradas e enviadas pelos dispositivos sejam facilmente distinguidas pelos usuários, além de se manterem seguras. Com o protocolo de comunicação MQTT aplicado a microcontroladores e microcomputadores é possível automatizar processos e aparelhos eletrônicos, aplicando desta forma o conceito de automação.

3.4 AUTOMAÇÃO

O conceito de automação (industrial e/ou residencial) usando controladores vem se difundindo em todo o mundo. Com o avanço da tecnologia e com o uso da IoT é possível com um microcontrolador e algumas linhas de código automatizar processos dentro de uma casa ou uma indústria.

Um exemplo de sistema de automação construído por Panth e Jivani (2013) apresenta um circuito eletrônico que possui vários relés de energia, cada um interligado a uma tomada em uma casa, esses relés estão conectados à saída digital de um dispositivo *Bluetooth,* desta forma é possível controlar os reles apenas usando um aplicativo para celular Android.

A automação residencial pode ser caracterizada pelo conjunto de serviços proporcionado por sistemas tecnológicos integrados como o melhor meio de satisfazer as necessidades básicas de segurança, comunicação, conforto de uma habitação e gestão energética na qual é o tema deste presente trabalho (MURATORI; DAL BÓ, 2011).

Segundo a Fluxo Consultoria (PIRAGIBE, 2016), atualmente, são definidos 3 níveis de automação residencial: sistemas autônomos, sistemas integrados, e sistemas complexos. Sistemas autônomos são capazes de operar de forma independente, exercendo todas as funções para os quais foram projetadas sem que haja necessidade de alguém os controlando. Sistemas integrados são sistemas que conseguem operar utilizando uma central inteligente programada por um microcomputador. Sistemas complexos utilizam a integração total dos dispositivos e eletroeletrônicos, sendo que para seu uso é necessário planejamento, que é feito a partir do projeto de construção ou de uma reforma do imóvel.

O processo de automação conta com o uso de dispositivos com características próprias, sendo simples e pequenos de forma a passarem imperceptíveis aos olhos humanos sendo tão pequenos que podem ser embutidos em fiações e tomadas, tomando o nome de *Embedded System* (Computação Embarcada).

3.5 COMPUTAÇÃO EMBARCADA

A computação embarcada possui microcontroladores e microcomputadores tão pequenos que se tornam encapsulados dentro do dispositivo, controlando e monitorando seu funcionamento. As informações de controle partem de sensores que são capazes de captar as informações do ambiente como humidade, temperatura, corrente, tensão, pressão, dentre outros. Seus usos são vastos e vão desde controladores em um casa, até o ensino em escolas, como, por exemplo, o sistema de aprendizado sobre Luz e Cor proposto por Rubim (2014), que tem como objetivo usar microcontroladores no ensino de Física para o ensino médio em uma escola pública.

Segundo o Guia do Hardware (DELAI, 2013) sistemas embarcados compõem o maior uso da computação no mundo, superam e muito o número de PCs, notebooks, servidores e correlatos. Eles estão por toda parte, no telefone, carro, bicicleta, roteador WiFi e até em óculos como o Google Glass (ROUSE, 2013). Sistemas embarcados podem ou não interagir com usuários humanos, dependendo de seu objetivo. A ideia é que um sistema embarcado seja capaz de funcionar sem que ele possa ser notado.

Os sistemas de automação complexos que podem necessitar da reforma do imóvel, onde na maioria das vezes, usam o conceito de computação embarcada para esconder seus dispositivos, tornando o ambiente na qual esse sistema foi aplicado menos poluído visualmente, por não conter mais fiações e cabos passando pelas paredes ou até interruptores expostos.

Alguns sistemas de automação, que contam com computação embarcada, acabam tendo a possibilidade de usufruir de alguns serviços *online* para armazenar dados e informações de sensores para maior conforto do usuário, que pode acessar esses dados através da Internet, esses serviços são chamados IoT *Cloud*.

3.6 IOT CLOUD

IoT *Cloud* (do inglês: Nuvem IoT) é uma plataforma *online* que é projetada para armazenar e processar dados de Internet das Coisas. A nuvem é criada para suportar grandes volumes de dados gerados por dispositivos como: sensores, sites, turbinas, microcontroladores e computadores. Também tem a função de responder em tempo real a esses dados e informações (ROUSE, 2018).

Com a criação de vários dispositivos conectados à Internet por todo o mundo, a produção de dados cresceu em larga escala. É necessário que essa enorme quantidade de dados produzidos pelos serviços IoT sejam armazenados e processados. Algumas empresas largaram na frente e foram capazes de gerar serviços que possuem garantia e que são funcionais. Alguns desses serviços são capazes de disponibilizar ao usuário funcionalidades prontas sem que ele precise desenvolver, sendo compatíveis com a grande maioria dos dispositivos IoT, entregando ao usuário um serviço com baixo custo de manutenção, atualizações

constantes, segurança e inovação. Alguns dos grandes serviços disponíveis na Internet são conhecidos como:

- Amazon Web Service (Amazon, 2018).
- Microsoft Azure (Microsoft, 2018).
- IBM Watson (IBM, 2018).

Esses serviços possuem versões gratuitas para que seus usuários possam experimentar suas funcionalidades, porém, possuindo várias restrições como por exemplo no caso do Azure, que conta com apenas 5GB de armazenamento de arquivos e restrições de 10 aplicativos por usuário, dentre outras restrições e o mesmo vale para o *Web Service* da Amazon. A maioria dos serviços *IoT Cloud* disponibiliza ao usuário versões grátis com restrições de seus sistemas por tempo determinado de aproximadamente 1 ano.

Os tipos de informações que são produzidas para serem armazenadas e disponibilizadas nestes tipos de serviços online descendem dos microcomputadores e microcontroladores que tem a capacidade de transformar os dados captados do ambiente em informações para os usuários.

3.7 USO DE MICROCONTROLADORES E IOT PARA MEDIÇÃO DE GRANDEZAS ELÉTRICAS

A ilha de medição de grandezas elétricas apresentada no presente trabalho tem a capacidade de ler e processar dados do tipo elétrico usando um circuito de condicionamento de sinais e um microcontrolador. No mercado atual existe um tipo de produto comercializado semelhante, o Sonoff Pow (ITEAD, 2017). O Sonoff Pow, de acordo com a especificações do fabricante[1], dentre outras funcionalidades, é capaz de monitorar em tempo real a potência instantânea em Watts e o consumo de energia elétrica em KWh, além de controlar (ligar e desligar) qualquer dispositivo conectado a ele.

O Sonoff Pow possuí um microcontrolador WiFi baseado no microcontrolador ESP8266 para realizar os cálculos e a conexão com a Internet. Os

[1] Disponível em: https://www.itead.cc/sonoff-pow.html

usuários podem controlar e monitorar os dispositivos conectados a ele de qualquer lugar do mundo por meio da Internet, com a ajuda de um software chamado eWeLink[2] que auxilia na configuração e na comunicação do dispositivo com a Internet.

O Sonoff Pow traz consigo algumas desvantagens quando comparada ao presente trabalho, não disponibilizando outros tipos de cálculo para o usuário como fator de potência, valor eficaz tensão, valor eficaz da corrente. A fabricante permite que softwares de terceiros sejam inseridos em seu firmware, porém, não disponibilizam o código fonte para os usuários do serviço, impossibilitando assim que usuários possam aprimorar os serviços ou explorar os cálculos que são feitos pelo sistema.

[2] Disponível em: https://sonoff.itead.cc/en/ewelink

4 METODOLOGIA

Neste capítulo, foram feitas descrições breves dos caminhos seguidos para a realização deste trabalho.

4.1 HARDWARE

A ilha de medição de grandezas elétricas é o hardware principal tendo sido projetado e montado especificamente para o projeto. Nesta seção serão descritos os componentes básicos e estratégias adotadas.

4.1.1 CIRCUITO DE CONDICIONAMENTO DE SINAIS

De forma resumida, os sinais elétricos de interesse são os valores instantâneos de corrente e tensão. A partir destes valores é possível calcular indiretamente as diversas outras grandezas pretendidas. Pode-se caracterizar tensão e corrente típicas de um eletrodoméstico conectado à rede elétrica da seguinte forma: são funções periódicas, alternadas e com frequência de 60Hz, tensão nominal de 127 ou 220 Volts (valor eficaz) e corrente definida pela potência do equipamento utilizado. Para este projeto, optou-se por otimizar o circuito de condicionamento de sinais para a tensão de 127 Volts, o que implica, para uma função senoidal, uma amplitude de 180 Volts (ou seja, um sinal que pode variar entre -180 e +180 volts). Para a corrente, optou-se por otimizar o circuito para amplitude de até 10 Ampères, compatível com uma ampla gama de eletrodomésticos.

Portanto, foi necessário um circuito de condicionamento de sinais capaz de compatibilizar estas grandezas com as características de entrada do canal ADC do microcontrolador. Este circuito já estava disponível e foi integrado ao microcontrolador visando à implementação da ilha como planejado.

Para a tensão foi necessário implementar um divisor que compatibilizasse uma faixa de 360 Volts pico-a-pico, deixando alguma margem de tolerância para sobretensão, com a faixa de tensão permitida na entrada do canal ADC que é de 0 a 3,3 Volts. Mas simplesmente reduzir a amplitude do sinal não é suficiente, pois o

canal ADC não aceita tensão negativa. Daí a necessidade de se introduzir um *off-set*, sobrepondo o sinal alternado a um valor DC igual à metade do valor máximo permitido na entrada do canal ADC (3,3 Volts / 2 = 1,65 volt).

Raciocínio equivalente foi usado para a corrente com a diferença que, neste caso, o circuito precisa ainda converter corrente para tensão, o que foi conseguido por intermédio de um resistor de valor conhecido. Dado que a tensão em um resistor é proporcional à corrente que passa pelo mesmo, então ele pode ser usado como sensor de corrente, bastando que sua resistência seja conhecida.

O circuito de condicionamento de sinais é baseado no amplificador operacional LM324[3] e permite ajuste do ganho e do *off-set*. Os ganhos dos amplificadores são aplicados de forma inversa no software do microcontrolador, permitindo a recuperação dos valores reais das grandezas medidas, corrente e tensão.

4.1.2 MICROCONTROLADOR WEMOS LOLIN ESP32

O microcontrolador é o dispositivo responsável por coordenar a ilha de medição de grandezas elétricas. Ele tem a capacidade de ligar e desligar quaisquer dispositivos eletrônicos que estiverem conectados à ilha através dos relés de energia; ler e interpretar os dados de tensão e corrente que estiverem passando no circuito quando algum dispositivo eletrônico estiver em funcionamento e realizar os cálculos de potência necessários a partir dos dados lidos e então gerar as informações de interesse do usuário.

Inicialmente escolheu-se trabalhar com um microcontrolador ESP8266 D1 Mini apresentado nos ANEXOS A e B, porém pode-se observar na Tabela 3 que se comparado com a Wemos Lolin ESP32 representadas nos ANEXOS C e D a ESP8266 conta apenas com 1 canal ADC tornando impossibilitadas as leituras dos valores de tensão e corrente simultaneamente, já a ESP32 possuí 12 canais ADC possibilitando a leitura simultânea dos valores de tensão e corrente usando 2 canais separados.

---

[3] Especificação do dispositivo disponível em: <http://www.alldatasheet.com/datasheet-pdf/pdf/17871/PHILIPS/LM324.html>

Ainda na Tabela 3, outros pontos negativos da ESP8266 se comparados com a ESP32 são: a precisão dos canais ADC que é de apenas 10 bits (2^{10} bits) representando apenas valores lidos entre 0 e 1023, já a precisão dos canais ADC da ESP32 são de 12 bits (2^{12} bits) podendo representar valores entre 0 e 4095 (Essa diferença na precisão dos canais ADC é fundamental na interpretação dos dados lidos no microcontrolador pois quanto maior o intervalo de leitura, mais preciso são os dados lidos.); e a capacidade de processamento da ESP32 que pode chegar a 240Mhz, uma diferença de 80Mhz se comparada com a ESP8266.

Tabela 3 - Comparação das especificações da ESP8266 com a ESP32

Propriedades	**Wemos ESP8266 D1 Mini**	**Wemos Lolin ESP32**
Canais ADC	1	12
Resolução dos Canais ADC	2^{10} bits = 0 a 1023	2^{12} bits = 0 a 4095
Clock (MAX)	160 MHz	240 MHz

Fonte: Autor (2018).

Após ler e interpretar os dados de tensão e correntes processados pelo circuito de condicionamento de sinais, a ESP32 processa esses dados gerando informações compreendidas pelo usuário. Os usuários interessados nas informações pode acessá-las através de algum servidor *online* criado na própria ESP32 e acessado via conexão local, ou enviados para um microcomputador. A ESP32 também recebe dados de acionamento de dispositivos dos usuários através do microcomputador.

4.1.3 MICROCOMPUTADOR RASPBERRY PI

O microcomputador é ser responsável por monitorar a ilha de medição, enviando dados de acionamento de dispositivos dos usuários e recebendo as informações elétricas que foram geradas pelo microcontrolador ESP32. O microcomputador escolhido para trabalhar foi o Raspberry Pi 2B apresentado nos ANEXOS E e F. O único motivo pertinente da escolha deste hardware foi a disponibilidade do mesmo para o uso no projeto, deixando evidente que existem

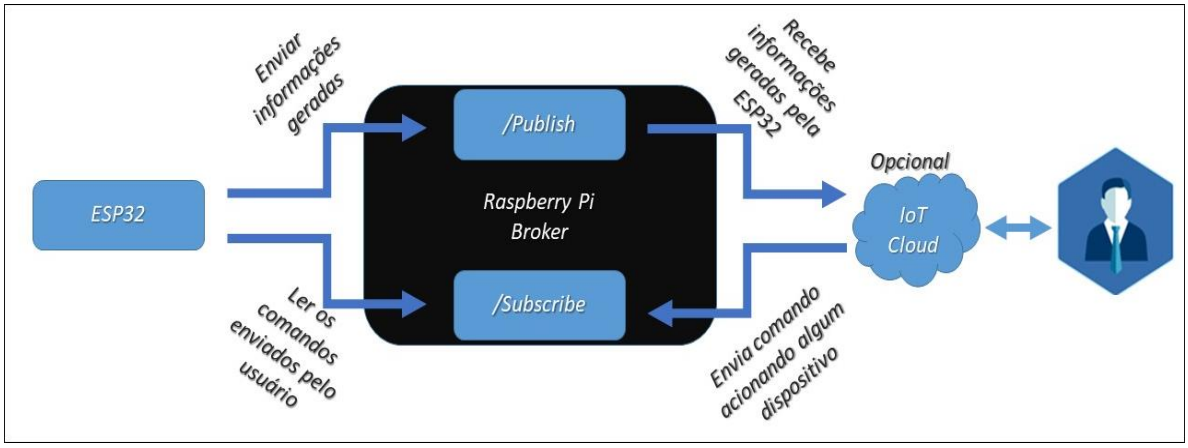
outros microcomputadores no mercado que poderiam ser usados para tal procedimento.

A Raspberry Pi poderá também ter algumas funções como: enviar as informações recebidas da ilha de medição para algum servidor *online* que por sua vez exibirá para os usuários e receber dados de acionamento de aparelhos conectados a ilha através dos relés dos usuário.

A comunicação entre microcomputador Raspberry Pi e o microcontrolador ESP32 que compõe a ilha é feita usando a rede local, por meio do protocolo de comunicação MQTT, tornando o microcomputador responsável por manter um servidor interno que gerencia esse protocolo de comunicação.

Como representado na Figura 1, o protocolo MQTT é implementado de forma que a ESP32 seja capaz publicar as informações geradas no tópico /publish do *broker* que foi criado na Raspberry Pi, e se inscrever no tópico /subscribe para ler os dados de acionamento de dispositivos enviados pelos usuário.

Figura 1 - Estrutura de funcionamento do protocolo MQTT



Fonte: Autor (2018).

Ainda na Figura 1, o servidor *broker* programado na Raspberry Pi fica encarregado de repassar as informações recebidas da ESP32 ao usuário que fizer a requisição e receber as informações de acionamento que podem ser enviadas pelo usuário usando a Raspberry Pi. A comunicação do *broker* com o usuário poderá ser feita usando algum servidor intermediário como um sistema de IoT *Cloud*.

4.2 SOFTWARE

O software foi implementado para rodar no microcontrolador ESP32 que fica acoplado à ilha de medição. A programação do software é feita usando a linguagem Arduino e a IDE de desenvolvimento Atom (ATOM, 2018) que é compatível com a ESP32. O software tem função de coordenar a ilha e captar os dados de tensão e corrente que passam pela ilha no momento em que algum aparelho conectado a ela estiver em funcionamento. Com os dados captados o software implementado realiza cálculos envolvendo os valores captados, de forma que consiga gerar as informações necessárias ao usuário como tensão e corrente eficaz, potência média e fator de potência.

O software também implementa o protocolo de comunicação MQTT para se comunicar com o microcomputador Raspberry Pi, podendo enviar e receber os dados através do Broker. O formato de dados usado na comunicação entre a ESP32 e a Raspberry Pi é o JSON, por ser um formato de envio de informações leve, prático e de fácil interpretação.

4.2.1 BIBLIOTECAS

A IDE Atom possui o sistema PlataformIO (PLATAFORMIO, 2018) definido como um ecossistema aberto de IoT que facilita a programação de dispositivos IoT. O PlataformIO reuni a maioria das bibliotecas de desenvolvimento envolvendo IoT em sua base de dados facilitando assim que as bibliotecas que forem necessárias sejam facilmente encontradas. As bibliotecas extras necessárias para o envio e recebimento das informações usando MQTT e JSON são:

- PubSubCliente.h: necessária para fazer uma publicação/assinatura de mensagens com um servidor (*Broker*) que usa protocolo de comunicação MQTT (OLEARY, acesso em 10 dez. 2018). O servidor em questão é a Raspberry Pi, configurada para tal.
- ArduinoJSON.h: biblioteca para auxiliar na produção de *strings* com conteúdo JSON para envio e recebimento de dados (HUMFREY, 2018).

- WiFiServer.h: tem como funcionalidade criar um servidor Web no dispositivo programado caso haja necessidade de expor os dados para usuário (ARDUINO, acesso em 10 dez. 2018).

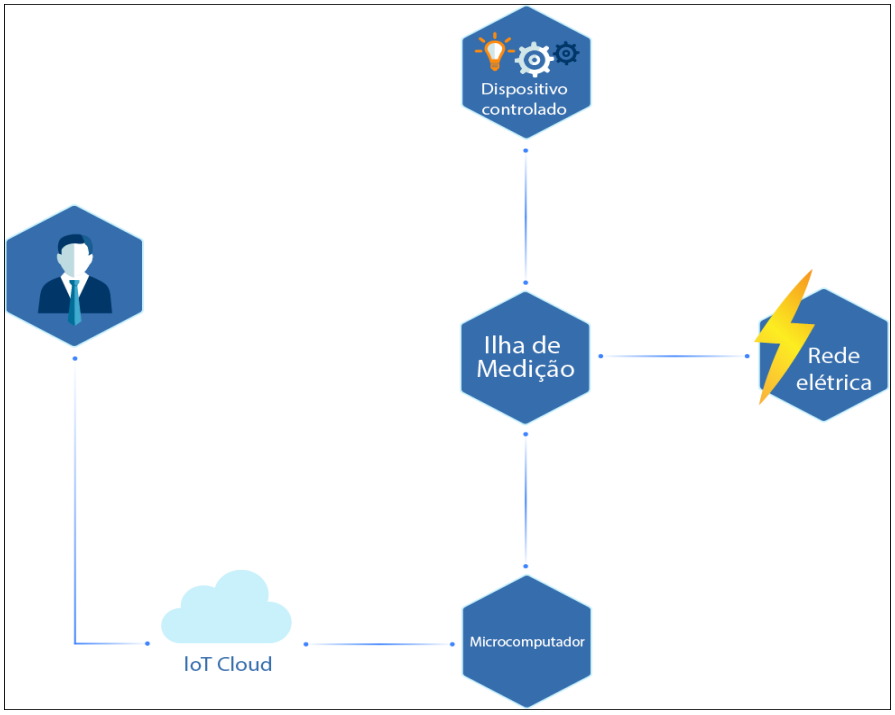
Outras bibliotecas não descritas aqui que foram usadas, são bibliotecas necessárias para o funcionamento da ESP32 que a própria IDE Atom configura por padrão de acordo com o microcontrolador e linguagem escolhidos.

4.3 SISTEMA

Na Figura 2 segue a representação de como é o sistema final implementado. Representada na Figura 2, a ilha de medição de grandezas elétricas tem um microcontrolador ESP32 que é responsável por controlar os dispositivos conectados a ela, interpretar os dados elétricos que estiverem passando pelo circuito quando algum dispositivo conectado a ilha estiver em funcionamento, tratar esses dados e gerar informações de interesse do usuário, enviar essas informações para o microcomputador Raspberry Pi, que por sua vez, repassa essas informações para o usuário podendo usar um servidor intermediário IoT *Cloud.*

Ainda na Figura 2, o usuário faz o processo inverso, enviando dados de acionamento de dispositivo para a ilha de medição usando o microcomputador diretamente ou o servidor intermediário.

Figura 2 - Sistema integrado proposto pelo projeto



Fonte: Autor (2018).

5 DESENVOLVIMENTO

As informações geradas para o usuários são baseadas nos dados captados dos canais ADC's. Dito isso é necessário captar os valores desses canais e processar esses dados usando as funções próprias da linguagem Arduino. Para saber o que acontece no canal ADC do microcontrolador, basta utilizar a função analogRead() passando como parâmetro o canal que será lido. No projeto, foi escolhido os canais 32 e 34 da placa Wemos Lolin ESP32. A função analogRead() tem um retorno baseado na precisão de leitura do canal ADC da respectiva placa. Exemplo:

- analogRead(32).
- analogRead(34).

Para utilizar o canal digital é usado uma outra função específica da linguagem Arduino, mas antes é preciso definir qual canal será utilizado com a função pinMode() e em seguida repassar a informação para o canal digital utilizando a função digitalWrite(), se ele for de saída, ou ler a informação utilizando digitalRead(), se ele for de entrada.

Para armazenar os valores captados, foi necessário criar algumas coleções de variáveis que podem ser acessadas usando índices chamadas de vetor. Começando pelos vetores de tensão e corrente incialmente de tamanho e de tipo *float*, basta usar um laço de repetição que percorre os vetores de tensão e corrente lendo os dados que estão passando pelo circuito de condicionamento de sinais através dos canais ADC e preenchendo os dois vetores respectivamente.

Sabe-se que a frequência da rede elétrica no Brasil é de 60Hz, portanto um período de onda demora em média 16,667ms para se formar. Os dados de tensão e corrente necessários para realizar os cálculos correspondem ao período de onda, com isso então é necessário criar os vetores de tensão e corrente com tamanhos baseados na quantidade de valores que a ESP32 consegue captar durante um período de onda.

Para determinar o tamanho exato dos vetores criados foi necessário realizar um teste com o processador da ESP32, usando a função micros() medindo a

proporção de quantos dados o microcontrolador é capaz de ler dos canais ADC's com apenas 16,667ms. Após a medição, verificou-se que a ESP32 consegue ler dos canais ADC's 1667 valores. Separando essa quantidade de valores em dois vetores, os vetores de tensão e corrente podem conter até 833 valores.

Após realizar o laço de repetição novamente preenchendo os dois vetores de tamanho 833 com os valores lidos dos canais ADC's, foi observado que, por alguma razão não especificada pelo fabricante, a composição da memória interna onde ficam armazenadas as estrutura de dados não comportam um vetor com essa quantidade de valores. Ao tentar alocar espaço para construir os dois vetores de 833 valores, o endereçamento das posições de alguns índices dos vetores eram perdidos resultando assim em uma interrupção do sistema interno não permitindo que o microcontrolador fosse iniciado.

Para resolver o problema de indexação, foi necessário reduzir o tamanho dos vetores. Após realizar os testes de redução, diminuindo o tamanho dos vetores até o limite de indexação de endereços que a ESP32 consegue alocar, chegou-se à conclusão que o microcontrolador tinha a capacidade de armazenar vetores de aproximadamente 556 posições sem perder a indexação de valores, possibilitando a execução do código.

Anteriormente foi dito que a ESP32 é capaz de ler 1667 valores com o tempo de 16,667ms. Como o tamanho dos vetores foi definido em 556 valores, isso daria apenas 1112 valores, que corresponde a aproximadamente 11,118ms. De acordo com o tempo encontrado para preencher os dois vetores de 556 valores é notável que há uma diferença de 5,549ms. Graças a essa diferença, se tornou necessário complementar o tempo de execução de 11,118ms para tornar a leitura de dados dos canais ADC's mais precisa, inserindo uma instrução extra, como observado na linha 4 da Figura 3, fazendo com que uma variável sem utilidade seja preenchida com uma informação lida de um canal ADC qualquer. A instrução adicionada faz com que a ESP32 consuma o tempo necessário para complementar o tempo de 11,118ms até o tempo necessário de 16,667ms, formando assim o tempo exato de um período de onda.

A partir desta etapa, foi possível definir um tamanho fixo de 556 valores para os dois vetores, ou seja, um período de onda completo será representado com 556 valores. A Figura 3 representa o processo de preenchimento dos vetores de tensão

e corrente, fazendo a coleta das amostras necessárias para formar o período de onda no tempo certo.

Figura 3 - Leitura dos canais ADC's e armazenamento nas estruturas

Algoritmo para leitura dos canais ADC

```
1. para n ← 0 ate periodo de onda completo faça
2.        vetorTensao[n] ← analodRead(canal ADC 1);
3.        vetorCorrente[n] ← analogRead(canal ADC 2);
4.        variavelSemUtilidade ← analogRead(canal ADC qualquer);
5. fim
```

Fonte: Autor (2018).

Após ler os canais ADC's e preencher os vetores, é preciso converter esses dados para que seja possível interpretá-los, isso por que os canais ADC's leem os valores que passam no circuito, porém, são capazes de representar apenas valores entre 0 ~ 4095 de acordo com sua precisão. A fórmula de conversão de valores é dada abaixo:

$$Valor = \frac{\left(Valor\ lido * (voltagem\ de\ operação)\right)}{precisão\ canal\ ADC}$$

Onde:

- Valor lido = valor lido no canal ADC.
- Voltagem de operação = 3.3 Volts de acordo com a especificação do microcontrolador
- Precisão do canal ADC = 12 bits = 2^{12} = 4.096 (0 ~ 4095).

Então como mostra a Figura 4, basta usar o laço de repetição para percorrer os dois vetores novamente, mas agora, convertendo os valores lidos usando a fórmula acima.

Figura 4 – Pseudocódigo de conversão de valores lidos dos canais ADC's

Algoritmo para conversão dos valores lidos dos canais ADC's

1. $\boldsymbol{para}$ n $\leftarrow 0$ $\boldsymbol{ate}$ $periodo\ de\ onda\ completo$ $\boldsymbol{faça}$
2. $\quad vetorTensao[n] \leftarrow (vetorTensao[n] * voltagem\ de\ operacao\ microcontrolador)/precisão\ do\ canal\ ADC;$
3. $\quad vetorCorrente[n] \leftarrow (vetorCorrente[n] * voltagem\ de\ operacao\ microcontrolador)/precisão\ do\ canal\ ADC;$
4. $\boldsymbol{fim}$

Fonte: Autor (2018).

Após a conversão, tem-se os vetores de tensão e corrente criados e convertidos. Mas isso ainda não é suficiente para se ter valores reais. Além dessa conversão é necessário também ajustar o valor convertido para o valor real da grandeza (corrente e tensão alternados). Até aqui, apenas trabalhou-se com valores positivos que variam de 0 a 3.3 Volts.

As grandezas que o circuito é capaz de medir possuem valores reais, que podem ser estimados usando algum medidor como voltímetro ou multímetro. Para que os valores lidos dos canais ADC's sejam compatíveis é preciso adicionar constantes que são capazes de transformar os valores em que o circuito trabalha nos valores reais de tensão e corrente que o amperímetro é capaz de medir. A fórmula de ajuste de valores para a tensão é dada como:

$$Valor = (valor\ lido - voltagemOperacao) * constante\ de\ operação\ da\ tensão$$

Onde:

- Valor lido = valor lido no canal ADC.
- Voltagem operação = 3,3 Volts / 2 = 1,65 V.
- Constante de operação da tensão = 142.7 V/V.

A correção a cima permite que uma grandeza alternada seja compatibilizada com a entrada do ADC além de compensar o ganho do circuito de condicionamento de sinais. De forma análoga, o mesmo ajuste é aplicado aos valores de corrente:

$$Valor = (valor\ lido - \ voltagemOperacao) * constante\ de\ operação\ da\ corrente$$

Onde:

- Valor lido = valor lido no canal ADC.
- Voltagem operação = 3,3 Volts / 2 = 1,65 V.
- Constante de operação da corrente = 8 A/V.

Sem aplicar os ajudes de valores, os vetores de tensão e corrente possuem apenas os valores entrada representados como 0 a 3,3 Volts. Portanto, esses valores ainda não são reais, pois a intenção é trabalhar com valores de tensão que estão entre -180 a +180v e de corrente que estão entre -13,2 A +13,2 A.

A Figura 5 representa a aplicação da fórmula acima nos vetores de tensão e corrente, os transformando assim em vetores com valores reais da grandeza que vão possibilitar a partir de agora a realização do cálculos de medição.

Figura 5 - Pseudocódigo para conversão e ajuste de valores

Algoritmo para ajuste da não-linearidade

1. $\boldsymbol{para}$ n $\leftarrow 0\ \boldsymbol{ate}\ periodo\ de\ onda\ completo\ \boldsymbol{faça}$
2. $\quad vetorTensao[n] \leftarrow \big((vetorTensao[n] - 1{,}65) * 142{,}7\big);$
3. $\quad vetorCorrente[n] \leftarrow \big((vetorCorrente[n] - 1{,}65) * 8\big);$
4. $\boldsymbol{fim}$

Fonte: Autor (2018).

Os cálculos de potência e valores eficazes representarão para o usuário informações sobre consumo e sobre uso dos dispositivos conectados à ilha. A potência instantânea pode ser definida como a quantidade de energia gasta em um período de tempo infinitesimal. Dada a fórmula:

$$Potência\ Instantânea(t) = \ U(t).I(t)$$

Onde U(t) e I(t) são a tensão e corrente respectivamente em um dado instante t. Com um vetor de tensão e um vetor de corrente, precisa-se então de um

vetor auxiliar para calcular a potência instantânea. Representado na Figura 6, após criar o vetor auxiliar de potência instantânea, basta preencher esse vetor, percorrendo os vetores de tensão e corrente aplicando a fórmula acima.

Figura 6 - Pseudocódigo do cálculo de potência instantânea

Algoritmo para o cálculo da potência instantânea

1. $\boldsymbol{para}$ n $\leftarrow 0$ $\boldsymbol{ate}$ $periodo\ de\ onda\ completo$ $\boldsymbol{faça}$
2. $\quad potenciaInstantanea[n] \leftarrow vetorTensao[n] * vetorCorrente[n];$
3. $\boldsymbol{fim}$

Fonte: Autor (2018).

Com o vetor de potência instantânea preenchido, agora basta calcular a potência instantânea acumulada. Como mostra a Figura 7, a potência instantânea acumulada representa a soma de todos os valores de potência instantânea calculados na Figura 6.

Figura 7 - Pseudocódigo do cálculo da potência instantânea acumulada

Algoritmo para o cálculo da potência instantânea acumulada

1. $\boldsymbol{para}$ n $\leftarrow 0$ $\boldsymbol{ate}$ $periodo\ de\ onda\ completo$ $\boldsymbol{faça}$
2. $\quad potenciaInstantaneaAcumulada \leftarrow$
   $\quad potenciaInstantaneaAcumulada + potenciaInstantanea[n];$
3. $\boldsymbol{fim}$

Fonte: Autor (2018).

A potência média de um dispositivo conectado a placa é dada pela seguinte fórmula:

$$Potência\ Média = \frac{1}{n}\sum_{i=1}^{n} Pi$$

Onde n representa a quantidade de amostras ou tamanho do vetor. Visto que a potência instantânea acumulada já foi calculada, basta replicar a equação abaixo no código.

$$potenciaMedia = \frac{potenciaInstantaneaAcomulada}{tamanho\ do\ vetor}$$

O valor eficaz pode ser calculado usando a seguinte fórmula:

$$Valor\ Eficaz = \sqrt{\frac{x^2 1 + x^2 2 + \cdots + x^2 n}{n}}$$

Onde x é o valor de tensão ou corrente e n representa o número de amostras ao longo do período de onda completo. Como pode ser observado na Figura 8, para calcular o valor eficaz de tensão, basta realizar a soma dos valores do vetor de tensão em uma variável auxiliar, dividir a soma desses valores pelo período de onda completo e então calcular a raiz quadrada deste valor.

Figura 8 - Pseudocódigo do cálculo do valor eficaz da tensão

Algoritmo para o cálculo da valor eficaz tensão

1. $valorEficazTensao \leftarrow 0;$
2. $\boldsymbol{para}$ n $\leftarrow 0$ $\boldsymbol{ate}$ $periodo\ de\ onda\ completo$ $\boldsymbol{faça}$
3. $\quad valorEficazTensao \leftarrow valorEficazTensao + (vetorTensao[n] * vetorTensao[n]);$
4. $\boldsymbol{fim}$
5. $valorEficazTensao \leftarrow srqt(valorEficazTensao/ periodo\ de\ onda\ completo);$

Fonte: Autor (2018).

Com isso, tem-se então o valor eficaz da tensão calculado e armazenado na respectiva variável. O processo representado na Figura 9 é análogo ao processo da Figura 8, porém, com a nova variável auxiliar representando o valor eficaz da corrente e consequentemente usando o vetor de corrente representando os valores a serem somados.

Figura 9 - Pseudocódigo do cálculo do valor eficaz da corrente

Algoritmo para o cálculo da valor eficaz corrente

1. $valorEficazCorrente \leftarrow 0;$
2. $\boldsymbol{para}$ n $\leftarrow 0$ $\boldsymbol{ate}$ $periodo\ de\ onda\ completo$ $\boldsymbol{faça}$
3. $\quad valorEficazCorrente \leftarrow valorEficazCorrente + (vetorCorrente[n] * vetorCorrente[n]);$
4. $\boldsymbol{fim}$
5. $valorEficazCorrente \leftarrow srqt(valorEficazCorrente/ periodo\ de\ onda\ completo);$

Fonte: Autor (2018).

Com os valores eficazes de tensão e corrente calculados, agora finalmente é possível calcular o fator de potência. O fator de potência é calculado a partir da seguinte equação:

$$Fator\ de\ Potência = \frac{Potência\ Média}{Valor\ Eficaz\ de\ Tensão * Valor\ Eficaz\ Corrente}$$

Logo, aplicando a fórmula acima no código tem-se:

$$fatorPotencia \leftarrow potenciaMedia/(valorEficazTensao * valorEficazCorrente);$$

Com os vetores de tensão e corrente preenchidos é possível expor os dados para que sejam reproduzidos em tabelas e até construção de gráficos. Os valores calculados agora são informações apresentáveis ao usuário, são eles: potência média, valor eficaz tensão, valor eficaz da corrente e fator de potência.

Para maior precisão nos dados obtidos foi calculada uma média móvel entre os valores disponíveis. Representada na Figura 10, essa média é baseada em um vetor com *n* valores na qual o *loop* interno do programa deve ser executado *n* vezes para que o vetor de média móvel fique totalmente preenchido e assim gerar valores médios de: valor eficaz de tensão, valor eficaz corrente, potência média e fator de potência. Logo após a n-ésima execução do código é possível observar os dados de forma mais precisa.

Figura 10 - Pseudocódigo de cálculo da média móvel

Algoritmo para o cálculo da média móvel

1. $vetorMediaPotenciaMedia[contadorMediaMovel] \leftarrow potenciaMedia;$
2. $vetorMediaValorEficazTensao[contadorMediaMovel] \leftarrow valorEficazTensao;$
3. $vetorMediaValorEficazCorrente[contadorMediaMovel] \leftarrow valorEficazCorrente;$
4. $vetorMediaFatorPotencia[contadorMediaMovel] \leftarrow FatorPotencia;$
5. $mediaPotenciaMedia \leftarrow 0;$
6. $mediaValorEficazCorrente \leftarrow 0;$
7. $mediaValorEficazTensao \leftarrow 0;$
8. $mediaFatorPotencia \leftarrow 0;$
9. $\boldsymbol{para}$ n $\leftarrow 0$ $\boldsymbol{ate}$ $tamanho\ desejado\ do\ vetor\ de\ media\ móvel$ $\boldsymbol{faça}$
10. $\quad mediaPotenciaMedia \leftarrow mediaPotenciaMedia + vetorMediaPotenciaMedia[tamanho\ desejado\ do\ vetor\ de\ media\ movel];$
11. $\quad mediaValorEficazTensao \leftarrow mediaValorEficazTensao + vetorMediaValorEficazTensao[tamanho\ desejado\ do\ vetor\ de\ media\ movel];$
12. $\quad mediaValorEficazCorrente \leftarrow mediaValorEficazCorrente + vetorMediaValorEficazCorrente[tamanho\ desejado\ do\ vetor\ de\ media\ movel];$
13. $\quad mediaFatorPotencia \leftarrow mediaFatorPotencia + vetorMediaFatorPotencia[tamanho\ desejado\ do\ vetor\ de\ media\ movel];$
14. $\boldsymbol{fim}$

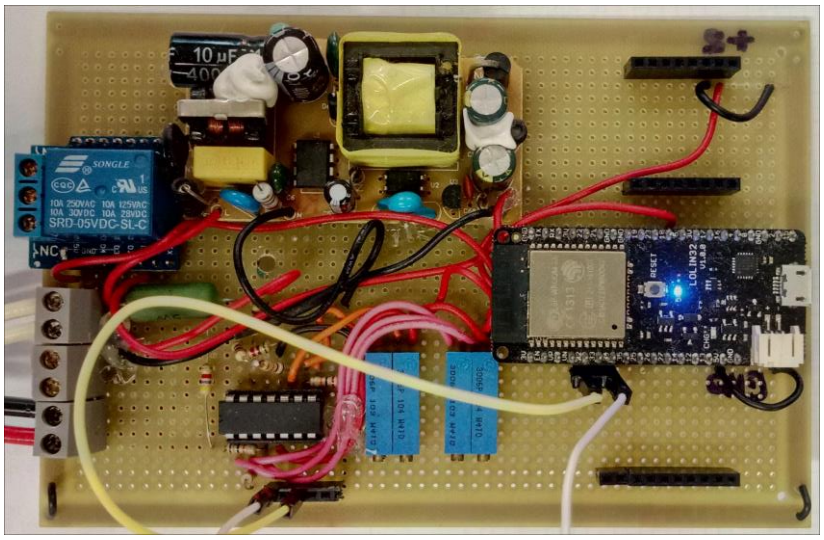
Fonte: Autor (2018).

Obs.: A média móvel usada foi baseada em 5 valores (tamanho do vetor), escolhido para que o algoritmo não demore muito para gerar resultados. Caso o usuário ache necessário, essa média pode ser feita com mais valores, tornando assim mais precisa, porém, consumindo mais tempo de execução.

6 RESULTADOS

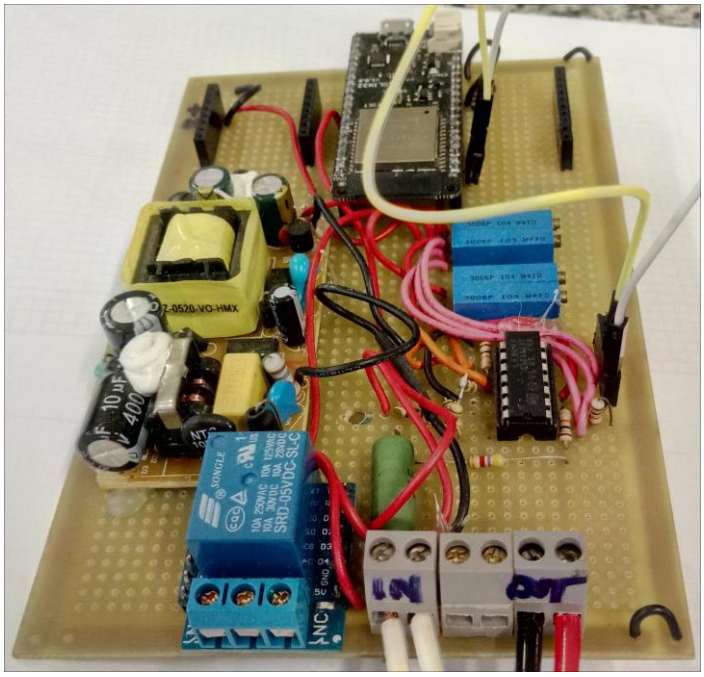
Apresentada nas Figuras 11 e 12, a ilha de medição de grandezas elétricas contém o circuito de condicionamento de sinais, um microcontrolador Wemos Loin ESP32 e o relé, além da fonte de alimentação.

Figura 11 - Ilha de medição de grandezas elétricas



Fonte: Autor (2018).

Figura 12 - Ilha de medição de grandezas elétricas



Fonte: Autor (2018).

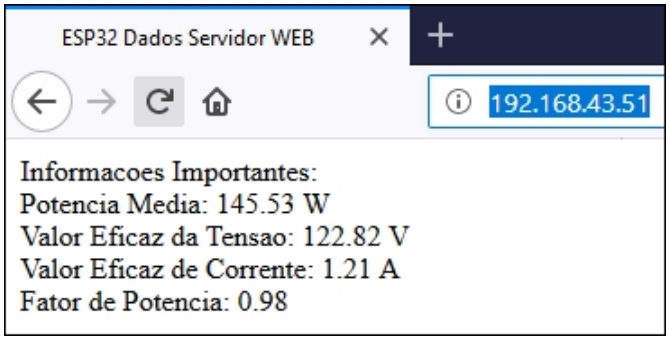
O sistema desenvolvido apresentado na Figura 13, é classificado como um sistema integrado, ou seja, é um sistema que consegue operar utilizando uma central inteligente. Trata-se de um sistema de automação simples, baseado em uma ilha de medição que é observada por um microcomputador Raspberry Pi e controlada por um microcontrolador Wemos Lolin ESP32.

Figura 13 - Sistema completo: ilha de medição, Raspberry Pi, carga elétrica, voltímetro e amperímetro

Fonte: Autor (2018).

Na figura 13 ilha de monitoramento tem como dispositivo conectado a ela uma série de lâmpadas incandescentes ligadas em paralelo. Como mostram as Figuras 14 e 15 o sistema é capaz de reproduzir as informações geradas via servidor Web local.

Figura 14 - Apresentação dos dados de tensão e corrente eficazes, potência média e fator de potência para o conjunto de lâmpadas apresentado na Figura 13



Fonte: Autor (2018).

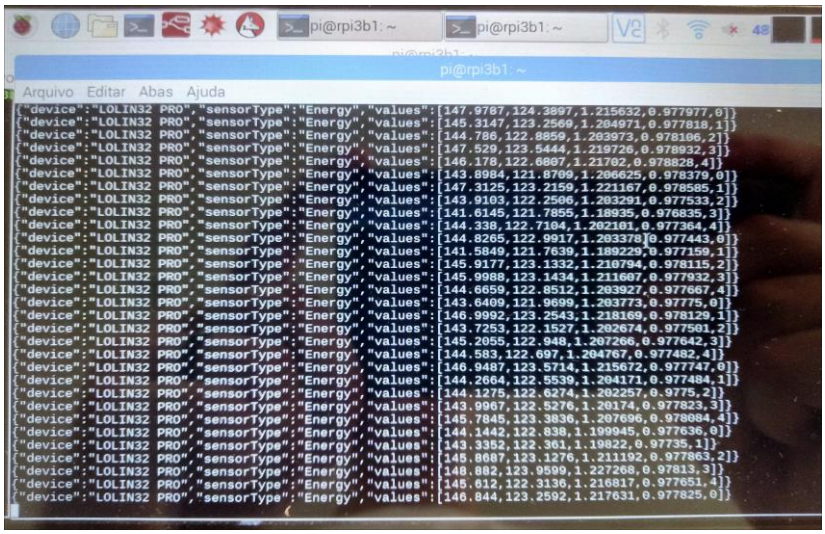
Figura 15 - Amostragem dos dados que representam os vetores de corrente e tensão correspondente a um período de onda de 60 Hz

Vetor de Corrente:	Vetor de tensao:
-1.02 -0.99 -1.02 -0.99 -1.05 -1.00 -0.98 -1.00 -0.97 -0.89 -0.94 -0.96	-139.70 -139.12 -139.46 -136.56 -135.51 -135.04 -134.69 -132.13 -130.25
-0.54 -0.53 -0.52 -0.47 -0.47 -0.49 -0.40 -0.40 -0.41 -0.40 -0.38 -0.33	-103.94 -102.86 -102.74 -99.27 -96.50 -95.90 -93.86 -90.96 -88.79 -85.53 -
0.13 0.11 0.14 0.16 0.21 0.25 0.25 0.34 0.25 0.23 0.34 0.31 0.37 0.01	-47.36 -43.34 -42.25 -42.49 -38.22 -36.88 -38.22 -32.01 -30.19 -30.07 -28.
0.89 0.90 0.93 0.93 0.93 0.90 0.97 0.98 1.10 1.01 1.05 1.07 1.11 1.10	13.41 16.28 22.34 25.07 23.76 28.14 30.97 31.79 37.08 38.48 39.18 41.05
1.47 1.47 1.54 1.53 1.52 1.60 1.54 1.60 1.60 1.64 1.66 1.66 1.72 1.73	77.07 79.84 81.27 83.91 85.12 85.78 87.74 90.46 90.89 94.56 94.99 95.85
1.92 1.89 1.95 1.89 1.92 1.91 1.94 1.95 1.92 1.95 1.89 1.94 1.89 1.84	124.46 126.75 128.13 126.65 127.74 130.59 130.89 129.81 132.25 134.20
1.74 1.74 1.82 1.74 1.74 1.73 1.74 1.74 1.68 1.68 1.67 1.68 1.65 1.63	161.22 158.52 162.35 164.83 164.58 160.96 166.78 167.62 167.20 165.85
1.31 1.31 1.28 1.25 1.20 1.21 1.08 1.18 1.13 1.11 1.11 1.10 1.09 1.09	171.44 171.11 171.28 170.87 171.03 172.51 170.12 171.28 171.36 170.87
0.59 0.64 0.46 0.50 0.48 0.46 0.46 0.44 0.42 0.40 0.46 0.35 0.36 0.34	165.34 164.49 164.66 163.64 161.92 162.00 160.35 158.96 157.72 157.46
-0.22 -0.23 -0.20 -0.30 -0.27 -0.29 -0.29 -0.37 -0.29 -0.33 -0.37 -0.40	138.33 136.03 134.29 134.39 133.32 130.49 130.00 129.71 127.24 125.85
-0.81 -0.80 -0.83 -0.83 -0.88 -0.85 -0.83 -0.98 -0.90 -0.98 -0.96 -0.98	100.86 97.78 96.81 96.39 90.89 90.89 89.16 86.43 85.56 83.70 82.38 81.0
-1.17 -1.28 -1.27 -1.27 -1.26 -1.34 -1.33 -1.32 -1.35 -1.30 -1.34 -1.39	40.35 36.96 35.55 32.26 32.85 31.91 27.08 25.42 22.46 22.70 20.56 16.63
-1.37 -1.42 -1.38 -1.40 -1.37 -1.40 -1.39 -1.39 -1.38 -1.31 -1.41 -1.36	-25.93 -29.09 -30.43 -34.69 -36.40 -39.81 -40.30 -43.22 -45.17 -42.49 -45.
-1.33 -1.30 -1.32 -1.25 -1.30 -1.30 -1.30 -1.32 -1.29 -1.27 -1.28 -1.27	-79.72 -82.51 -85.17 -87.46 -87.46 -92.65 -92.53 -96.14 -95.66 -97.95 -100
-1.03 -1.03 -1.01 -1.01 -1.00 -0.97 -0.98 -0.94 -0.96 -0.94 -0.87 -0.91	-126.97 -126.97 -127.79 -124.85 -129.67 -130.72 -130.49 -132.24 -132.94

Fonte: Autor (2018).

O sistema apresentado na Figura 13 também tem a função enviar os dados via MQTT em formato JSON para a Raspberry Pi como mostra a Figura 16, que por sua vez, pode realizar comunicação com outros servidores a escolha do usuário. Caso o cliente envie algum comando usando um servidor, o sistema deve ser capaz de interagir com a ilha de medição através do protocolo MQTT. O servidor que for escolhido, por sua vez, poderá ter a função de exibir os dados gerados pela ilha em forma de gráficos interativos em tempo real e controlar (ligar/desligar) qualquer dispositivo conectado à ilha de medição.

Figura 16 - Valores recebidos via MQTT em formato JSON apresentados no terminal da Raspberry Pi em tempo real



Fonte: Autor (2018).

A interpretação dos valores mostrados na Imagem 6 são uma sequência de dados recebidos via MQTT em formato JSON que significam na respectiva ordem: {Média móvel da potência média, Média móvel do valor eficaz tensão, Média móvel do valor eficaz corrente, Média móvel fator de potência, variável de depuração}.

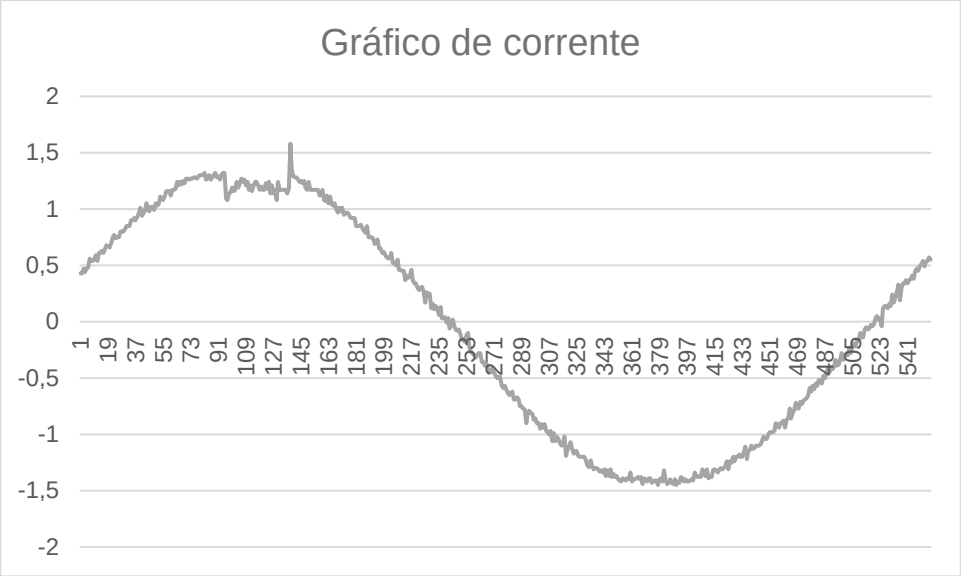
Na primeira bateria de testes, foi possível gerar os gráficos *offline* com as senoides a partir da exibição dos vetores de tensão e corrente no servidor Web da ESP32. Foi observado, como mostra o Gráfico 1 que o canal ADC da ESP32 acaba gerando valores não-lineares. O fabricante da ESP32 Espressif divulgou no manual e documentação (EXPRESSIF SYSTEMS, 2018, p. 22) que os chips da ESP32 podem apresentar uma diferença de mais ou menos 6% de um chip para outro nos resultados medidos usando os canais ADC's. Além disso, a sua conversão não possui uma resposta linear por toda faixa disponível e sugere que os usuários implementem outros métodos além do citado no *datasheet* (EXPRESSI, 2018) caso julguem necessários.

Gráfico 1 - Achatamento da senoide de tensão no semiciclo positivo apresentando a não-linearidade do canal ADC

Fonte: Autor (2018).

Verifica-se no Gráfico 2 que a não-linearidade não se expressa no sinal de corrente uma vez que para os valores de corrente na lâmpada, o valor real na entrada do ADC não se aproxima da tensão limite superior (3,3 Volts).

Gráfico 2 - Senoide de corrente

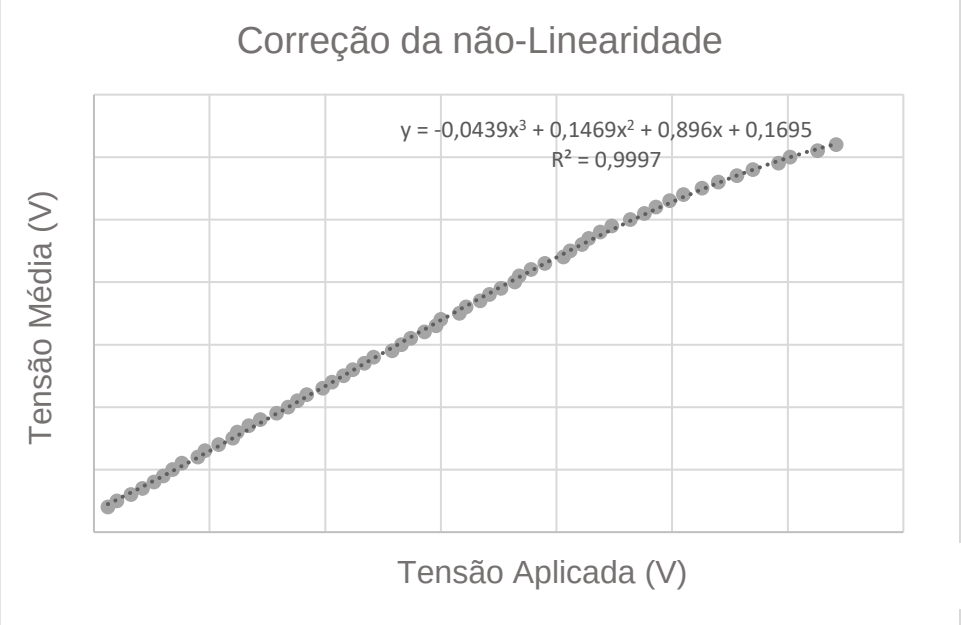


Fonte: Autor (2018).

Para a correção da não linearidade foi preciso estudar o comportamento da função e ajustar seus valores a partir de um polinômio de terceiro grau. Como mostra o Gráfico 3, o R^2 que é chamado de coeficiente de determinação é uma medida de ajustamento de um modelo estatístico linear generalizado, como a regressão linear. O R^2 varia entre 0 e 1, indicando, em percentagem, o quanto o modelo consegue explicar os valores observados. Quanto maior o R^2, mais explicativo é o modelo e melhor ele se ajusta à amostra. Usando a ferramenta Microsoft Excel[4] para gerar o Gráfico 3, foi possível encontrar o valor de R^2 a partir de um polinômio de terceiro grau. A ferramenta retorna a equação que melhor representa o modelo.

[4] Disponível em: <https://products.office.com/pt-br/excel>

Gráfico 3 - Ajuste da não-linearidade a partir de um polinômio de terceiro grau



Fonte: Autor (2018).

A equação gerada pelo software Microsoft Excel, foi aplicada aos vetores de tensão e corrente para realizar a correção da não-linearidade do canal ADC como mostra a Figura 17.

Figura 17 - Pseudocódigo para correção da não-linearidade

Algoritmo para ajuste da não-linearidade

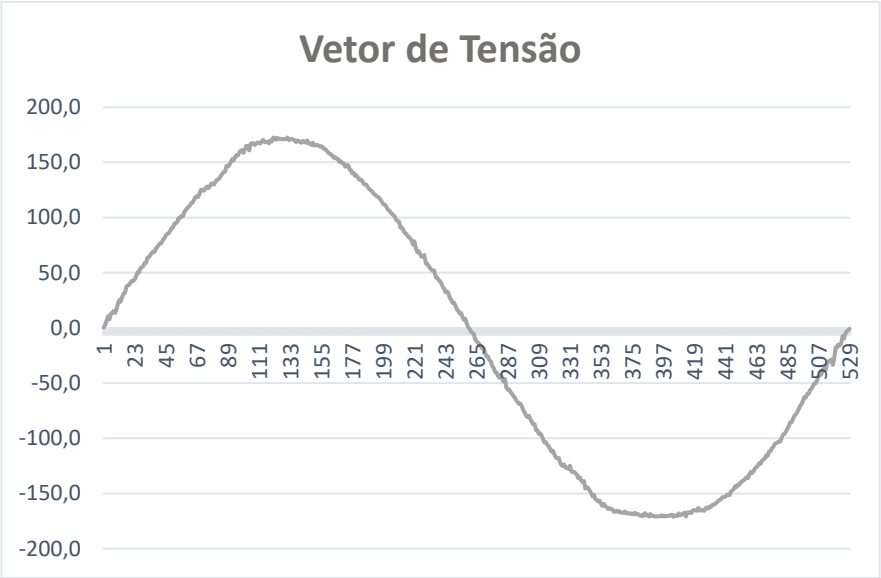
1. $\boldsymbol{para}$ n $\leftarrow 0$ $\boldsymbol{ate}$ $periodo\ de\ onda\ completo$ $\boldsymbol{faça}$
2. $vetorTensao[n] \leftarrow \left(-0.0439 * (vetorTensao[n] * vetorTensao[n] * vetorTensao[n])\right) + \left(0.1469 * (vetorTensao[n] * vetorTensao[n])\right) + (0.896 * vetorTensao[n]) + 0.1695;$
3. $vetorCorrente[n] \leftarrow \left(-0.0439 * (vetorCorrente[n] * vetorCorrente[n] * vetorCorrente[n])\right) + \left(0.1469 * (vetorCorrente[n] * vetorCorrente[n])\right) + (0.896 * vetorCorrente[n]) + 0.1695;$
4. $\boldsymbol{fim}$

Fonte: Autor (2018).

Com a correção da não linearidade do canal ADC da ESP32, foi possível gerar os gráficos de tensão e corrente mais precisos. Pode-se observar no Gráfico 4

a correção da não-linearidade no primeiro semiciclo onde o achatamento que havia antes foi removido.

Gráfico 4 - Gráfico de tensão após a correção da não linearidade



Fonte: Autor (2018).

Analisando os dois gráficos (Gráficos 1 e 4), foi possível perceber que o problema da não linearidade foi resolvido, permitindo assim que os valores gerados com os cálculos pudessem ser apresentados com maior correção.

Para a situação ilustrada na Figura 14 e analisando os dados gerados pelo sistema e publicados no servidor Web da ESP32 (Figura) tem-se:

- Potência média: **145,43 W**
- Valor Eficaz da Tensão: **122.82 V_{rms}**
- Valor Eficaz da Corrente: **1.21 I_{rms}**
- Fator de Potência: **0.98 FP**

Para comprovar a validade dos dados, foram feitas medições usando um amperímetro e um voltímetro de tal forma que pode-se garantir que os valores encontrados e calculados pelo sistema fossem coerente com os valores reais medidos pelos aparelhos.

O fator de potência contabilizado de 0.98 FP representa que a energia utilizada está sendo 98% toda consumida. O dispositivo ligado ao circuito para os cálculos de consumo foram algumas lâmpadas incandescentes ligadas em paralelo representadas na Figura 18.

Figura 18 - Lâmpadas incandescentes ligadas em paralelo

Fonte: Autor (2018).

O fator de potência produzido por uma lâmpada incandescente é sempre 1, pois a lâmpada é classificada como resistiva. Comparando com o valor encontrado de 0.98 pode-se concluir que o sistema funciona e que os cálculos se mostraram adequados.

A potência média encontrada pelo sistema foi de 154,43W. Para verificar que esse valor está coerente deve se usar a fórmula de potência média dada por:

$$Potência\ média = V * I * cos\rho$$

Onde V = Tensão, I = Corrente e $cos\rho$ = fator de potência.

Para encontrar o valor da tensão (V) foi usado um voltímetro, que foi colocado em paralelo com o circuito para realizar a medição no momento em que a lâmpada estava ligada, como mostra a Figura 19.

Figura 19 - Voltímetro ICEL MD-5570 com o valor da tensão no circuito no momento em que os dados foram gerados pela Ilha

Fonte: Autor (2018).

Para encontrar o valor da corrente (I) foi usado um amperímetro, que foi adaptado no circuito de forma que ele também pudesse medir a corrente no momento em que o circuito estivesse ligado, como mostra a Figura 20.

Figura 20 - Amperímetro MINIPA ET3200 AC com o valor da corrente no circuito no momento em que os dados foram gerados pela Ilha

Fonte: Autor (2018).

O $cos\rho$ é o fator de potência da lâmpada incandescente que por padrão é 1. Com os valores adquiridos é possível realizar o cálculo para verificar se os valores coincidem. Tem-se então que:

$$Potência\ média = 128.0 * 1.03 * 1$$

logo,

$$Potência\ média = 131{,}84\ W.$$

Comparando o valor encontrado (131,84W) com o valor gerado pelo circuito (145,43W) é possível ver que o erro é bem pequeno e que pode ser apenas uma questão de ajuste nas constantes de conversão ou ajuste de valores, bem como também a calibração da não linearidade. Eles retratam os reais valores que as lâmpadas estão trabalhando no exato momento em que foram medidas.

7 CONCLUSÃO

O desenvolvimento deste projeto possibilitou a construção de um sistema integrado composto por um microcomputador e uma ilha de medição de grandezas elétricas. A ilha de medição se mostrou eficiente, capaz de medir grandezas elétricas e inclusive controlar um dispositivo conectado a ela, atingindo assim o objetivo proposto inicialmente.

O sistema ainda pode e deve ser mais desenvolvido, inclusive se mostra necessário a realização de testes exaustivos em ambientes diferentes e com equipamentos de medições diferentes. Os testes são essenciais para esse sistema pois ajudam na manutenção das variáveis de apoio e calibrações de dados.

O sistema produzido neste trabalho pode e deve ser expandido, aumentando suas funcionalidades e também ser produzido em pequenas escalas de tal forma que o microcomputador Raspberry Pi passe a controlar não só uma, mas várias ilhas de medições simultaneamente, que por sua vez poderão ter a capacidade de controlar vários dispositivos simultaneamente.

Ao fazer alguns testes e ajustes de variáveis, verificou-se que as partes mais complexas e desgastantes do processo são a necessidade de realizar testes exaustivos para a calibração das variáveis de tal forma que o sistema possa processar e gerar dados mais precisos, além também de ter que fazer repetitivos testes em diferentes ESP32 devido a taxa de erro de leitura de um microcontrolador para o outro.

Dada à importância do assunto, torna-se necessário o desenvolvimento de novas funcionalidades e a expansão das funcionalidades já existentes para que em um futuro próximo seja possível tornar o sistema em questão em um produto comercializável.

REFERÊNCIAS

ALLDATASHEET. **LM324 Datasheet (PDF) - NXP Semiconductors**. [s.d.]. Disponível em: <http://pdf1.alldatasheet.com/datasheet-pdf/view/17871/PHILIPS/LM324.html>. Acesso em: 14 dezembro 2018.

AMAZON. **Amazon Web Service.** [s.d.]. Disponível em: <https://aws.amazon.com/pt/free/>. Acesso em: 09 dezembro 2018.

AQUINO, A. INTERNET DAS COISAS: Nós, as cidades, os robôs, os carros: Tudo conectado!. **Computação Brasil**. Porto Alegre: Sociedade Brasileira de Computação, n. 29, p. 1-58, mar. 2015.

ARDUINO. **ARDUINO UNO REV3**. 2018. Disponível em: <https://store.arduino.cc/usa/arduino-uno-rev3>. Acesso em: 08 dezembro 2018.

ARDUINO. **WiFiServer()**. [s.d.]. Disponível em: <https://www.arduino.cc/en/Reference/WiFiServer> Acesso em: 10 dezembro 2018.

ATOM. **A hackable text editor for the 21st Century**. 2018. Disponível em: <https://atom.io/>. Acesso em: 10 dezembro 2018.

BEAGLEBOARD. **PocketBeagle:** What is PocketBeagle?. 2018. Disponível em: <http://beagleboard.org/pocket>. Acesso em: 08 dezembro 2018.

BLUETOOTH. **About Us**. 2018. Disponível em: <https://www.bluetooth.com/about-us>. Acesso em: 09 dezembro 2018.

DELAI, A.L. **Sistemas Embarcados: a computação invisível**. Guia do Hardware, 2013. Disponível em: <http://www.hardware.com.br/artigos/sistemas-embarcados-computacao-invisivel/>. Acesso em: 2 dezembro de 2018.

EGHAM, U.K. **Gartner Says 8.4 Billion Connected "Things" Will Be in Use in 2017, Up 31 Percent From 2016**. Gartner, 2017. Disponível em: < https://www.gartner.com/newsroom/id/3598917>. Acesso em: 2 dezembro 2018.

ESPRESSIF SYSTEMS. **ESP8266EX**: Datasheet. 2018. Disponível em: <https://www.espressif.com/sites/default/files/documentation/0a-esp8266ex_datasheet_en.pdf>. Acesso em: 07 dezembro 2018.

EXPRESSIF SYSTEMS. **ESP32 Series:** Datasheet. 2018. Disponível em: <https://www.espressif.com/sites/default/files/documentation/esp32_datasheet_en.pdf>. Acesso em: 10 dezembro 2018.

FIREWARE LAB. **Apresentação do protocolo MQTT como alternativa para comunicação IoT**. 2016. Disponível em: <http://fiwarelabsp.org/2016/07/apresentacao-do-protocolo-mqtt/>. Acesso em: 09 dezembro 2018.

HUMFREY, N. **ArduinoJson**. 2018. Disponível em: <https://www.arduinolibraries.info/libraries/arduino-json> Acesso em: 10 dezembro 2018.

IBM. **IBM Watson**. [s.d.]. Disponível em: <https://www.ibm.com/watson/index.html>. Acesso em: 09 dezembro 2018.

ITEAD. **eWeLink:** Internet of Everything. 2017. Disponível em: <https://sonoff.itead.cc/en/ewelink>. Acesso em: 09 dezembro 2018.

ITEAD. **Sonoff Pow WiFi Switch With Power Consumption Measurement.** 2017. Disponível em: <https://www.itead.cc/sonoff-pow.html>. Acesso em: 09 dezembro 2018.

MATTENRN, F.; FLOERKEMEIER, C. **From the Internet of Computers to the Internet of Things**. In: Lecture Notes in Computer Science. [S.I.: s.n., 2010].

MICROSOFT. **Microsoft Azure.** [s.d.]. Disponível em: <https://azure.microsoft.com/pt-br/overview/what-is-azure/>. Acesso em: 09 dezembro 2018.

MQTT.ORG. **What is MQTT?**. [s.d.]. Disponível em: <http://mqtt.org/faq/>. Acesso em: 09 dezembro 2018.

MURATORI, J.R.; DAL BÓ, P.H. **Automação Residencial: histórico, definições e conceitos**. Atitude Editorial. [S.I.], 2011. cap. 1, p. 70-77.

NFC FORUM. **About the Technology:** NFC and Contactless Technologies. [s.d.]. Disponível em: <https://nfc-forum.org/what-is-nfc/about-the-technology/> Acesso em: 09 dezembro 2018.

OLEARY, N. **Arduino Client for MQTT.** [s.d.]. Disponível em: <https://pubsubclient.knolleary.net/> Acesso em: 10 dezembro 2018.

PANTH, S.; JIVANI, M. **Home Automation System (HAS) using Android for Mobile Phone**. International Journal of Electronics and Computer Science Engineering, Gujarat, India, v. 3, n. 1, p. 1-10, 2013.

PIRAGIBE, L.F. **O que é "Internet das Coisas" e como ela funciona?**. Fluxo Consultoria, 2016. Disponível em: <http://fluxoconsultoria.poli.ufrj.br/blog/tecnologia-informacao/automacao-residencial/>. Acesso em: 2 dezembro 2018.

PLATAFORMIO. **PlataformIO is an open source ecosystem for IoT development**. 2018. Disponível em: < https://platformio.org/> Acesso em: 10 dezembro 2018.

RASPBERRY PI. **Raspberry Pi 2 Model B**. 2015. Disponível em: <https://www.raspberrypi.org/products/raspberry-pi-2-model-b/>. Acesso em: 08 dezembro 2018.

ROUSE, M. **Google Glass**. IoT Agenda, Tech Target, 2013. Disponível em: <https://internetofthingsagenda.techtarget.com/definition/Google-Glass>. Acesso em: 2 dezembro 2018.

ROUSE, M. **IoT Cloud (Salesforce IoT Cloud)**. Search Salesforce, Tech Target, 2016. Disponível em: <https://searchsalesforce.techtarget.com/definition/IoT-Cloud>. Acesso em: 2 dezembro 2018.

RUBIM, J. **Microcontrolador Arduino no ensino de Física**: Proposta e aplicação de uma situação de aprendizagem sobre o tema Luz e Cor. 2014. Dissertação (Mestrado em Ensino de Ciências Exatas). Programa de Pós-Graduação em Ensino de Ciências Exatas, Universidade Federal de São Carlos, São Carlos, 2014.

WEISER, Mark. **The computer for the 21st century**. SIGMOBILE Mob. Comput. Commun. [S.I.], 1999.

WEISER, Mark. **Ubiquitous Computing**. Computer, HOT TOPICS, IEEE Computer Society. [S.I.], 1993.

WEMOS Electronics. **D1 mini**: A mini WiFi board with 4MB flash based on ESP8266EX. 2018. Disponível em: <https://wiki.wemos.cc/products:d1:d1_mini>. Acesso em: 07 dezembro 2018.

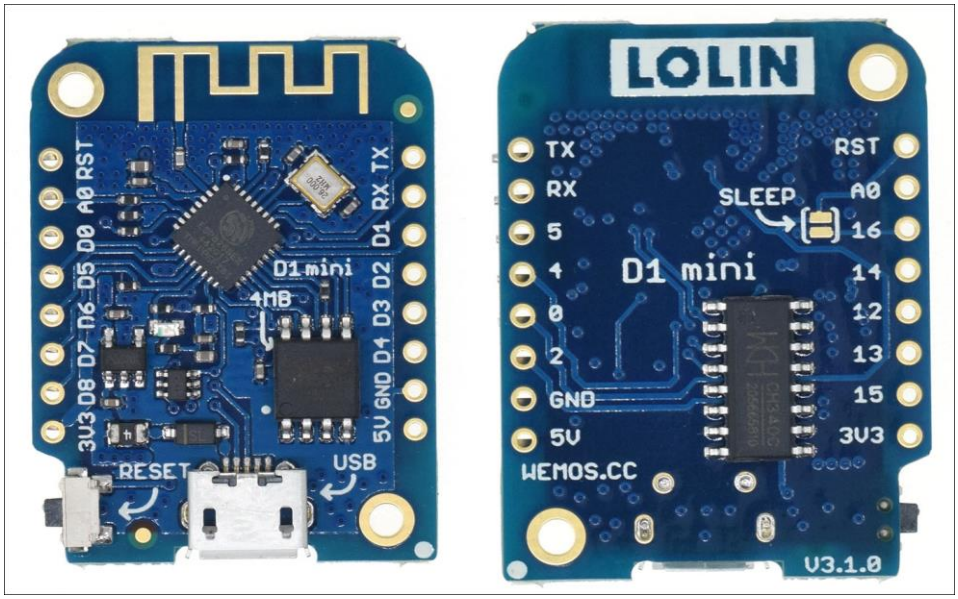
WEMOS Electronics. **LOLIN32**: A WiFi&*Bluetooth* board based ESP-32. 2018. Disponível em: <https://wiki.wemos.cc/products:lolin32:lolin32>. Acesso em: 07 dezembro 2018.

ANEXOS

ANEXO A – FIGURA DO MICROCONTROLADOR WEMOS ESP8266 D1 MINI

Abaixo segue uma imagem do microcontrolador Wemos ESP8266 D1 mini, um dos hardwares principais no mundo da IoT.

Figura 21 - Fotografia de uma Wemos ESP8266 D1 mini



Fonte: Wemos (2018).

ANEXO B – QUADRO DE ESPECIFICAÇÕES WEMOS ESP8266 D1 MINI

Abaixo segue o quadro de especificações do microcontrolador Wemos ESP8266 D1 mini, um dos hardwares principais no mundo da IoT.

Quadro 1 - Especificações Wemos ESP8266 D1 mini

Especificações do equipamento	
Voltagem de operação	3,3v
Entrada/Saída digital	11
Entradas Analógicas (ADC)	1
Clock (Max)	160 Mhz
Memória Flash	4MB
Comprimento	34,2mm
Largura	25,6mm
Peso	3g
WiFi	Sim

Fonte: Wemos (2018).

ANEXO C – FIGURA DO MICROCONTROLADOR WEMOS LOLIN ESP32

Abaixo segue uma imagem do microcontrolador Wemos Lolin ESP32, a evolução da Wemos ESP8266 D1 mini.

Figura 22 - Wemos Lolin ESP32

Fonte: WEMOS Electronics (2018).

ANEXO D – QUADRO DE ESPECIFICAÇÕES DO MICROCONTROLADOR WEMOS LOLIN ESP32

Abaixo segue o quadro de especificações do microcontrolador Wemos Lolin ESP32, a evolução da Wemos ESP8266 D1 mini.

Quadro 2 - Especificações da Wemos Lolin ESP32

Especificações do equipamento	
Voltagem de operação	3,3v
Entrada/Saída digital	26
Entradas Analógicas (ADC)	12
Clock (Max)	240 Mhz
Memória Flash	4MB
Comprimento	58mm
Largura	24,4mm
Peso	5,8g
WiFi	Sim
Bluetooth	Sim

Fonte: WEMOS Electronics (2018).

ANEXO E – FIGURA DO MICROCOMPUTADOR RASPBERRY PI 2B

Abaixo segue uma imagem do microcomputador Raspberry Pi 2B.

Figura 23 - Fotografia de uma RaspBerry Pi 2 Modelo B

Fonte: Raspberry Pi (2015).

ANEXO F – QUADRO DE ESPECIFICAÇÕES DO MICROCOMPUTADOR RASPBERRY PI 2B

Abaixo segue o quadro de especificações do microcomputador Raspberry Pi 2B.

Quadro 3 - Especificações da Raspberry Pi 2B

Especificações do equipamento	
CPU	900 MHz quad-core ARM Cortex A7
GPU	250 MHz Broadcom VideoCore IV
Memória	1 GB (Compartilhado com GPU)
Portas USB	4
Saída de vídeo	HDMI
Entrada/Saída de áudio	Sim
Armazenamento	MicroSD
Rede	10/100Mbps Ethernet
Entrada/Saída digital	17
Voltagem de operação	5V (Via USB)
Potência nominal	800 mA (4,0 W)
Peso	45g
Cumprimento	85,6mm
Largura	56,5mm
Espessura	17mm

Fonte: Raspberry Pi (2015).