



**UNIVERSIDADE ESTADUAL PAULISTA
“JÚLIO DE MESQUITA FILHO”
Campus de Bauru**

IRENO GONÇALVES MACHADO JUNIOR

**Estudo da Colheita de Energia para Medição de Vazão em
Aplicações de Internet das Coisas**

Bauru

2020

IRENO GONÇALVES MACHADO JUNIOR

Estudo da Colheita de Energia para Medição de Vazão em Aplicações de Internet das Coisas

Dissertação apresentada ao Programa de Pós-Graduação em Engenharia Elétrica da Faculdade de Engenharia Elétrica da Universidade Estadual Paulista “Júlio de Mesquita Filho” para a obtenção do título de Mestre em Engenharia Elétrica.

Área de Concentração: Automação

Linha de Pesquisa: Mecatrônica

Orientador: Prof. Dr. Eduardo Paciência Godoy

Bauru – SP

2020

M149e Machado Junior, Ireno Gonçalves
Estudo da Colheita de Energia para Medição de Vazão em
Aplicações de Internet das Coisas / Ireno Gonçalves Machado Junior.
-- Bauru, 2020
95 f. : il., tabs., fotos
Dissertação (mestrado) - Universidade Estadual Paulista (Unesp),
Faculdade de Engenharia, Bauru
Orientador: Eduardo Paciência Godoy
1. Engenharia Elétrica. 2. Energia. 3. Internet das coisas. I. Título.

Sistema de geração automática de fichas catalográficas da Unesp. Biblioteca da Faculdade de
Engenharia, Bauru. Dados fornecidos pelo autor(a).

Essa ficha não pode ser modificada.

ATA DA DEFESA PÚBLICA DA DISSERTAÇÃO DE MESTRADO DE IRENO GONÇALVES MACHADO JÚNIOR, DISCENTE DO PROGRAMA DE PÓS-GRADUAÇÃO EM ENGENHARIA ELÉTRICA, DA FACULDADE DE ENGENHARIA - CÂMPUS DE BAURU.

Aos 04 dias do mês de março do ano de 2020, às 09:00 horas, no(a) ICTS - Instituto de Ciência e Tecnologia de Sorocaba UNESP, reuniu-se a Comissão Examinadora da Defesa Pública, composta pelos seguintes membros: Prof. Dr. EDUARDO PACIÊNCIA GODOY - Orientador(a) do(a) Departamento de Engenharia de Controle e Automação / Instituto de Ciência e Tecnologia - UNESP - Câmpus de Sorocaba, Prof. Dr. EVERSON MARTINS do(a) Departamento de Engenharia de Controle e Automação / Instituto de Ciência e Tecnologia - UNESP - Câmpus de Sorocaba, Prof. Dr. MARCELLO BELLODI do(a) Engenharia / Universidade Paulista - UNIP, sob a presidência do primeiro, a fim de proceder a arguição pública da DISSERTAÇÃO DE MESTRADO de IRENO GONÇALVES MACHADO JÚNIOR, intitulada **ESTUDO DA COLHEITA DE ENERGIA PARA MEDIÇÃO DE VAZÃO EM APLICAÇÕES DE INTERNET DAS COISAS**. Após a exposição, o discente foi arguido oralmente pelos membros da Comissão Examinadora, tendo recebido o conceito final: APROVADO. Nada mais havendo, foi lavrada a presente ata, que após lida e aprovada, foi assinada pelos membros da Comissão Examinadora.

Prof. Dr. EDUARDO PACIÊNCIA GODOY

Prof. Dr. EVERSON MARTINS

Prof. Dr. MARCELLO BELLODI

AGRADECIMENTOS

Agradeço aos meus familiares pelo suporte durante todo o tempo desta caminhada.

Aos meus colegas da UNESP, em especial ao Jefferson Aparecido Dias e ao Alan Marques Molina, pelo incentivo e suporte para buscar soluções no decorrer dos estudos.

À UNESP como um todo, corpo docente e seus funcionários, pela receptividade, pelas aulas, disposição e principalmente pela oportunidade para que eu voltasse a estudar.

Ao meu orientador, professor doutor Eduardo Paciência Godoy, pelo apoio, discernimento e profissionalismo no dia a dia acadêmico, permitindo que eu aprendesse os caminhos para realização do trabalho.

“Há duas formas para viver a sua vida. Uma é acreditar que não existe milagre. A outra é acreditar que todas as coisas são um milagre.”

Albert Einstein

RESUMO

A colheita de energia emerge como uma das soluções para a alimentação sustentável de sensores e dispositivos de monitoramento em redes sem fio nas aplicações da Internet das Coisas (IoT), permitindo até mesmo prescindir do uso de baterias. Este trabalho descreve o desenvolvimento de uma plataforma de colheita de energia e monitoramento de vazão a partir de redes de distribuição hidráulicas. O sistema experimental montado apresenta um circuito fechado de fluxo variável de água, criado a partir de uma bomba pressurizadora controlável eletronicamente, com sensor de vazão e um micro hidrogerador. A energia elétrica recuperada através do micro hidrogerador é utilizada para alimentar diferentes dispositivos de IoT com comunicação via Wi-Fi dos dados de monitoramento da vazão e volume, assim como, fornece energia para baterias. Os testes operacionais tiveram como objetivo dimensionar a quantidade de energia recuperada, em comparação com a energia consumida pelos dispositivos e bateria conectadas, em função dos valores de vazão obtidos pela plataforma. Nos testes realizados foi possível demonstrar a capacidade da colheita de energia pelo micro hidrogerador, com geração de aproximadamente 1W de energia e sua utilização para a operação contínua de dispositivos em aplicações de IoT. O estudo e implantação de diferentes modos de operação para os dispositivos medidores de vazão possibilitaram a redução do consumo de energia em até 54,4% e maximizaram a eficiência energética da solução, comprovando o potencial de captação de energia como uma alternativa viável para garantir a operação de medidores de vazão autossustentados para aplicações de IoT.

Palavras-chave: *Colheita de Energia, Medição de Vazão, Internet das Coisas, Redes de Sensores Sem Fio.*

ABSTRACT

Energy harvesting emerges as one of the solutions for sustainable power supply of sensors and monitoring devices in wireless networks in the Internet of Things (IoT) applications, even allowing to dispense with the use of batteries. This work describes the development of an energy harvesting and flow monitoring platform from hydraulic distribution networks. The assembled experimental system features a variable water flow closed loop, created from an electronically controllable pressurizing pump with flow sensor and a micro hydrogenerator. The electrical energy recovered through the micro-generator supplies different IoT device models with Wi-Fi communication of the flow and volume monitoring data and also provides power for storage batteries. The operational tests aimed to measure the amount of energy recovered, compared to the energy consumed by the connected devices and battery, according to the flow values obtained by the platform. In the tests carried out it was possible to demonstrate the capacity to harvest energy by the micro-generator, generating approximately 1W of energy and its use for the continuous operation of devices in IoT applications. The study and implementation of different modes of operation for flow meter devices make it possible to reduce energy consumption by up to 54.4% and maximize the energy efficiency of the solution, proving the potential of energy recovery as a viable alternative to guarantee the operation of self-sustaining meters for IoT applications.

Keywords: Energy harvesting, flow monitoring, internet of things, wireless sensor networks

LISTA DE FIGURAS

Figura 1:Taxonomia de fontes de colheita de energia para redes de sensores sem fio.....	16
Figura 2:Fontes de energia e transdutores para a alimentação de nós sensores	18
Figura 3 :Modelo de nó sensor de rede sem fio com dispositivo de colheita de energia.....	19
Figura 4 : Colheita de energia com dispositivos internos à tubulação hidráulica	23
Figura 5 :Colheita de energia com dispositivos externos à tubulação hidráulica.....	24
Figura 6 : Medidor de água predial com coletor de energia acoplado.....	25
Figura 7 : Mini hidrogerador para sistema híbrido de colheita de energia	26
Figura 8 : Mecanismo de recuperação de energia hidráulica para alimentação de sensores de pressão.....	26
Figura 9 : Protótipo de colheita de energia da vazão de água com dispositivo eletromagnético	27
Figura 10 : Esquemático da plataforma de colheita de energia hidráulica e medição de vazão	30
Figura 11 – Chip NodeMCU com módulo ESP8266.....	31
Figura 12: Diagrama esquemático do sensor de fluxo YF-S201.....	32
Figura 13: Minigerador de energia hidráulico.....	33
Figura 14 : Módulo regulador de tensão LM2596.....	34
Figura 15: Módulo conversor de nível lógico 5V-3.3V.....	34
Figura 16 : Diagrama esquemático da aplicação de IoT no projeto proposto.....	35
Figura 17: Visão Geral da Plataforma de Testes	37
Figura 18: Descrição funcional dos componentes da plataforma.....	38
Figura 19: Variáveis de interesse para análise dos resultados.....	40
Figura 20: Diagrama esquemático do circuito de energia.....	40
Figura 21: Diagrama ilustrativo e montagem em protoboard do circuito de energia.....	41
Figura 22: Testes iniciais de comunicação Wi-Fi com o Thinger.io.....	42
Figura 23: Painel do instrumento virtual no software LabVIEW	43
Figura 24: Diagrama de blocos do instrumento virtual no software LabVIEW.....	44
Figura 25: Gráfico da potência gerada, consumida e vazão do sistema	46
Figura 26: Gráfico de tensão terminal e corrente fornecida pela microturbina.....	48
Figura 27: Gráfico da corrente consumida nos picos devido à comunicação.....	48
Figura 28 : Gráfico da potência gerada, consumida e vazão obtido em ensaio experimental..	49
Figura 29: Gráfico da corrente consumida nos picos devido à comunicação.....	50

Figura 30: Dashboard construída na plataforma ThingSpeak: vazão (L/min) e Volume acumulado (L).....	51
Figura 31: Potência gerada e consumida e vazão no sistema para dados a cada 15 s via rede Wi-Fi.....	52
Figura 32: Potência gerada, consumida em (a) aproximadamente 5,34 L/min e (b) aproximadamente 7,46 L/min.....	53
Figura 33 : Potência gerada e consumida no sistema para dados a cada 15s e <i>Modem Sleep</i> ..	54
Figura 34 : Potência gerada, consumida: (a) aproximadamente 5,35 L/min e (b) aproximadamente 7,45 L/min.....	54
Figura 35 : Corrente solicitada pelo módulo NodeMCU, conexão Wi-Fi e baixo consumo <i>Modem Sleep</i>	55
Figura 36 : Potência gerada, consumida pelo NodeMCU, com dados a cada 20 s e conexão à rede Wi-Fi.....	56
Figura 37: Potência gerada, consumida pelo NodeMCU para vazão fixa em (a) 5,33 L/min e (b) 7,47 L/min.....	57
Figura 38: Potência gerada, consumida no sistema para transmissão a cada 20 s e <i>Modem Sleep</i>	57
Figura 39: Potência gerada, consumida pelo NodeMCU com vazão fixa em (a) 5,35 L/min e (b) 7,53 L/min.....	58
Figura 40: Corrente solicitada pelo módulo NodeMCU, conexão Wi-Fi ininterrupta e <i>Modem Sleep</i>	59
Figura 41: Potência gerada, consumida e vazão de fluido, transmissão a cada 60 s e conexão Wi-Fi.....	59
Figura 42: Potência gerada, consumida para vazão fixa em (a) 5.38 L/min e (b) 7.51 L/min.	60
Figura 43: Potência gerada, consumida no sistema, transmissão a cada 60 s e <i>Modem Sleep</i> .	61
Figura 44: Potência gerada, consumida para vazão fixa em (a) 5,38 L/min e (b) 7,60 L/min.	61
Figura 45: Corrente solicitada pelo módulo NodeMCU, conexão Wi-Fi e configuração <i>Modem Sleep</i>	62
Figura 46: Potência gerada, consumida e vazão de fluido para transmissão a cada 30 s, e <i>Modem Sleep</i>	63
Figura 47: Potência gerada, consumida no sistema, transmissão a cada 30 s, <i>Modem Sleep</i> e bateria com carga completa.....	64
Figura 48: Corrente fornecida/solicitada pela bateria para carga incompleta da bateria.	64
Figura 49: Imagem do chip Devkit contendo o módulo ESP32.....	66

Figura 50: Arquitetura interna funcional do módulo ESP32.....	67
Figura 51: Diagrama esquemático simplificado do módulo ESP32.....	69
Figura 52: Diagrama esquemático simplificado do módulo ESP32 – modo ativo.....	70
Figura 53 : Diagrama esquemático simplificado do módulo ESP32 – modo modem sleep	70
Figura 54 : Diagrama esquemático simplificado do módulo ESP32 – modo light sleep.....	71
Figura 55 : Diagrama esquemático simplificado do módulo ESP32 – modo deep sleep	72
Figura 56 :Diagrama esquemático simplificado do módulo ESP32 – modo hibernação.....	72
Figura 57 :Fluxograma do algoritmo para medição de vazão no modo deep sleep	73
Figura 58 : Potência gerada e consumida pelo módulo ESP32 Devkit e bateria com transmissão de dados a cada 60 s e utilização de Deep Sleep	75
Figura 59 : Potência consumida pelo ESP32 Devkit durante Deep Sleep e com operação apenas do coprocessador ULP realizando aquisição de dados para vazão constante em aproximadamente 5.41L/min.....	76
Figura 60 : Potência consumida pelo ESP32 Devkit durante Deep Sleep em intervalos de tempo de 1 s e aquisição de dados com ULP para vazão de 7.53 L/min.....	77
Figura 61: Potência consumida pelo ESP32 Devkit durante Deep Sleep em intervalos de tempo de 3 s e aquisição de dados com ULP para vazão de 7.48 L/min.....	78
Figura 62: Potência gerada e consumida pelo ESP32 Devkit no modo Deep Sleep, comunicação a cada 15 s e vazão variável.	80
Figura 63: Potência gerada e consumida pelo ESP32 Devkit no modo Deep Sleep, comunicação a cada 15 s e vazão fixa média de 5,7 L/min.....	80
Figura 64: Potência gerada e consumida pelo ESP32 Devkit no modo Deep Sleep, comunicação a cada 15 s e vazão fixa máxima de 7,8 L/min.....	81
Figura 65: Potência gerada e consumida pelo ESP32 Devkit no modo Deep Sleep, comunicação a cada 20 s e vazão variável	82
Figura 66: Potência gerada e consumida pelo ESP32 Devkit no modo Deep Sleep, comunicação a cada 20 s e vazão fixa média de 5,7 L/min.....	82
Figura 67: Potência gerada e consumida pelo ESP32 Devkit no modo Deep Sleep, comunicação a cada 20 s e vazão fixa máxima de 7,8 L/min.....	83
Figura 68: Potência gerada e consumida pelo ESP32 Devkit no modo Deep Sleep, comunicação a cada 60 s e vazão variável	84
Figura 69:Potência gerada e consumida pelo ESP32 Devkit no modo Deep Sleep, comunicação a cada 60 s e vazão fixa média de 5,7 L/min.....	84

Figura 70:Potência gerada e consumida pelo ESP32 Devkit no modo Deep Sleep, comunicação a cada 60 s e vazão fixa máxima de 7,8 L/min..... 85

LISTA DE ABREVIATURAS E SIGLAS

ABNT – Associação Brasileira de Normas Técnicas

ANEEL - Agência Nacional de Energia Elétrica

HIoT - *Human Internet of Things* (Internet das Coisas Humana)

IEEE - *Institute of Electrical and Electronic Engineers* (Instituto dos Engenheiros Elétricos e Eletrônicos)

IloT - *Industrial Internet of Things* (Internet das Coisas Industrial)

IoT - *Internet of Things* (Internet das Coisas)

IPv6 - *Internet Protocol – version 6* (Protocolo de Internet – versão 6)

M2M - *Machine to Machine* (Máquina para Máquina)

MCA – Metros de Coluna de Água

MCU – *MicroController Unit* (Unidade Microcontroladora)

NodeMCU – Placa de Desenvolvimento da Família do chip ESP 8266

QoS - *Quality of Service* (Qualidade de Serviço)

RSSF - Rede de Sensores sem Fio

RTC - *Real Time Clock* (Clock de Tempo Real)

SABESP - Saneamento Básico do Estado de São Paulo

ULP - *Ultra Low Power* (Potência Ultrabaixa)

IP – Internet Protocol (Protocolo Internet)

WSN – Wireless Sensor Network (Rede de Sensores Sem Fio)

SUMÁRIO

1	INTRODUÇÃO	10
1.1	JUSTIFICATIVA.....	10
1.2	OBJETIVO.....	13
1.3	ESTRUTURA E CONTEÚDO.....	13
2	REVISÃO BIBLIOGRÁFICA	15
2.1	COLHEITA DE ENERGIA E INTERNET DAS COISAS	15
2.2	CONCEPÇÃO DOS SISTEMAS DE COLHEITA DE ENERGIA	19
2.3	COLHEITA DE ENERGIA DE FONTES HIDRÁULICAS	22
3	PROPOSTA DO TRABALHO	29
3.1	HARDWARE.....	31
3.2	SOFTWARE	34
4	DESENVOLVIMENTO.....	37
4.1	PLATAFORMA DE TESTES	37
4.2	DEFINIÇÃO DAS VARIÁVEIS DE INTERESSE.....	39
4.3	PROJETO DO CIRCUITO DE ENERGIA.....	40
4.4	PROGRAMAÇÃO DO NODEMCU E DA INTERFACE VISUAL	42
4.5	MODOS DE BAIXO CONSUMO DE ENERGIA.....	44
5	ANÁLISE DO POTENCIAL DE COLHEITA DE ENERGIA	46
5.1	ENSAIOS DE VERIFICAÇÃO DO POTENCIAL DE COLHEITA DE ENERGIA .	46
5.2	REDUÇÃO DO CONSUMO DE ENERGIA - OPERAÇÃO EM MODEM SLEEP .	49
5.3	VERIFICAÇÃO DO CONSUMO DE ENERGIA COM COMUNICAÇÃO IOT	51
5.4	ANÁLISE DA GERAÇÃO DE ENERGIA PARA A CARGA DA BATERIA.....	62
6	ATUALIZAÇÃO DO HARDWARE DO DISPOSITIVO.....	66
6.1	ESTUDO DO ULP PARA REDUÇÃO DO CONSUMO ENERGÉTICO	66
6.2	PERFIS DE CONSUMO DE ENERGIA DO ESP32	68
6.3	ALGORITMO DE MEDIÇÃO PARA OTIMIZAÇÃO DE CONSUMO	73
6.4	ENSAIOS PARA VALIDAÇÃO DO POTENCIAL DE REDUÇÃO DE CONSUMO DO ESP32.....	75
6.5	ENSAIOS DE VALIDAÇÃO DO ALGORITMO PARA MEDIÇÃO DE VAZÃO EM MODO DEEP SLEEP	76
7	VERIFICAÇÃO DO CONSUMO DE ENERGIA NO MODO DEEP SLEEP	79
7.1	VERIFICAÇÃO DO CONSUMO DE ENERGIA COM TRANSMISSÃO DE DADOS A CADA 15 S.....	79

7.2	VERIFICAÇÃO DO CONSUMO DE ENERGIA COM TRANSMISSÃO DE DADOS A CADA 20 S.....	81
7.3	VERIFICAÇÃO DO CONSUMO DE ENERGIA COM TRANSMISSÃO DE DADOS A CADA 60 S.....	83
8	RESULTADOS E DISCUSSÕES	86
9	CONCLUSÕES	90
10	REFERÊNCIAS BIBLIOGRÁFICAS.....	93

.....

1 INTRODUÇÃO

1.1 JUSTIFICATIVA

A Internet das Coisas (IoT – *Internet of Things*) advém da evolução da tecnologia de comunicação Máquina-a-Máquina (M2M – *Machine to Machine*), que através da interconexão das “coisas” ou objetos inteligentes, visa enfatizar, além da monitoração e controle, os processos de otimização e autonomia. Através da combinação de diversas tecnologias, a IoT tem encurtado a distância de monitoramento e gerenciamento de processos de maneira crescente nos últimos anos, possibilitando a conexão de bilhões de dispositivos e ocasionando mudanças de processos e comportamento (PEREIRA & CARVALHO, 2017).

O termo IoT foi inventado por Kevin Asthon em 1999 no contexto de Cadeia de Suprimentos (*Supply Chain*) da indústria. Porém, a definição se expandiu para outras aplicações, como na área de saúde, transporte, comunicação (SUNDMAEKER et al., 2010). Atualmente, as pesquisas em torno da IoT têm atraído interesse e focado em diferentes aplicações (STANKOVIC, 2014). Cada foco de aplicação da IoT possui características e requisitos específicos. Quando aplicada à área industrial, a IoT industrial (IIoT – *Industrial Internet of Things*) busca simplificar e criar arquiteturas de sistemas que são mais acessíveis, com comunicações eficientes e interação entre a produção e sensores, atuadores, entre outros, para aprimorar o desempenho e flexibilidade da indústria (LYDON, 2014). A evolução tecnológica da IoT não é restrita à área industrial. Inúmeras aplicações de IoT envolvem telemetria, coleta de dados remotos em ambientes diversos, possibilidade de atuação direta sobre objetos de todos os tipos, relacionamento em rede e interação transparente entre objetos e entre objetos e pessoas (FILHO, 2016). Nessas aplicações de IoT, o foco é viabilizar a aquisição e disponibilização desses dados de forma interoperável e padronizada para uma tomada de decisão diferenciada.

Os sistemas e aplicativos da IoT apresentam como um dos principais pilares para sua efetiva implementação, as redes de sensores sem fio - RSSF (WSN – *Wireless Sensor Networks*). As RSSF são compostas por um grande número de nós sensores com capacidades restritas de processamento, armazenamento e energia. Os dispositivos sensores da IoT são compostos por dispositivos eletrônicos de baixo custo e com inteligência para coletar dados úteis e encaminhá-los ao usuário final. Esses dispositivos cooperam uns com os outros via

conexões sem fio, a fim de formar uma rede com o objetivo de disseminar e analisar as informações provenientes do meio ambiente. Quanto mais dados mensurados e informações estiverem disponíveis, melhor será a percepção do estado da aplicação e do sistema. Isso permite um processo de tomada de decisão mais eficiente, considerando uma gama mais ampla de critérios (SHAIKH & ZEADALLY, 2015).

Geralmente em aplicações de IoT usando RSSF, os dados coletados em um nó sensor são roteados através de outros nós da rede até chegarem em um nó de fronteira, onde são transferidos para um ponto da rede denominado de nó coordenador ou *sink node*. Este coordenador atua como um *gateway*, com maior capacidade de processamento para se comunicar com um nó gerenciador de tarefas através de conexões públicas que podem ser desde uma rede internet até conexões via satélite. O usuário final receberá os dados coletados e executará algum tipo de processamento a fim de obter os resultados para o qual o sistema foi proposto (TAN & PANDA, 2010).

A implementação das RSSF e aplicações de IoT têm um impacto ambiental no consumo de energia que se transforma em desafio a ser ultrapassado. Mesmo no modo de baixa energia, os 14 bilhões de dispositivos habilitados para rede, que estão atualmente em uso no mundo todo, desperdiçam 400 TWh (terawatt-hora) de eletricidade. Com 50 bilhões de dispositivos em 2020, o consumo aumentará em fator 3.5 para mais de 1.400 TWh. O desperdício de energia é uma questão global para a IoT e se opõe a seus benefícios de economia de recursos. Além disso, se todos esses dispositivos fossem alimentados por baterias, as pessoas precisariam se desfazer de centenas de toneladas de resíduos perigosos a cada ano, com um impacto ambiental irresponsável. Mesmo a reciclagem de baterias não é uma alternativa, pois é preciso de 6 a 10 vezes mais energia para recuperar metais de algumas baterias recicladas do que para produzi-las por outros meios, incluindo a mineração (ENOCEAN, 2015).

Eficiência energética pode ser definida como a possibilidade de melhorar o aproveitamento no consumo de energia associada a uma atividade, contribuindo assim para a redução dos gastos com este insumo. De forma simplificada, isso significa realizar a mesma atividade consumindo menos energia. Em uma época em que se evidenciam ações dedicadas à sustentabilidade e ao meio ambiente (CHU et al., 2007), o processo de eficiência energética está condicionado à integração de novas tecnologias ao meio industrial. Nesse contexto, a IoT tem sido considerada ideal para este ambiente (HUANG et al., 2011).

Especificamente sobre a aplicação da IoT para melhoria da eficiência energética, muitos trabalhos têm sido desenvolvidos (LIU et al., 2014). Tais aplicações podem consistir de simples

sensores (sensoriamento da temperatura de um ambiente), sensores mais complexos (sensoriamento do consumo energético de equipamentos), atuadores (sistemas de ar condicionado e motores) e até sistemas mais complexos (controladores industriais e sistema supervisórios de controle e aquisição de dados). Da mesma forma, a temática de colheita de energia do ambiente para aplicações de IoT tem tido crescente interesse (KAMALINEJAD et al., 2015).

Colheita ou recuperação de Energia (*Energy Harvesting*) é o processo pelo qual a energia presente em fontes externas no ambiente é capturada e armazenada para uso por dispositivos eletrônicos de escala reduzida para fins de monitoramento de variáveis de interesse. A colheita de energia está atrelada à recuperação de energia em pequenas quantidades e surge como um dos principais temas de pesquisa e desenvolvimento nas diversas áreas de engenharia, principalmente devido ao rápido progresso da IoT (TAN & PANDA, 2010). De acordo Kamalinejad et al. (2015) e Shaik & Zeadally (2015) existem várias formas de colheita de energia do ambiente, incluindo-se a energia solar, eólica, térmica, hidráulica, ondas eletromagnéticas e a energia gerada pelo movimento ou vibrações de estruturas físicas e do corpo humano. Essa energia pode ser capturada para abastecer dispositivos eletrônicos de baixo consumo de energia tais como sensores eletrônicos e sistemas de comunicação sem fio.

Dentre as possibilidades para colheita de energia, a vazão de fluidos é uma fonte essencial, em particular, a vazão existente nas redes de distribuição de fluidos em áreas urbanas e industriais torna-se uma fonte alternativa com enorme potencial para a colheita de energia (CUNEFARE et al., 2013). Os sistemas hidráulicos têm inerentemente alta intensidade de energia associada à pressão e fluxo hidráulicos. O levantamento do potencial de colheita de energia em vazões de sistemas hidráulicos tem sido realizado em diferentes configurações (ALAREFI & WALKER, 2015). Estes trabalhos variam desde vazões de distribuições hidráulicas prediais, até configurações utilizando vazões das tubulações industriais e de sistemas de abastecimento urbano (YAN et al., 2011). A colheita de energia em sistemas hidráulicos pode até viabilizar a utilização de nós sensores sem fio autossustentáveis do ponto de vista de alimentação elétrica (HOFFMANN et al., 2013).

No contexto da aplicação da IoT, em que os equipamentos se comunicam entre si e com a web, e da geração de energia de forma sustentável, capturar essa energia já disponível no ambiente é uma estratégia muito relevante. A colheita de energia torna-se neste cenário um aspecto fundamental para a viabilização dos avanços da IoT, uma vez que, a capacidade computacional para essas aplicações já está disponível localmente ou através de

armazenamento nas nuvens, restando como grande desafio, a implementação e alimentação elétrica de um enorme número sensores e atuadores e conectá-los de uma maneira viável e adequada (SHAIKH & ZEADALLY, 2015).

Com a disponibilização de RSSF e aplicações de IoT com capacidade de recuperação de energia do meio ambiente para suprir sua demanda energética (energia elétrica), tornam-se viáveis aplicações de monitoramento e processamento de dados onde as alternativas de redes cabeadas ou com baterias não são suficientes (ENOCEAN, 2015). Nesse contexto, este trabalho busca estudar o potencial de colheita de energia proveniente da movimentação de fluido em tubulações hidráulicas de baixa vazão e pressão para o suprimento energético de um dispositivo de medição de vazão para aplicações de IoT.

1.2 OBJETIVO

O objetivo deste trabalho é estudar o potencial de colheita de energia hidráulica para medição de vazão em tubulações de baixa vazão e pressão visando o desenvolvimento de uma aplicação de IoT energeticamente autossustentável.

1.3 ESTRUTURA E CONTEÚDO

A estrutura deste trabalho está dividida em 10 capítulos considerando esta introdução e as referências bibliográficas.

O capítulo 2 trata da revisão bibliográfica, buscando as mais relevantes pesquisas que já foram publicadas acerca do tema, os conceitos essenciais para o desenvolvimento do trabalho, as várias alternativas e tecnologias envolvidas na recuperação de energia do meio ambiente, as simulações e experimentos similares para o levantamento e aplicação do potencial de colheita de energia a partir da vazão hidráulica.

O capítulo 3 apresenta os materiais da proposta do trabalho, as premissas para o atingimento de resultados e as necessidades de equipamentos, hardware e software integrados para a realização das simulações de colheita de energia e medição de vazão.

O capítulo 4 discorre sobre o método de desenvolvimento e montagem do protótipo e seus componentes, as ferramentas computacionais integradas ao sistema, levantamento e análise dos dados extraídos a partir das simulações executadas.

O capítulo 5 analisa o potencial de colheita de energia hidráulica do protótipo montando, apresentado alguns pontos de interesse na operação do protótipo.

O capítulo 6 detalha sobre a alteração do hardware microcontrolador do dispositivo de medição visando uma redução do consumo de energia. Os modos de operação desse novo hardware e sua adequação à proposta desse trabalho são discutidos. Ensaios para validação do algoritmo desenvolvido para este novo hardware, bem como para verificação do consumo de energia do dispositivo de medição de vazão operando com o mesmo são apresentados.

O capítulo 7 apresenta os resultados obtidos com o dispositivo de medição operando em modo de baixo consumo de energia e de acordo com períodos diferentes de transmissão dos dados (vazão/volume) para a nuvem.

O capítulo 8 apresenta uma compilação dos resultados obtidos nos ensaios e configurações desenvolvidas neste trabalho. Uma comparação e discussão dos resultados é apresentada.

As conclusões obtidas com o desenvolvimento deste trabalho são apresentadas no capítulo 9.

2 REVISÃO BIBLIOGRÁFICA

2.1 COLHEITA DE ENERGIA E INTERNET DAS COISAS

A colheita ou recuperação de energia do meio ambiente tem sua motivação no rápido desenvolvimento das redes de sensores sem fio dentro do paradigma da Internet das Coisas, onde um dos principais desafios para a sua efetiva consolidação e desenvolvimento de novas aplicações é a disponibilização de energia para os seus dispositivos e componentes. Quando a energia dos dispositivos IoT atinge os níveis mínimos necessários para o correto funcionamento, os resultados para o qual a rede de sensores e sua aplicação foram projetados não são mais confiáveis e a perda de funcionalidade de todo o sistema fica comprometida.

Em geral, a principal fonte de energia de redes sem fio está baseada em uma bateria existente em cada um dos dispositivos, entretanto, a utilização da energia de baterias traz desafios adicionais, incluindo, por exemplo, a diminuição ou perda de energia das baterias mesmo quando fora de uso, com a limitação da densidade de energia existente. Outro aspecto de máxima importância refere-se à operação e manutenção do conjunto de componentes com baterias e sujeitos às condições adversas de tempo e em locais de difícil acesso, condições que geralmente estão presentes nas redes sem fio de monitoramento e controle existentes.

Esses aspectos podem colocar em risco as instalações através do vazamento de produtos químicos dessas baterias ou mesmo inviabilizar a reposição desses componentes. Deve-se considerar ainda que o restante dos dispositivos constituintes dos nós das redes sem fio são fabricados para operação ininterrupta e com ciclo de vida muito superior aos das baterias. Como consequência, pode haver a parada de funcionamento de alguns nós e mesmo que sejam poucos, toda a rede de sensores fica comprometida em seus aspectos de operação e desempenho. O consumo de energia e utilização de baterias nas aplicações de IoT devem ser considerados em um contexto que envolve outros requerimentos importantes, que, no mínimo, devem incluir: (ENOCEAN, 2015)

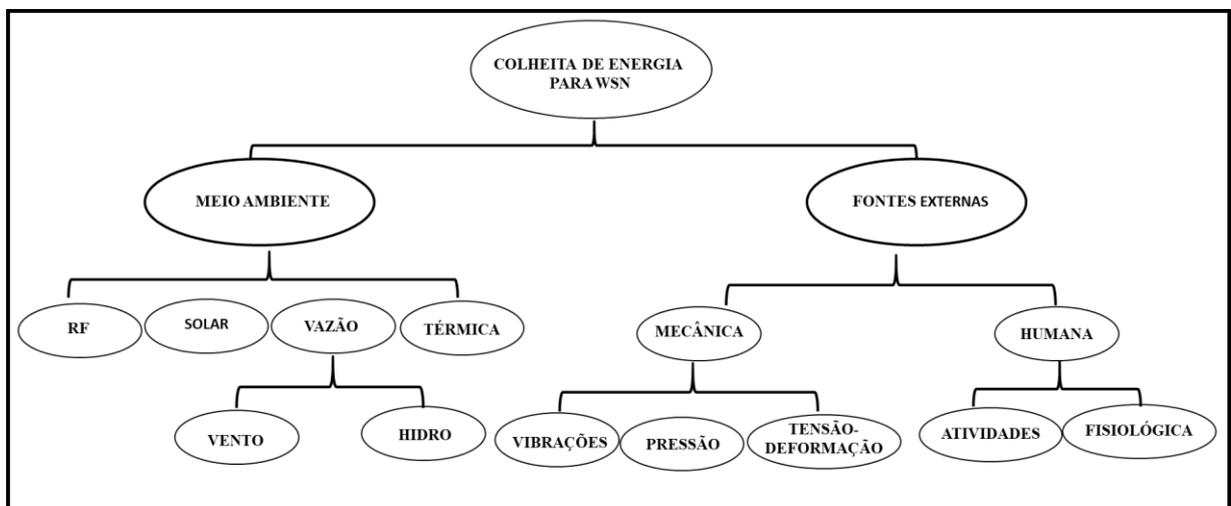
- Investimento de instalação: um grande número de novos sensores e atuadores precisam ser alocados, muitas vezes na infraestrutura existente, ao mesmo tempo em que se mantém a complexidade da instalação atual;

- Esforço de manutenção: as exigências de serviços de manutenção para esses dispositivos devem ser mínimas para implementações em grande escala, principalmente em relação ao quesito energético (troca de bateria);

- Comunicação IPv6: necessidade de consolidação do formato de endereçamento dos dispositivos via Protocolo de Internet (IP) versão 6. No entanto, não é necessário que os dispositivos se comuniquem fisicamente via IPv6, desde que, a conversão entre o protocolo do dispositivo de rede e IPv6 seja transparente (ex: tunelamento via 6LowPAN);
- Confiabilidade e segurança: a troca de dados confiável e segura é fundamental quando as informações entre equipamentos e sistemas são realizadas em rede;
- Impacto Ambiental: o consumo de energia, mesmo no modo de baixa potência, com os “bilhões” de dispositivos habilitados para conexão em rede tornarão o consumo de energia uma questão global para a IoT. Além disso, se todos esses dispositivos fossem alimentados por baterias, a quantidade de resíduos perigosos gerados todos os anos representaria um impacto ambiental inviável.

A energia do ambiente ao nosso redor pode encontrar-se em qualquer lugar e está disponível na forma de radiação solar, rádio frequências, energia cinética e mecânica. Embora a geração de macro energia esteja presente na história humana e sua evolução, o grande desafio frente aos avanços da mecatrônica e internet das coisas está na colheita de baixas quantidades de energia a fim de alimentar dispositivos de IoT em redes sem fio, suprindo de forma quase perpétua a energia elétrica para tais dispositivos, tornando-os dependentes apenas de baterias de apoio e seus ciclos de recarregamento. Conforme Shaikh & Zeadally (2015), a Figura 1 apresenta a taxonomia para classificação das técnicas de recuperação de energia.

Figura 1: Taxonomia de fontes de colheita de energia para redes de sensores sem fio.



Fonte: Adaptado (SHAIKH & ZEADALLY, 2015).

De acordo com Shaikh & Zeadally (2015), as duas grandes categorias para fontes de colheita de energia são: fontes do meio ambiente e fontes externas. Essas fontes devem ser analisadas conforme as necessidades de consumo dos dispositivos a serem alimentados e conforme os mecanismos de obtenção dessa energia, os quais podem ser categorizados em:

- Energia mecânica ou de vibrações e movimento de pisos e máquinas que transferem energia para baterias através de transdutores eletromecânicos, eletromagnéticos, eletrostáticos ou piezelétricos;
- Energia térmica retirada de máquinas em operação, com sensores e transdutores do tipo células de efeito *Seebeck*;¹
- Energia biomecânica colhida dos movimentos de seres humanos;
- Energia do atrito entre substância e materiais ou efeito tribo elétrico e a série tribo elétrica definida como uma relação de materiais que, através do atrito, tem a capacidade de obter cargas elétricas;
- Energia solar através de painéis solares com diversas aplicações e grandes avanços na solução de energias alternativas.
- Energia de fontes magnéticas ou eletromagnéticas incluindo-se rádio frequências e as diversas fontes de radiação presentes no meio ambiente;
- Energia de sistemas hidráulicos e variações de pressão de vazão de líquidos;
- Energia retirada de reações eletroquímicas e radioativas.

Conforme Tan & Panda (2010), diante da grande quantidade de fontes para recuperação de energia do ambiente, sua utilização e o desenvolvimento de métodos e tecnologias capazes de viabilizar sua efetiva implementação tem sido amplamente pesquisada, através do uso de protótipos e novos materiais, visando maximizar os resultados da colheita de energia. Para as aplicações de IoT, torna-se evidente as vantagens da colheita de energia na sua implementação, destacando-se os seguintes aspectos:

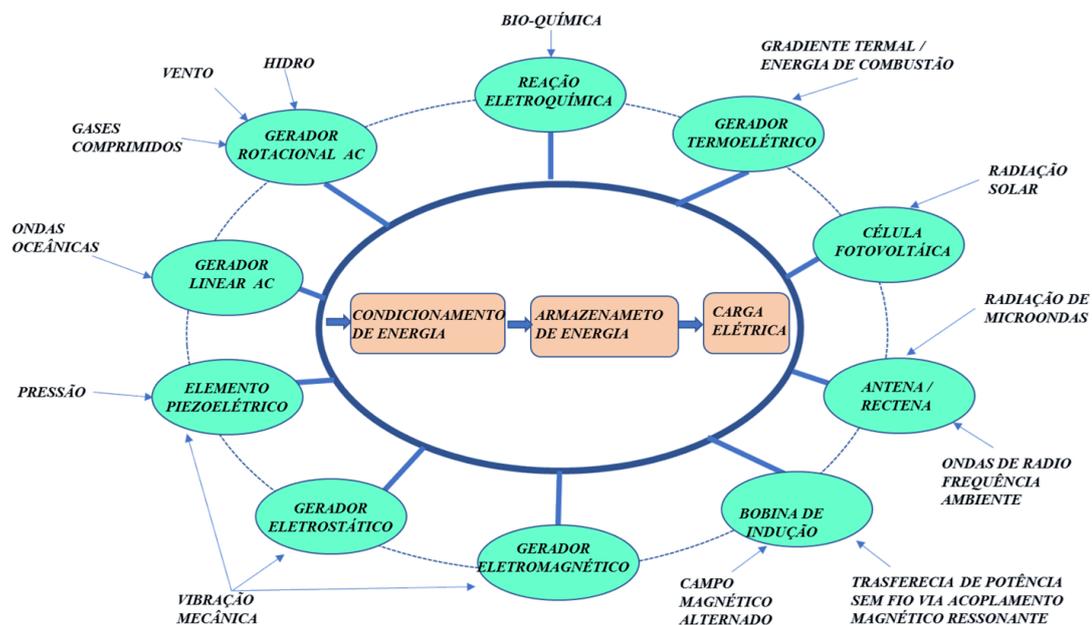
- Redução da dependência de baterias: a energia recuperada do meio ambiente pode tornar-se suficiente para eliminar o uso de baterias;

¹ Efeito Seebeck: Produção de tensão elétrica entre duas junções de condutores quando elas estão a diferentes temperaturas

- Redução de custos de instalação: os sensores autoalimentados não requerem cabos e instalações adicionais, tornando-se fácil sua instalação;
- Redução de custos de manutenção ou reposição de baterias;
- Habilita o monitoramento contínuo em locais de difícil acesso;
- Habilita a adoção de soluções de longo prazo, com os sensores autoalimentados mantendo-se funcionais enquanto as fontes de colheita de energia estiverem disponíveis;
- Redução do impacto ambiental, eliminando a necessidade de baterias.

Para obtenção do maior número de vantagens com a implementação da colheita de energia, torna-se importante o estudo das várias fontes para recuperação de energia a partir do meio ambiente e suas características relativas aos princípios de geração de energia disponíveis e os mecanismos transdutores de energia utilizados para viabilizar a energia recuperada para os nós sensores das redes sem fio existentes nas aplicações IoT. A Figura 2 apresenta as fontes de energia e os transdutores necessários (TAN & PANDA, 2010).

Figura 2: Fontes de energia e transdutores para a alimentação de nós sensores



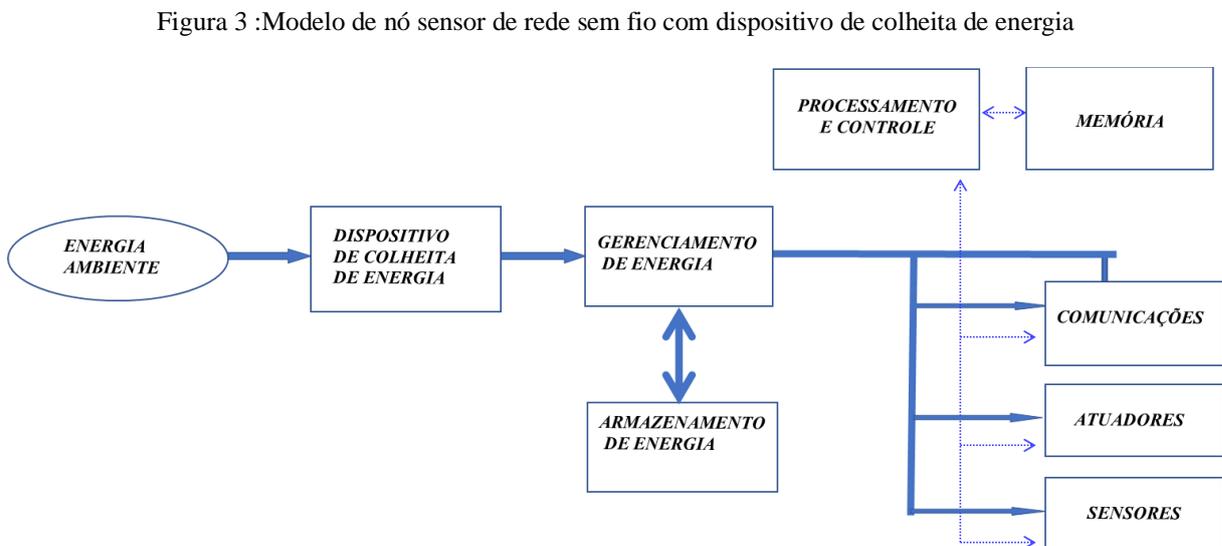
Fonte: Adaptado (TAN & PANDA, 2010).

2.2 CONCEPÇÃO DOS SISTEMAS DE COLHEITA DE ENERGIA

Conforme EDNA (2018) com o rápido crescimento das aplicações de IoT surge a necessidade do gerenciamento da energia consumida, com a responsabilidade de que os dispositivos e sensores utilizados consumam a menor quantidade de energia possível. Estes aspectos são essenciais na determinação dos elementos coletores e as tecnologias aplicadas para viabilização da colheita de energia. Os sistemas de coleta de energia geralmente consistem em: elementos de coleta de energia, hardware de conversão e condicionamento de energia e dispositivos de armazenamento. A energia colhida deve ser convertida em eletricidade e condicionada de uma forma apropriada para carregar as baterias do sistema ou alimentar diretamente a carga conectada.

O hardware de conversão de energia é projetado de diferentes maneiras para colher os vários tipos de energias da maneira mais eficiente possível. Um exemplo disso é o material da célula fotovoltaica no painel solar que é dopado de tal forma que quando a radiação solar é absorvida pela célula seja máximo. Todo o processo de coleta de energia envolve hardware de conversão de energia ambiental recuperada em energia elétrica, condicionamento de energia elétrica pelo circuito de gerenciamento de energia e, em seguida, armazenamento em elementos de baterias e finalmente fornecer para a carga elétrica.

A Figura 3 mostra um modelo sensor com a introdução funcional do bloco de colheita de energia (GILBERT& BALOUCHI, 2008).



Fonte: Adaptado (GILBERT& BALOUCHI, 2008).

Segundo Gilbert & Balouchi (2008), os requerimentos de consumo de cada um dos elementos de um nó sensor dependem da aplicação específica e torna-se difícil generalizar sobre qual parte do nó sensor consome mais energia. É comum, nos casos da presença e atuadores, estes consumirem uma porção principal da energia total. Adicionalmente, o consumo depende do modo de operação de um dispositivo, ou seja, se o dispositivo apresenta a alternativa de um modo de operação de redução de energia ou *sleep mode*, no qual o sistema permanece inativo com consumo mínimo de energia e, quando solicitado via rede de comunicações, ativa-se e executa todas as funcionalidades para o qual foi destinado. A escolha do processador é um fator determinante para o consumo de energia de um nó sensor. A unidade é responsável pelo gerenciamento de aquisição de dados, execução dos protocolos de comunicação, programação e preparação dos pacotes de dados para transmissão através da reunião, filtragem e sincronização dos dados dos sensores. O consumo do processador depende da sua arquitetura, tecnologia e velocidade de relógio ou clock utilizado.

De acordo com Shaikh & Zeadally (2015), o desenvolvimento de dispositivos e sistemas para a recuperação de energia devem considerar os aspectos de hardware e software aplicados, uma vez que, a quantidade de energia a ser colhida deve viabilizar a operação em condições específicas para cada aplicação dentro da arquitetura das redes de sensores sem fio a serem disponibilizadas. Portanto, é importante assegurar que a potência elétrica gerada apresente níveis de tensão e corrente adequados para a aplicação de projeto e, em consequência, a fonte de colheita de energia possa suprir os níveis de potência adequados aos dispositivos das redes de sensores. Por exemplo, nas fontes de colheita de energia solar, o aumento das células resulta diretamente no aumento da potência a ser disponibilizada.

Entretanto, conforme a natureza das fontes de colheita de energia, as incertezas das variações de níveis de energia recuperada tornam sua aplicação muito complexa. Nas aplicações de colheita e energia derivadas de vibrações no ambiente industrial dependem diretamente do aumento das vibrações existentes nas máquinas de produção, mas o aumento da vibração afeta diretamente as condições operacionais desses equipamentos, restringido o potencial dessas aplicações. Portanto, muitas aplicações de colheita de energia para redes de sensores sem fio requerem que os nós sensores sejam pequenos e flexíveis de acordo com a densidade de potência recuperada, sendo essa métrica muito utilizada nos desenvolvimentos de dispositivos e sistemas de colheita e energia.

Em consequência, a quantidade e a taxa de energia recuperada no tempo são dois parâmetros críticos para modelagem das aplicações e o desenvolvimento de sistemas de colheita

de energia. A previsibilidade e controlabilidade são características que distinguem as várias fontes de colheita de energia. Para fontes controláveis, não há a necessidade de previsão no tempo, uma vez que a energia estará disponível quando requerida. Nas fontes não controláveis, a previsão de disponibilidade deve ser avaliada a fim de viabilizar sua aplicação. A Tabela 1 apresenta uma visão geral das diversas fontes de colheita e relação aos aspectos de controle e previsão inerentes natureza dessas fontes (SHAIKH & ZEADALLY 2015).

Tabela 1: Fontes e Colheita de Energia e Aspectos de Controle e Previsão

Fonte de Energia	Previsível	Imprevisível	Controlável	Não controlável
Radiofrequência	X		X	
Solar	X			X
Térmica		X	X	
Vazão (Vento)	X			X
Vazão (Hidro)	X			X
Mecânica - Vibração		X	X	
Mecânica - Pressão		X	X	
Mecânica - Tração		X	X	

Fonte: Adaptado (SHAIKH & ZEADALLY, 2015)

No entanto, em muitas situações, uma variedade de fontes de energia coexiste. Por exemplo, energia cinética e energia térmica normalmente coexistem em máquinas e corpos humanos. A energia solar fornece não apenas uma fonte de energia de luz visível, mas também energia térmica de diferentes comprimentos de onda. Nestas situações, os coletores de energia de fonte única não podem colher toda energia e existe um "desperdício" de uma quantidade significativa da energia aproveitável, impedindo assim a maximização da capacidade de colheita. Portanto, os coletores de energia híbrida que são capazes de colher múltiplas fontes de energia tem sido motivo de pesquisa e desenvolvimento. Com a união de coletores de energia da luz, cinética e energia térmica, a otimização estrutural traz novas perspectivas, com configurações responsáveis por diferentes fontes de energia integradas em uma estrutura híbrida. O número de fontes de energia recuperáveis para os coletores de energia híbrida é aumentado e a potência de saída é significativamente melhorada, possibilitando maior eficiência do sistema de colheita de energia (BAI, JANTUNEN & JUUTI, 2018).

2.3 COLHEITA DE ENERGIA DE FONTES HIDRÁULICAS

A vazão de fluídos é considerada como um tipo de fonte de recuperação de energia “pronta” para exploração e presente praticamente em todas as infraestruturas de instalações, seja, elas comerciais, industriais ou urbanas. Fontes de energia ambiente que são essencialmente estáveis estão baseadas no fluxo de fluido, como em correntes de vento e ar e fluxo de água em canais naturais ou através de tubos, ou em torno de movimento contínuo de um objeto, como um eixo rotativo. As fontes de energia existentes no ambiente que estão prontas para utilização são baseadas no fluxo de fluidos em canais naturais ou em dutos construídos. Essas fontes são amplamente utilizadas em macroescala para geração de energia elétrica em turbinas eólicas e hidrelétricas, mas também são amplamente consideradas para aplicações de colheita de energia em menor escala (GILBERT & BALOUCHI, 2008)

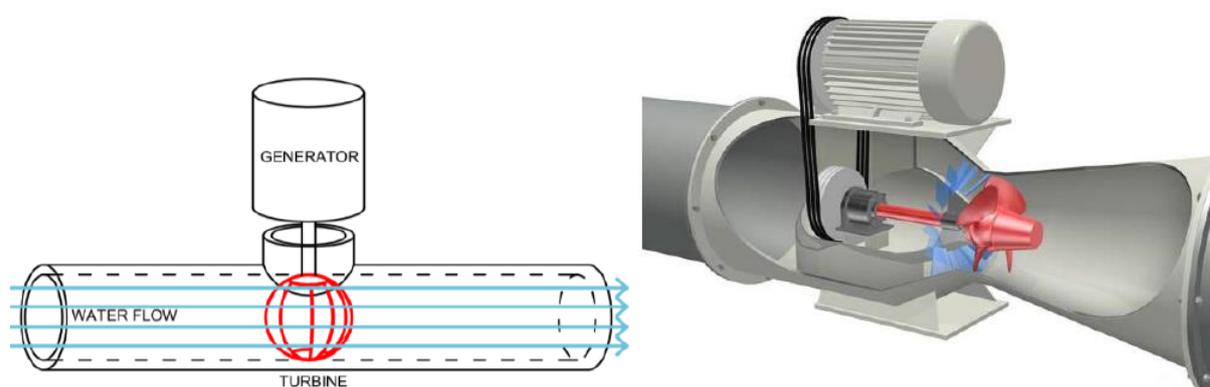
A colheita de energia hidráulica em pequenas quantidades através do aproveitamento do abastecimento de água no meio urbano e industrial segue o princípio básico que proporciona a construção de usinas hidroelétrica, ou seja, o uso da velocidade ou energia cinética do fluxo de água para movimentar mini e microturbinas e gerar eletricidade. A quantidade de energia gerada de sistemas de distribuição de água é diretamente relacionada com a taxa de vazão e pressão estática devido às diferenças de elevações existentes na tubulação de transporte dos fluídos, sendo que, quanto maior esses valores, maior a energia cinética existente para geração de energia elétrica. A pressão existente nos circuitos hidráulicos impulsiona a água através das tubulações e determina a sua vazão, mas, para evitar o desperdício ou inundações, os dutos são providos de válvulas redutoras de pressão (YAN, IBRAHIM & NOR, 2011).

No Brasil, as normas ABNT regulamentam o projeto de redes de distribuição de água e de instalações hidráulicas prediais. A norma ABNT NBR 12218:2017 - Projeto de Rede de Distribuição de Água para Abastecimento Público (ABNT, 2018) e a norma ABNT NBR 5626:1998 - Instalação predial de água fria (ABNT, 1998) regulamentam informações pertinentes ao trabalho como pressões e vazões de abastecimento. No contexto desse trabalho, que foca em sistemas hidráulicos de baixa pressão e vazão, os valores padrões normatizados para sistemas de abastecimento predial determinam pressões mínimas dinâmicas de distribuição de 10 mca (metros de coluna de água) e pressões estática máximas de 50 mca no abastecimento público. Considerando os valores de vazão nos pontos de suprimento predial e os valores mínimos de hidrômetros prediais de até 1,5 m³/h, uma vazão máxima de 25 L/min pode ser encontrada.

O potencial para colheita de energia a partir da vazão das redes de distribuição de água inclui um grande espectro de valores em relação quantidade de energia recuperada. Considerando sistemas hidráulicos de baixa pressão e vazão, é possível encontrar soluções com geração de energia elétrica a partir de micro e miliwatts, conforme apresentado em Yan, Ibrahim & Nor (2011), Hoffman et al. (2014) e Kayaalp, Zengin & Kara (2014). Considerando desenvolvimentos para geração de energia em grandes edificações urbanas, Daryl (2016) apresenta a colheita de até 100 watts de energia em redes de abastecimento de até 100mm de diâmetro. Chen (2013) apresenta um conceito de colheita de energia usando protótipos de turbinas axiais internas, alcançando até 228 watts de energia recuperada.

Casini (2015) descreve que o formato mais comum de colheita de energia hidráulica é através da instalação de dispositivos diretamente nas tubulações de fluidos em escoamento. Esse formato pode ser usado com um grande intervalo de vazões e pressões do fluido, além de ser compatível com os diferentes tipos de materiais usados em tubulações (concreto, aço e etc). Esses sistemas de colheita de energia podem ser instalados internamente ou externamente nas tubulações. A Figura 4 apresenta a estrutura usada nos sistemas de colheita de energia hidráulica com dispositivos internos à tubulação. Nesse tipo de solução, a maior parte do mecanismo de geração permanece interna na tubulação, simplificando e reduzindo o espaço físico necessário da instalação.

Figura 4 : Colheita de energia com dispositivos internos à tubulação hidráulica

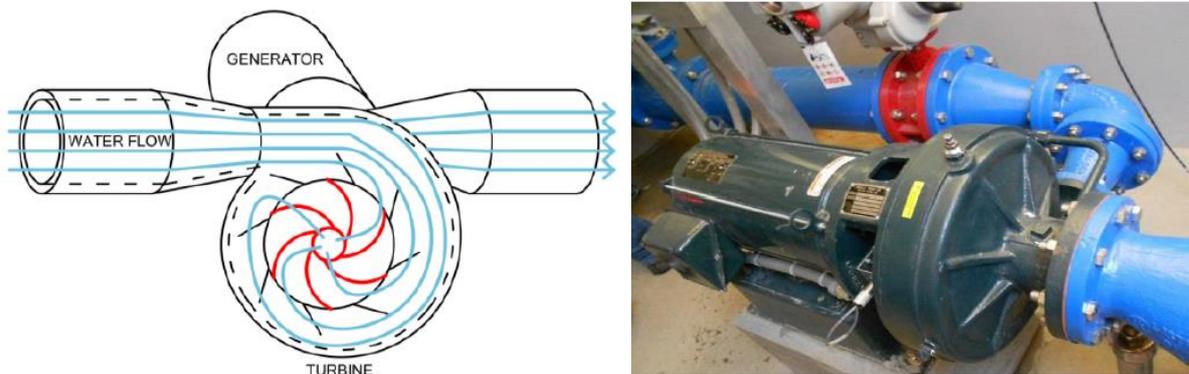


Fonte: (CASINI, 2015)

A Figura 5 apresenta a estrutura usado nas soluções de colheita de energia hidráulica com dispositivos externos à tubulação. Nesse tipo de solução, todo o mecanismo de geração é construído externamente à tubulação, de forma que o espaço físico necessário é geralmente

maior. Em contrapartida, o potencial de colheita de energia não fica restrito somente à largura da tubulação (vazão e pressão do fluido), e diferentes mecanismos externos de geração podem ser usados.

Figura 5 :Colheita de energia com dispositivos externos à tubulação hidráulica



Fonte: (CASINI, 2015)

Neste trabalho de Casini (2015), um projeto para recuperação de energia com turbinas axiais de instalação interna nas tubulações, conforme a Figura 4, teve como objetivo a recuperação de energia em grandes trechos de tubulações hidráulicas na cidade de Hong Kong. Nestes modelos de colheita de energia foi possível recuperar desde 10 W até potências da ordem de 100 KW. Considerando as aplicações com sistemas externos locais. O trabalho apresenta uma aplicação predial com turbinas de eixo vertical, a qual obteve capacidade de geração individual de até 100W e através da integração de vários sistemas alcançar uma economia de 720 KWh de eletricidade.

Em aplicações urbanas habitacionais com espaços reduzidos e restrições de instalação nas entradas hidráulicas, torna-se necessária a integração de dispositivos nos sistemas de dutos de distribuição, sem a interferência na vazão de água. Em Becker et al. (2013) apresenta o desenvolvimento de um medidor de água acoplado a um micro hidrogerador específico, o qual foi capaz de gerar até 300 mW em uma vazão de 20 l/min. O medidor de água desenvolvido utiliza um rotor diretamente acoplado a um transdutor eletromagnético, constituído por um ímã de dois polos e três bobinas de indução, juntamente com uma bateria conforme Figura 6.

Figura 6 : Medidor de água predial com coletor de energia acoplado



Fonte: (BECKER et. al, 2013)

No Brasil, de acordo com Portinari (2017), pretende-se aproveitar a pressão hidráulica nas tubulações de água para recuperação de energia através de mini hidrogeradores em 200 pontos da região da cidade de São Paulo. Nessa proposta, a energia recuperada será entregue à concessionária de energia em troca de abatimento dos gastos com energia da empresa SABESP (Saneamento Básico do Estado de São Paulo). Essa proposta é possível, pois a Aneel (Agência Nacional de Energia Elétrica) permite a microgeração de energia de até 75 KW desde que a fonte seja renovável ou sustentável.

Os protótipos para colheita de energia de vazões hidráulicas têm apresentado resultados diversos em aplicações com arquiteturas híbridas que incluem várias fontes de colheita de energia unificadas, com equipamentos e circuitos eletrônicos que possibilitam essa diversidade, como por exemplo, a fabricação de mini hidrogeradores comercialmente disponíveis para utilização em sistemas de irrigação, conforme mostrado na Figura 7 (MORAIS et al., 2008).

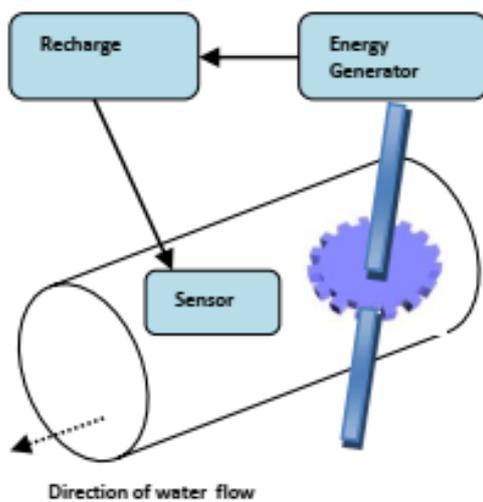
Kayaalp, Zengin & Kara (2014) desenvolveram um protótipo de um sistema de monitoramento da rede de distribuição de água através da inserção de sensores de pressão totalmente alimentados a partir da colheita de energia da vazão de água da tubulação monitorada. A colheita de energia foi realizada conforme o esquema de montagem da Figura 8. Esse esquema de colheita de energia, baseado na utilização de um gerador adaptado às condições do experimento, conseguiu gerar em torno de 240 mW através de uma pressão dinâmica da água na tubulação de 3 a 6 bar.

Figura 7 : Mini hidrogerador para sistema híbrido de colheita de energia



Fonte: (MORAIS et al., 2008)

Figura 8 : Mecanismo de recuperação de energia hidráulica para alimentação de sensores de pressão



Fonte: Adaptado (KAYAALP, ZENGİN & KARA, 2014)

Conforme a Figura 8, os sensores de pressão monitoram quedas de pressão anormais que indicam vazamentos ou quedas bruscas de vazão a fim de manter a qualidade da distribuição de água nas tubulações. A energia recuperada foi utilizada para a recarga das baterias dos sensores de pressão através de um circuito integrado específico de recarga e de proteção, mostrando-se eficiente para manter o sistema em funcionamento 24 hs sem qualquer outra intervenção necessária.

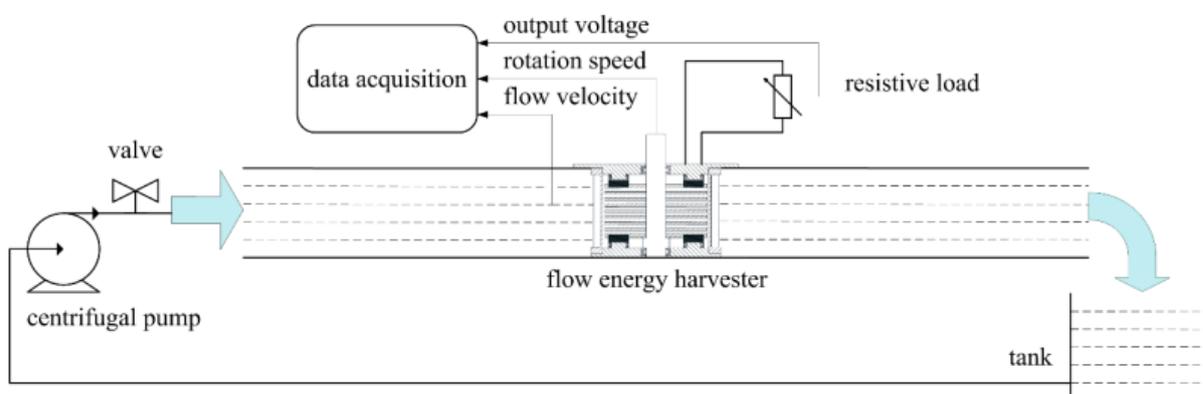
A colheita de energia da vazão de água apresenta um potencial de utilização em concentrações urbanas e edifícios de grande porte bastante significativo. De acordo com Daryl (2016), um sistema de colheita de energia foi desenvolvido para instalação em um edifício de grande porte em Hong Kong, sendo que, o mini hidrogerador concebido pela Escola Politécnica de Hong Kong com capacidade de gerar até 100 watts, com uma vazão de 10 m³/h, obtida com algumas variações sazonais do consumo de água nas instalações.

A estrutura dos protótipos para de colheita de energia pode variar também com a quantidade de mini hidrogeradores instalados. Conforme apresentado em Alarefi (2015), através da utilização de quatro mini hidrogeradores instalados em um circuito de vazão simulada de até 12 L/min e equipados com um sistema propulsor de hélices radiais internas acopladas, foi possível obter uma tensão de saída variando de 18 a 75 volts e 800 mA. Com os circuitos conversores e carga utilizada para alimentação de um aparelho celular, o resultado da colheita de energia foi de até 4,5 watts.

De acordo com Wang & Zhang (2018), em um experimento de avaliação de colheita de energia hidráulica, uma ideia direta para a captação de energia de fluxo de água é a miniaturização das turbinas convencionais e geradores elétricos. Os desenvolvimentos de sistemas coletores de energia eletromagnética com um rotor de turbina com ímãs e bobinas externas com núcleos de ferrite, tornaram-se nos últimos anos de grande interesse de pesquisas.

Baseado neste princípio, um protótipo foi montado conforme a Figura 9, utilizando um medidor de vazão e um tacômetro.

Figura 9 : Protótipo de colheita de energia da vazão de água com dispositivo eletromagnético



Fonte: Adaptado (WANG & ZHANG, 2018)

Os testes conforme apresentados na Figura 9 foram executados primeiramente sem carga e após com uma carga resistiva, com a utilização de um osciloscópio como equipamento de medição e aquisição de dados. Os resultados obtidos variaram de 1 a 13 volts pico a pico sem carga e de 1 até 9 volts pico a pico com cargas variando de 0 até 1000 ohms, com picos de energia de até 23 mW.

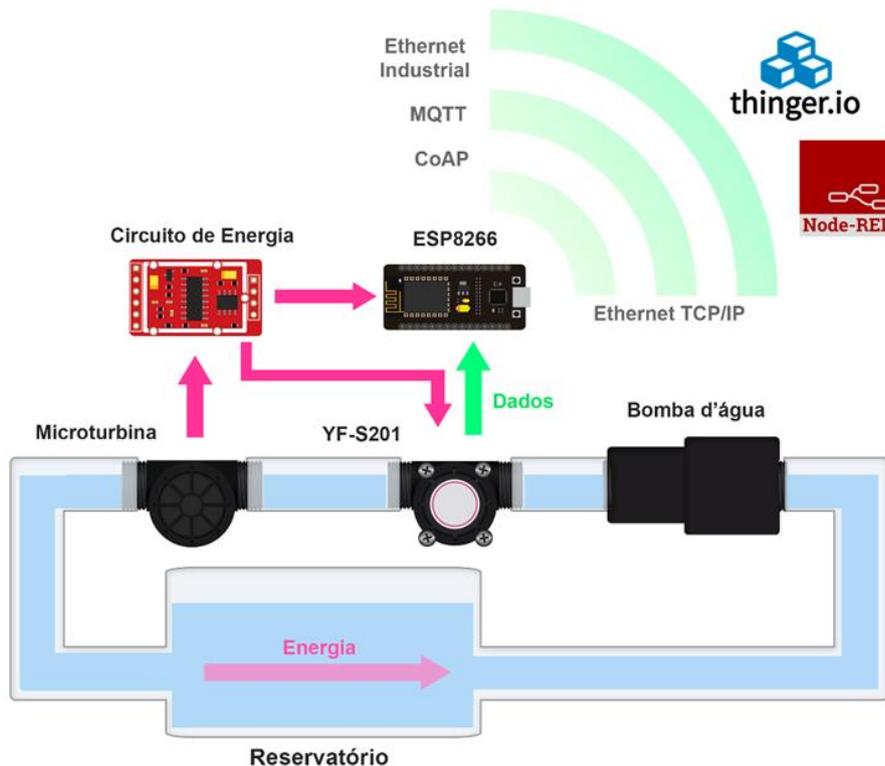
3 PROPOSTA DO TRABALHO

A proposta deste trabalho consiste em estudar o potencial de colheita de energia hidráulica em tubulações de baixa vazão e pressão para o desenvolvimento de uma aplicação de IoT energeticamente autossustentável para a medição de vazão. A partir da definição dos objetivos do trabalho, as etapas de trabalho englobaram:

- Desenvolvimento de uma plataforma de testes capaz de simular o escoamento de fluidos em tubulações;
- Desenvolvimento de um ambiente para realização de medições de grandezas elétricas envolvidas na micro hidrogeração de energia e das variáveis referentes ao consumo energético do sistema proposto;
- Implementação de técnicas e mecanismos de baixo consumo e otimização do processamento realizado pelo microcontrolador de modo a consolidar a viabilidade do sistema proposto;
- Desenvolvimento de um sistema de comunicação baseado em Internet das Coisas (IoT) para habilitar o armazenamento e disponibilização online e padronizada das informações de vazão do processo.

A plataforma de simulação foi construída para possibilitar os testes de escoamento de fluidos em tubulações de acordo com os padrões existentes em redes de distribuição hidráulicas (vazões de 0 a 10L/min), com a medição da vazão criada no circuito do fluido. O sistema descrito está ilustrado na Figura 10, composto por um reservatório e bomba d'água para simulação da vazão do fluido por toda a estrutura. A partir de uma microturbina hidráulica comercialmente disponível, pretende-se gerar energia elétrica oriunda da energia cinética associada ao fluido em movimento no sistema e realizar, a partir desta, a alimentação do módulo Wi-Fi ESP8266 (ESP8266, 2017). O sensor de fluxo (YFS201, 2017) é responsável pela coleta de informações relacionadas à vazão do fluido, as quais serão tratadas e disponibilizadas online pelo módulo ESP8266. Para que a alimentação seja realizada de maneira eficiente e adequada aos dispositivos integrantes, torna-se necessário o dimensionamento de um circuito responsável pela regulação e armazenamento da energia gerada pela microturbina e circuitos auxiliares de aquisição e análise dos dados produzidos.

Figura 10 : Esquemático da plataforma de colheita de energia hidráulica e medição de vazão



Fonte: Autoria Própria

As atividades de implementação englobam o desenvolvimento da comunicação entre a plataforma montada e aplicativos remotos de monitoramento através da utilização de softwares usando protocolos de comunicação sem fio baseados no Wi-Fi padronizados e compatíveis com a IoT, tais como o HTTP e o MQTT. Ainda, para monitoramento e disponibilização dos dados coletados de forma acessível é necessária a utilização de interfaces de interoperabilidade entre aplicações, tais como REST API, com a utilização de plataformas *open source* de monitoramento, por exemplo, Thinger.io.

O presente trabalho tem seu foco principal no levantamento do potencial da colheita de energia para o desenvolvimento do sistema de medição e monitoramento de vazão energeticamente autossustentável, a partir da montagem da plataforma descritos acima e a análise dos resultados de microgeração de energia pela microturbina versus a demanda da solução proposta. Após essa etapa, o objetivo é a integração da solução com dispositivos IoT e IIoT através da implementação dos protocolos de comunicação para troca de informações com a plataforma estabelecida.

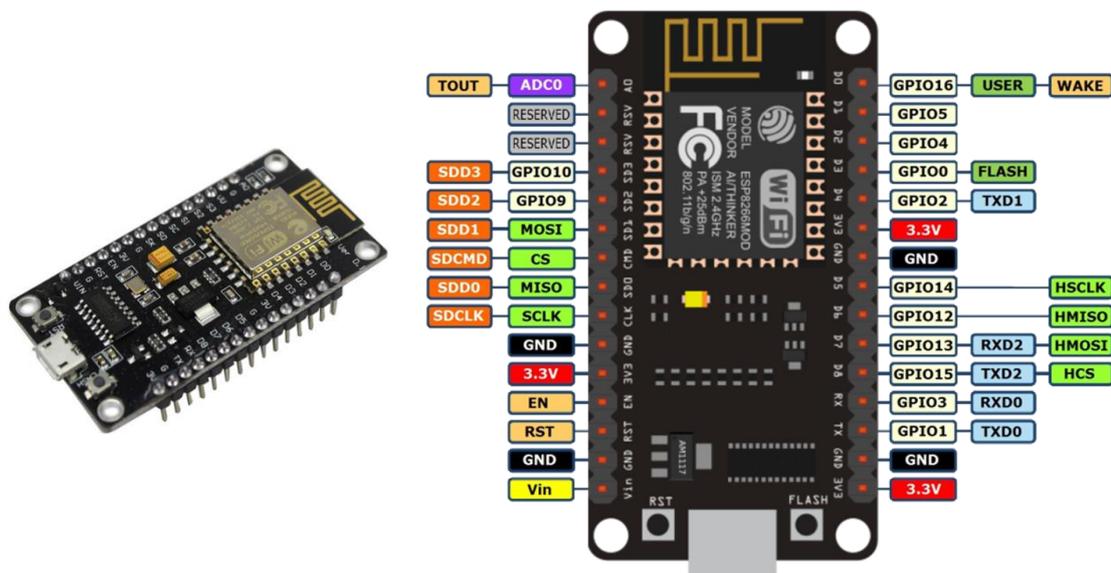
3.1 HARDWARE

Módulo ESP8266 e NodeMCU

O módulo ESP8266 é um SOC (System-on-Chip) com CPU 32-bit RISC Tensilica Xtensa LX106 rodando a 80 MHz, 64 KB de memória RAM de instruções e 96 KB de dados independente com protocolo TCP/IP integrado, que possibilita acesso do microcontrolador à rede Wi-Fi (ESP8266, 2017). Possui um processamento on-board e capacidade de armazenamento que permite integração com sensores e aplicações com outros dispositivos através de suas entrada e saídas (GPIOs).

Neste trabalho, o hardware que será utilizado é o NodeMCU conforme mostrado na Figura 11, que é uma plataforma open source de IoT. O NodeMCU é uma placa de desenvolvimento que combina o chip ESP8266-12E, uma interface USB-serial e um regulador de tensão, além de 11 pinos de I/O e conversor analógico-digital. A programação pode ser feita usando a IDE do Arduino e a linguagem empregada para programação desse microcontrolador também pode ser utilizada no NodeMCU mediante algumas configurações. Serão estudados métodos e estratégias para reduzir o consumo energético do módulo ESP8266 como funções de redução de consumo (*low energy*) do processador e modos de suspensão de operação (*sleep modes*) para a plataforma NodeMCU.

Figura 11 – Chip NodeMCU com módulo ESP8266



Sensor de fluxo

O sensor YF-S201 será empregado para coleta de dados de vazão (instantânea em L/min e total em L). Este sensor de vazão é do tipo turbina, composto por uma válvula em formato de cata-vento que rotaciona conforme a passagem do fluido. Tal válvula rotaciona um ímã que, a partir da velocidade do rotor, gera pulsos elétricos por meio de um sensor de Efeito Hall, o quais permitem a medição da vazão de forma proporcional. A alimentação do sensor pode ser realizada com tensões entre 5 ~ 24 V e a comunicação com o NodeMCU é realizada a partir de uma saída digital de sinal, conforme ilustrado na Figura 12. Tal saída gera pulsos de tensão em um dos GPIOs do NodeMCU, os quais devem ter amplitude dimensionada de acordo com os parâmetros de tensão aceitos pelo chip. A quantidade de pulsos por unidade de tempo (pulsos/s) é proporcional à vazão do líquido na tubulação.

Figura 12: Diagrama esquemático do sensor de fluxo YF-S201.



Fonte: YFS201. Datasheet do Sensor de Fluxo de Água 1/2"

Microturbina

A microturbina utiliza o mesmo princípio de funcionamento das turbinas utilizadas em usinas hidrelétricas. O movimento do fluido, isto é, energia cinética, é responsável pelo movimento do rotor que, por sua vez, induz a geração de energia elétrica. Dessa forma, a energia gerada deve ser proporcional à vazão do fluido. O dispositivo empregado na plataforma de testes pode ser observado na Figura 13 e é um minigerador de energia hidráulico capaz de gerar de 8.8 ~ 15VCC e corrente contínua de saída entre 128 ~ 260 mA. A pressão máxima suportada é de 1,6 MPa e é construtivamente compatível com tubulação hidráulica de 20 mm de diâmetro.

Figura 13: Minigerador de energia hidráulico.



Fonte: <https://www.usinainfo.com.br/mini-geradores-de-energia/mini-gerador-de-energia-hidraulico-gerador-hidreletrico-para-projetos-88v-a-15v-dc-2902.html>

Circuito para Colheita de Energia

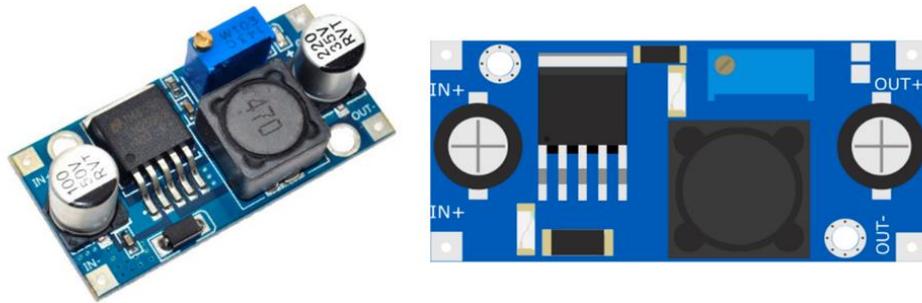
O sistema implementado, composto por reguladores de tensão, conversor de nível lógico de sinal e demais dispositivos necessários para permitir a correta alimentação do sistema, deve ser capaz de garantir o funcionamento do módulo e sensor, além de permitir a realização de testes e medições em pontos específicos. Por fim, o circuito deve garantir que o medidor possa coletar e fornecer dados mesmo se não alimentado diretamente pela microturbina. Para isso, deve ser dimensionado um sistema que assegure alimentação ininterrupta (nobreak) ao circuito, empregando, para isso, uma fonte de energia portátil (bateria) capaz de ser recarregada pela energia fornecida pela microturbina.

O segmento da solução denominado circuito de energia é o sistema responsável pelo gerenciamento da energia. O módulo ESP8266 NodeMCU opera com tensão de entrada de 4.5 ~ 9 VCC, o que exige que a tensão da microturbina seja regulada de modo a atender tais valores nominais. Além disso, é necessário que a tensão de alimentação possa ser ajustada de modo a minimizar seu consumo. O sensor YF-S201 opera com uma tensão mínima de 5 VCC, o que implica em pulsos gerados com valor de pico de até 5 VCC. O NodeMCU opera com 3.3VCC e, para garantir sinais compatíveis com o dispositivo e sem riscos de danificá-lo, é necessário a conversão dos picos dos pulsos do sensor.

O módulo responsável pela alimentação do NodeMCU é o regulador de tensão LM2596, o qual é ilustrado na Figura 14. O módulo possui uma tensão de entrada de 3.2 ~ 40 VCC e uma

tensão de saída ajustável de 1.5 ~ 35VCC, com uma eficiência de 92% (variável conforme a tensão).

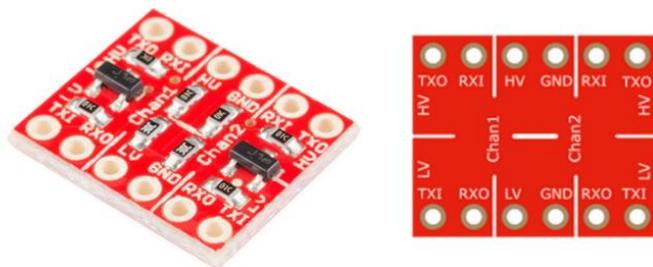
Figura 14 : Módulo regulador de tensão LM2596



Fonte: <https://www.filipeflop.com/produto/regulador-de-tensao-lm2596-conversor-dc-dc-step-down/>

Para a conexão do sensor de vazão com o NodeMCU de acordo com as especificações de tensão de ambos, tem-se o conversor de nível lógico 5 V-3.3 V, ilustrado na Figura 15. O conversor possui dois canais bidirecionais, isto é, realizam a conversão de 3.3 V para 5 V e de 5 V para 3.3 V; é composto por um divisor de tensão e um circuito baseado em MOSFET. Os pinos RXI e RXO (input/output) são do divisor de tensão e realizam a conversão de 5 V para 3.3 V de forma unidirecional, enquanto os pinos TXI e TXO são do circuito MOSFET e podem realizar a conversão nos dois sentidos.

Figura 15: Módulo conversor de nível lógico 5V-3.3V.



Fonte: <https://www.arduinoocia.com.br/2015/11/conversor-de-nivel-logico-33-5v-Arduino.html>

3.2 SOFTWARE

Dentre as tecnologias de software aplicáveis, incluem-se protocolos de comunicação de utilização na IoT, além dos aplicativos de integração das informações coletadas de

monitoramento designadas como plataformas virtuais de análise IoT que permitem agregar, visualizar, analisar e disponibilizar esses dados para manipulações posteriores.

A implementação de um dispositivo IoT com o NodeMCU possibilita a utilização de outros protocolos de comunicação, tais como o MQTT (*Message Queue Telemetry Transport*) que consiste em um protocolo de mensagens leve criado para comunicação M2M (*Machine to Machine*). Por exigir pouca capacidade de processamento e banda, este é um dos protocolos ideais para dispositivos dentro do conceito IoT. Em resumo, *publishers* enviam informação para o Broker, *subscribers* recebem informação do Broker e o Broker gerencia a troca de mensagens. Ou seja, gerenciamento da comunicação fica a cargo do Broker, deixando os sistemas livres para outras operações. O NodeMCU pode ser programado para interagir com um broker MQTT, ou seja, ele pode ser programado para ser um *publisher* ou um *subscriber* MQTT.

Nas aplicações de IoT incluem-se aplicativos para processar as informações provenientes do hardware, provendo ferramentas para análise e implementação de lógica para tomada de decisões. Tais aplicativos devem prover as funcionalidades de middleware e interfaceamento. Atualmente existem diversas ferramentas de software, sistemas operacionais e plataformas de integração para a IoT (HARVEY, 2014). A Figura 16 ilustra a utilização de IoT para exibição dos dados do medidor.

Figura 16 : Diagrama esquemático da aplicação de IoT no projeto proposto.



Fonte: Autoria Própria

Os seguintes softwares foram usados para a execução dos trabalhos propostos:

Thingier.io

Thingier.io é uma plataforma para IoT *open source* (THINGER, 2017). A plataforma permite a utilização de diversos hardwares disponíveis no mercado, abrangendo desde os mais acessíveis, como o Arduino e o ESP8266, até placas mais completas e com alta capacidade de processamento, como o Intel Edison. O Thingier.io oferece uma infraestrutura de nuvem escalável, permitindo a integração desde dispositivos em pequenos projetos até a conexão em larga escala para uso empresarial. Dessa forma, o Thingier.io mostra-se uma solução eficiente para projetos em IoT.

ThingSpeak

ThingSpeak é também uma plataforma para IoT *open source* (THINGSPEAK, 2018). Tal como o Thingier.io, a plataforma pode ser utilizada em diversos dispositivos; a ferramenta pode armazenar dados coletados através de sensores e exibi-los em dashboards e também enviar informações e comandos para os dispositivos integrados à infraestrutura de IoT, tudo através de requisições HTTP. Uma característica importante é a integração com o software de computação numérica MATLAB, da *MathWorks*, possibilitando a realização de análises bem mais abrangentes nos dados em questão sem a necessidade da aquisição de uma licença completa do software.

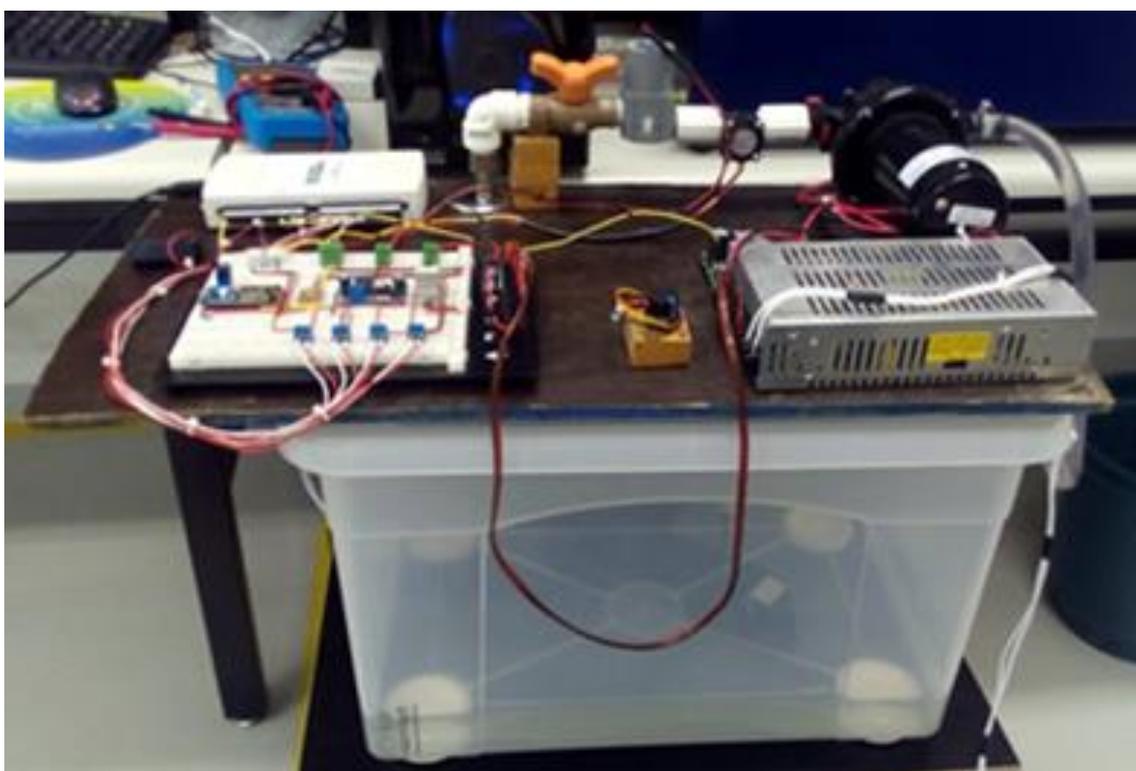
4 DESENVOLVIMENTO

4.1 PLATAFORMA DE TESTES

O foco desse projeto está na análise de viabilidade da colheita de energia hidráulica para alimentação dos dispositivos eletrônicos relacionados à medição de vazão em aplicações de IoT. Esse estudo investiga a capacidade de microgeração de energia da microturbina e se tal capacidade será suficiente para tornar a solução de medição de vazão autossustentável energeticamente (gerar sua própria energia), considerando a quantidade de energia requerida pelo microcontrolador da solução com comunicação de dados Wi-Fi.

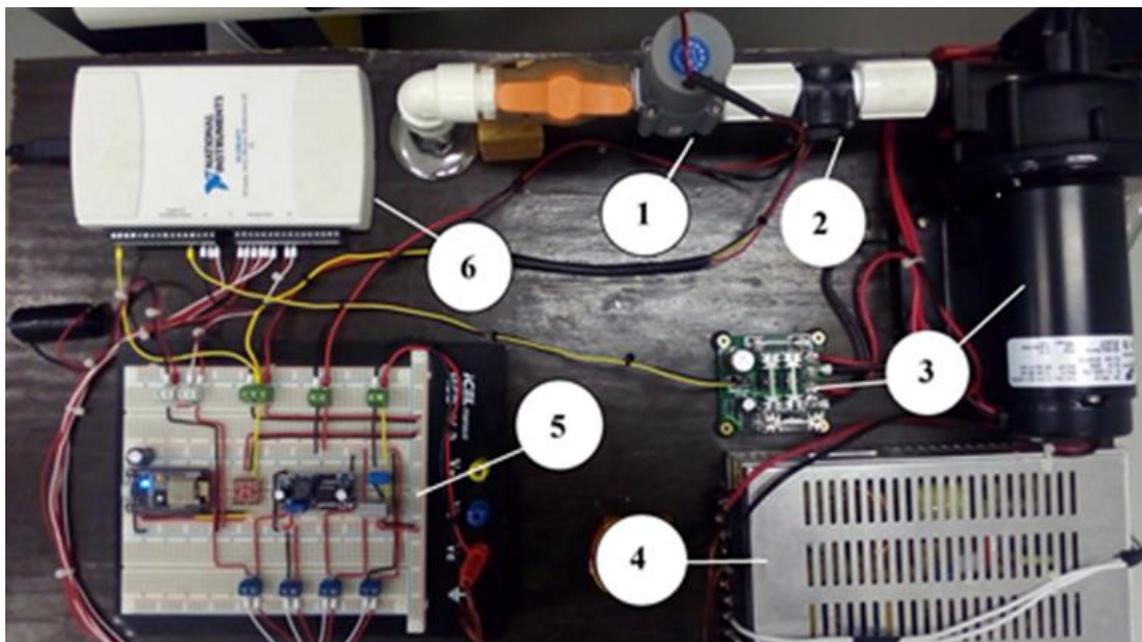
As Figuras 17 e 18 apresentam a plataforma de simulação de circulação de fluidos industriais construída. O reservatório é composto por uma caixa plástica de 40 L preenchida com cerca de 8 L de água; a água é drenada por uma bomba *Aqua Jet WPS 3.5*, da *Johnson Pump*. A bomba é capaz de gerar uma vazão máxima de 13 L/min e pode ser alimentada com 12 VCC ou 24 VCC, drenando cerca de 2.5 A e 1.1 A, respectivamente. É alimentada a partir de uma fonte de corrente contínua S-201-24, da *PWM Steady*, que fornece 24 VCC e até 8.3 A. A fonte é conectada à rede elétrica com tensão de 127/220 VAC.

Figura 17: Visão Geral da Plataforma de Testes



Fonte: Autoria Própria

Figura 18: Descrição funcional dos componentes da plataforma



Fonte: Autoria Própria

Os componentes e circuitos eletrônicos de montagem da plataforma de testes estão montados e identificados conforme a seguir:

1. Micro hidrogerador (microturbina), tensão de 8,8 a 15 VCC em carga, corrente de saída de 128 a 260 mA, pressão máxima 1,6 Mpa;
2. Sensor de Vazão tipo Efeito Hall, faixa de vazão de 1 a 30 L/min., tensão de operação de 5 a 24 VCC, pressão máxima de 2 Mpa;
3. Bomba de pressurização de vazão nominal de 13 L/min que utiliza um circuito eletrônico controlador de velocidade a partir de tensão de 0 a 5 VCC, e é alimentada em 12V ou 24V, consumo de 2,5 A ou 1,1 A respectivamente;
4. Fonte de alimentação auxiliar de 24 VCC para alimentação bomba pressurizadora, modelo S-201-24, da PWM Steady, que fornece 24 V e até 8.3 A. A fonte é conectada à rede elétrica com tensão de 220 VRMS;
5. Circuito eletrônico principal para colheita de energia do mini hidrogerador, conversores de tensão e sinal de vazão, microcontrolador programável NodeMCU, resistores shunts de 1 Ohm de medição para DAQ LabVIEW e acoplamento de conjunto de baterias auxiliares;

6. Módulo DAQ LabVIEW de entradas/saídas digital/analógica para interfaceamento de dados do circuito principal.

A tubulação industrial é simulada a partir de junções rosqueadas de PVC que conectam a bomba ao sensor de vazão YF-S201, o qual é ligado ao microgerador hidráulico e, por fim, a um registro de $\frac{1}{4}$ de volta. A água drenada realimenta o reservatório, mantendo um volume de fluido constante no sistema. O controle da rotação do motor da bomba é realizado a partir da variação da tensão entre 0 ~ 5 VCC em um módulo regulador para controle de motores de 24 VCC. Os testes iniciais foram efetuados a partir de um potenciômetro e, mais adiante no desenvolvimento do projeto, o componente foi substituído por um sinal de saída programado do DAQ USB-6211.

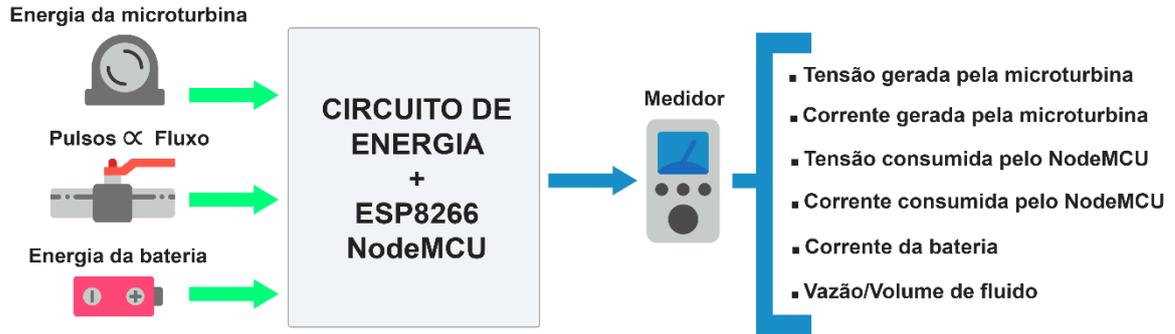
A microgeração é baseada na movimentação de fluidos em circulação nos processos industriais. A microturbina é um modelo reduzido de um gerador de corrente contínua e seu princípio de funcionamento é baseado no emprego de energia mecânica para geração de energia elétrica: a passagem de fluido rotaciona o enrolamento de armadura (rotor) do micro gerador; uma vez que o enrolamento está imerso em campo magnético, há o surgimento de uma tensão induzida em seus terminais que, se conectados a uma carga (no caso, o NodeMCU e o sistema de baterias) fornecem corrente elétrica. Assim, quanto maior a velocidade de rotação, maior a tensão gerada; para o sistema proposto, quanto maior a vazão de água pelo encanamento, mais energia mecânica é convertida.

4.2 DEFINIÇÃO DAS VARIÁVEIS DE INTERESSE

A análise da viabilidade da solução empregada só é possível após a coleta dos dados necessários para a realização de cálculos que permitam mensurar a energia gerada e consumida pelo sistema proposto. Os dados capazes de fornecer tais informações acerca da solução são ilustrados na Figura 19, na qual é possível observar, também, os periféricos empregados no sistema. Dessa forma, além de realizar a regulação e comunicação entre os periféricos e o dispositivo microcontrolado, permita que tais variáveis sejam facilmente medidas, isto é, estejam construtivamente acessíveis para acoplamento de instrumentos de medição.

A definição das variáveis necessárias para análise foi realizada de acordo com um parâmetro de geração e consumo energéticos passível de comparação; o parâmetro escolhido foi a potência ativa, que representa a energia gerada/consumida em determinada unidade de tempo. A potência elétrica é obtida a partir da tensão e corrente gerada/consumida pelo sistema.

Figura 19: Variáveis de interesse para análise dos resultados.

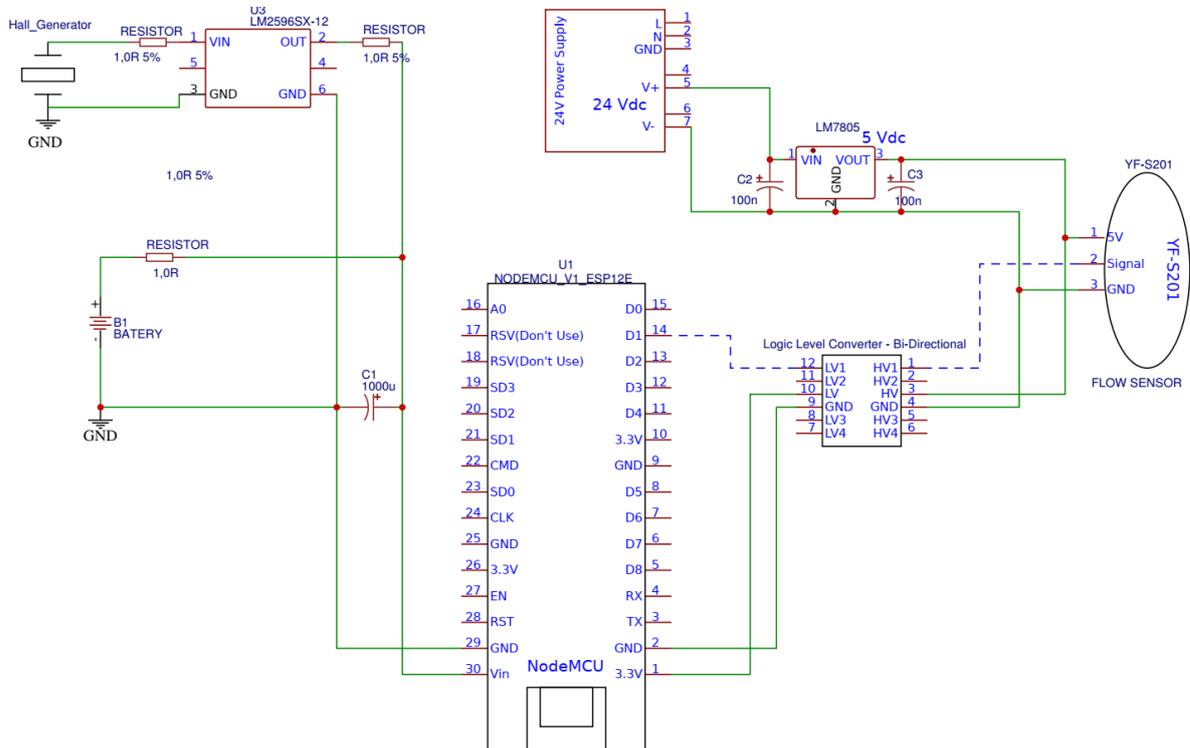


Fonte: Autoria Própria

4.3 PROJETO DO CIRCUITO DE ENERGIA

A Figura 20 ilustra o diagrama esquemático do circuito projetado para realizar o gerenciamento da energia gerada e consumida e o ajuste dos níveis de tensão para o sensor de vazão e o módulo NodeMCU. Um regulador de tensão de 24/5 VCC foi construído para garantir 5 VCC fixo através de um circuito integrado 7805 e dois capacitores de 100 nF.

Figura 20: Diagrama esquemático do circuito de energia.



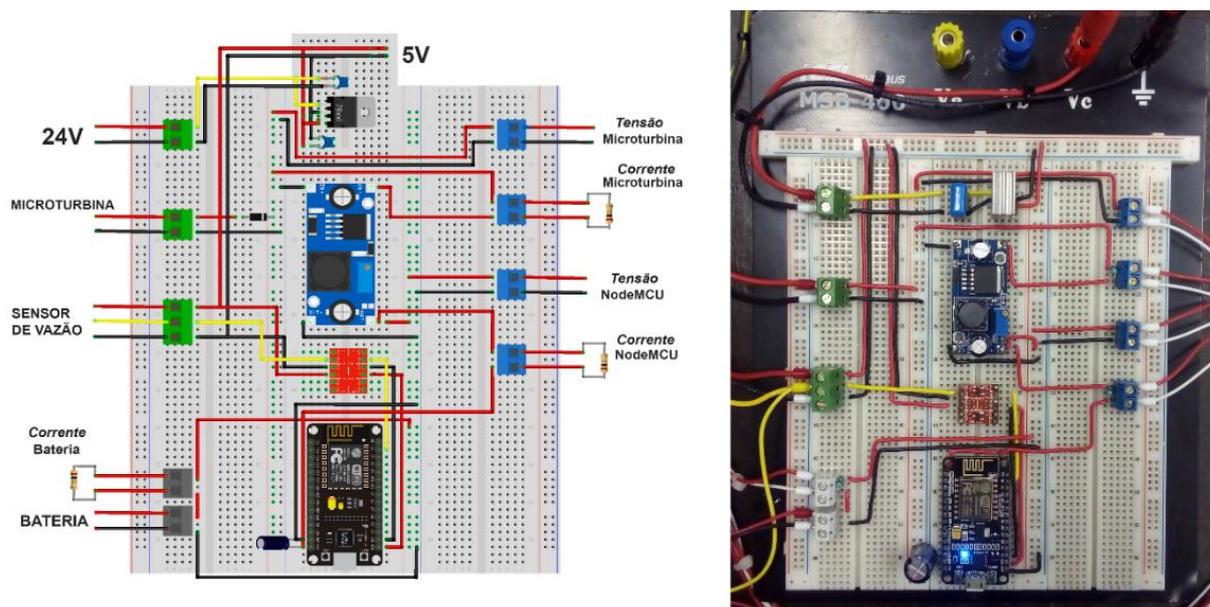
Fonte: Autoria Própria

O nível fixo de 5 VCC alimenta o sensor de vazão e é referência para o conversor de nível lógico no pino HV. O conversor recebe a saída de sinal do sensor YF-S201 como entrada do pino RXI, convertendo-o para o nível de 3.3 VCC no pino RXO, o qual é lido pelo pino D1 do ESP8266 NodeMCU, correspondente ao GPIO5. Além disso, o nível de 3.3 VCC do conversor é referenciado no pino LV pela saída de 3.3 VCC do NodeMCU.

A tensão gerada pela microturbina é regulada pelo módulo LM2596; os pinos de saída do módulo fornecem, em paralelo, alimentação ao NodeMCU e realizam a carga da bateria responsável pela função nobreak. O sistema de nobreak é composto por uma bateria de lítio-polímero 18650 com 2600 mAh capaz de fornecer entre 3.6 ~ 4 VCC ao NodeMCU. A Figura 21 apresenta o diagrama ilustrativo das variáveis de interesse e níveis de tensão no circuito, bem como a montagem do circuito em protoboard.

As entradas e saídas do circuito foram construídas empregando bornes KRE com parafusos, o que mantém o circuito fixo e ao mesmo tempo flexível para modificações. O capacitor de 1000 uF conectado em paralelo à entrada de alimentação do NodeMCU tem função de reduzir as variações de tensão. Por fim, nota-se que foram utilizados resistores intermediando a alimentação entre microturbina-LM2596, entre LM2596-NodeMCU e entre LM2596-nobreak. Tais resistores têm valor nominal de 1 Ω e exatidão de 1% e são empregados com função *shunt* para a medição de corrente pelo dispositivo de aquisição de dados.

Figura 21: Diagrama ilustrativo e montagem em protoboard do circuito de energia.



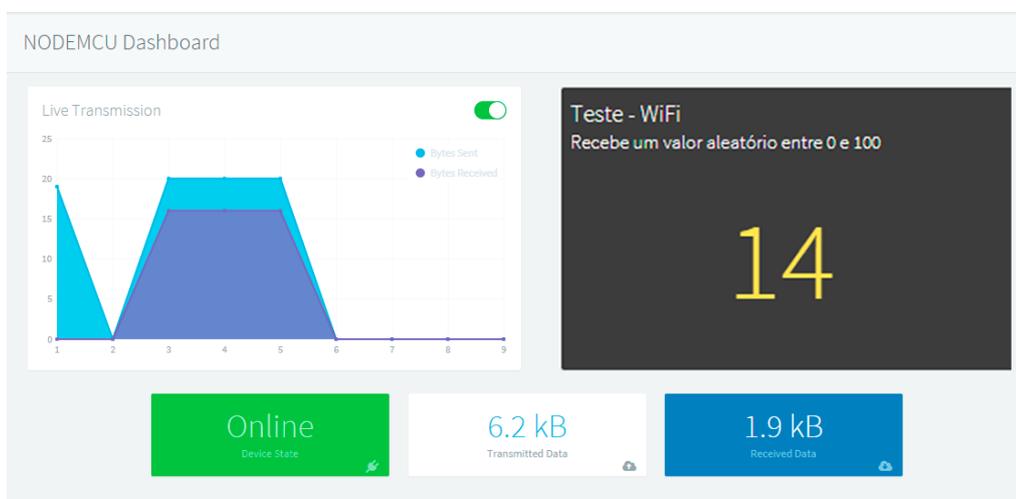
4.4 PROGRAMAÇÃO DO NODEMCU E DA INTERFACE VISUAL

A programação para contagem de pulsos e cálculo de vazão foi realizada na mesma linguagem utilizada nos microcontroladores Arduino, baseada em linguagem C. A lógica do programa consiste na contagem de pulsos lidos do sensor YF-S201 e um ganho determinado no datasheet do sensor converte tais pulsos em valores de vazão (em L/min) e volume (em L).

Uma interrupção ativa em borda de subida foi configurada para o pino D1 do NodeMCU, responsável pela leitura dos pulsos do sensor. Os pulsos são contabilizados e, a cada segundo, o somatório de pulsos é multiplicado por um fator de $(1/7,5)$, convertendo o valor em Hz para vazão em L/min com aproximadamente 3% de exatidão. A número de pulsos acumulados durante a execução do programa é multiplicado por um fator de $(1/450)$, uma vez que o sensor gera cerca de 450 pulsos por litro.

Para o início das medições foi programado, também, um algoritmo simples que conecta o NodeMCU ao Wi-Fi e transmite, através da plataforma Thinger.io, um número aleatório a cada um segundo. O algoritmo foi adaptado de um código-exemplo da biblioteca do Thinger para o ESP8266, onde inicialmente é realizada a configuração para conexão com a plataforma, com as credenciais da rede Wi-Fi a qual o NodeMCU irá se conectar e, por fim, uma função de atribuição realiza a transmissão dos dados. Os dados transmitidos são visualizados na seção da plataforma denominada Dashboard e a quantidade de bytes transmitidos/recebidos pode ser acompanhada em tempo real; ambas informações estão ilustradas na Figura 22.

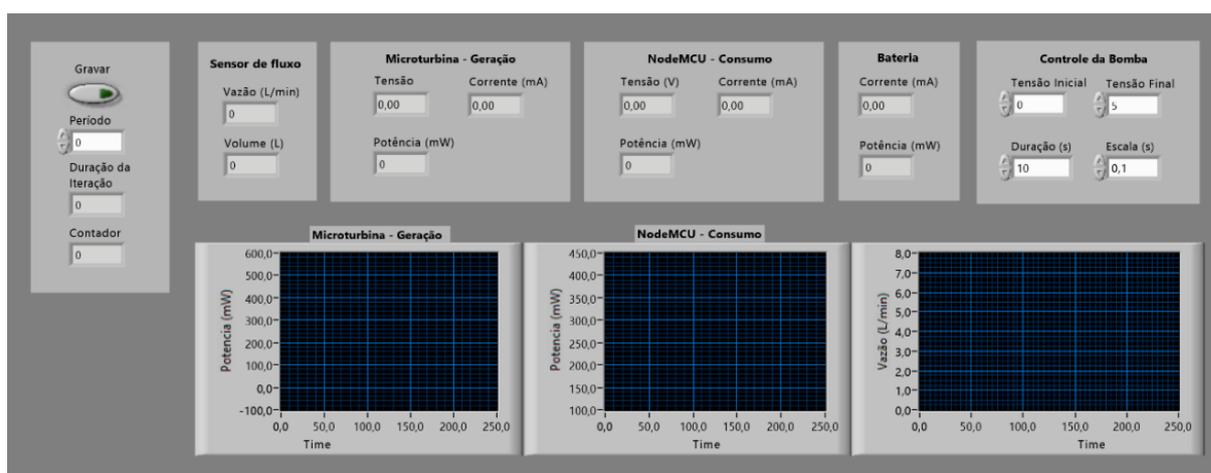
Figura 22: Testes iniciais de comunicação Wi-Fi com o Thinger.io.



Fonte: Autoria Própria

A Figura 23 apresenta o painel do instrumento virtual desenvolvido no software LabVIEW. O painel é composto por quatro gráficos que permitem acompanhar o histórico temporal das medições de maneira visual; os dados lidos a cada segundo podem ser observados na parte superior do painel, dividido em quatro seções referentes aos dados: do sensor de vazão, lidos pelo contador do dispositivo de aquisição; da geração de energia (tensão e corrente) pela microturbina, lidos a partir de portas analógicas de forma diferencial; do consumo de energia pelo NodeMCU e da corrente consumida/fornecida pela bateria, ambos os dados lidos também de maneira diferencial por portas analógicas do DAQ USB-6211. A seção no canto superior direito permite a entrada de dados para configuração de um sinal de rampa de 0 ~ 5 VCC em um intervalo de tempo variável na placa responsável pelo controle do motor da bomba hidráulica.

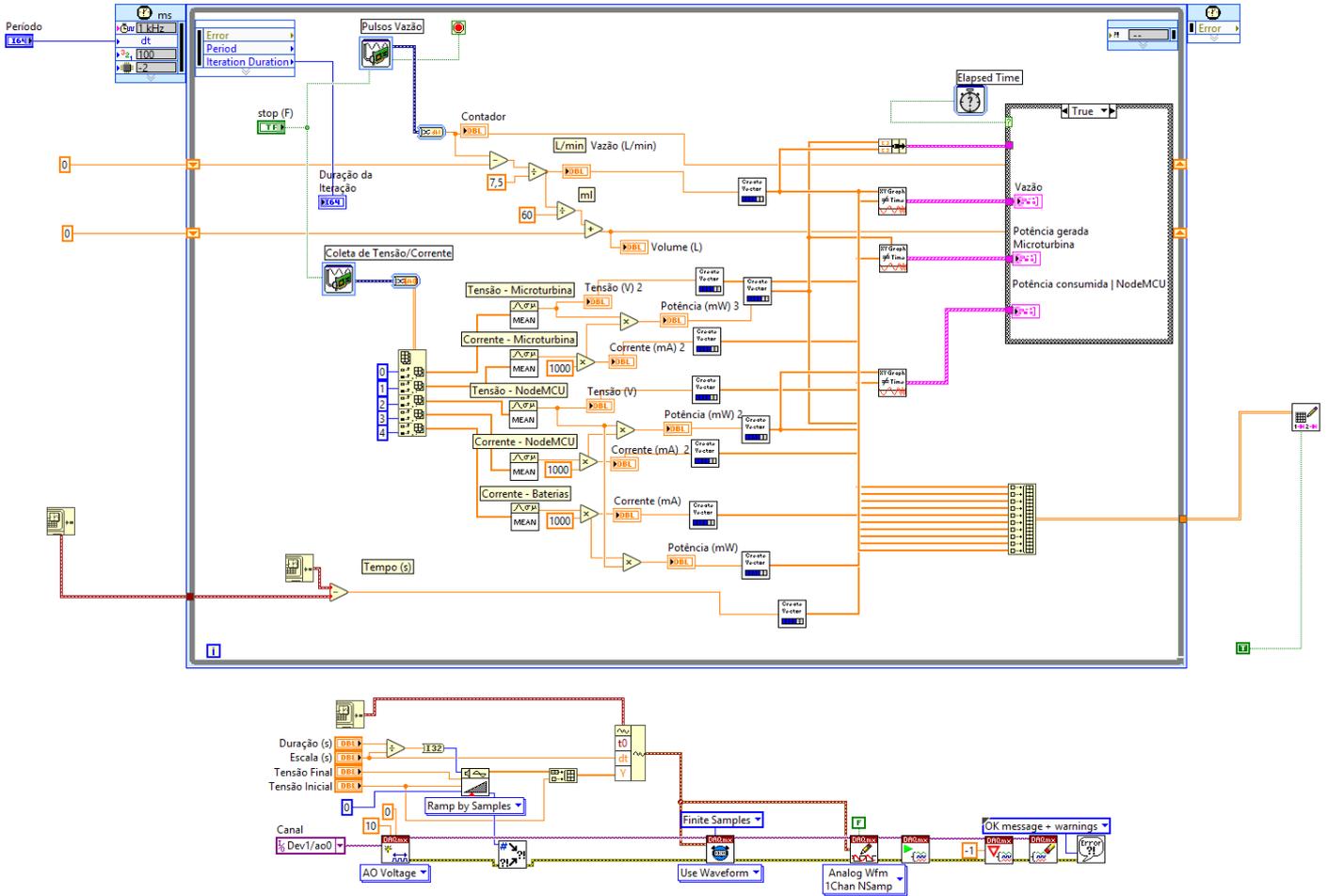
Figura 23: Painel do instrumento virtual no software LabVIEW



Fonte: Autoria Própria

Tem-se, na Figura 24, a programação empregada para a realização das medições e cálculos referentes à vazão e potência. Os dados são lidos continuamente e as potências gerada e consumida são calculadas a partir do valor médio do produto da tensão pela corrente (potência instantânea); alguns ganhos são empregados para realizar a conversão de unidades de modo a garantir ordens de grandezas compatíveis com as leituras. Os valores resultantes são exibidos no painel e empregados na construção dos gráficos. Além disso, os dados são exportados de forma tabelada em arquivo de texto, permitindo análise posterior a partir de *softwares* específicos para tratamento de dados.

Figura 24: Diagrama de blocos do instrumento virtual no software LabVIEW



Fonte: Autoria Própria

4.5 MODOS DE BAIXO CONSUMO DE ENERGIA

O módulo ESP8266 NodeMCU apresenta alguns modos de operação de baixo consumo energético. Para operar em tais modos, uma ativação deve ser efetuada através do algoritmo implementado na placa. A redução do consumo ocorre, pois, tais modos desabilitam funções específicas do hardware do dispositivo. Existem três modos de baixa potência (*sleep modes*) no módulo ESP8266 NodeMCU:

- *Modem Sleep* – o ESP8266 desliga o Wi-Fi em intervalos de tempo definidos. A CPU permanece ativa, permitindo a realização de operações locais, como coleta e tratamento de dados, por exemplo. O consumo de corrente quando operando nesse modo é da ordem de 15mA (*datasheet*).

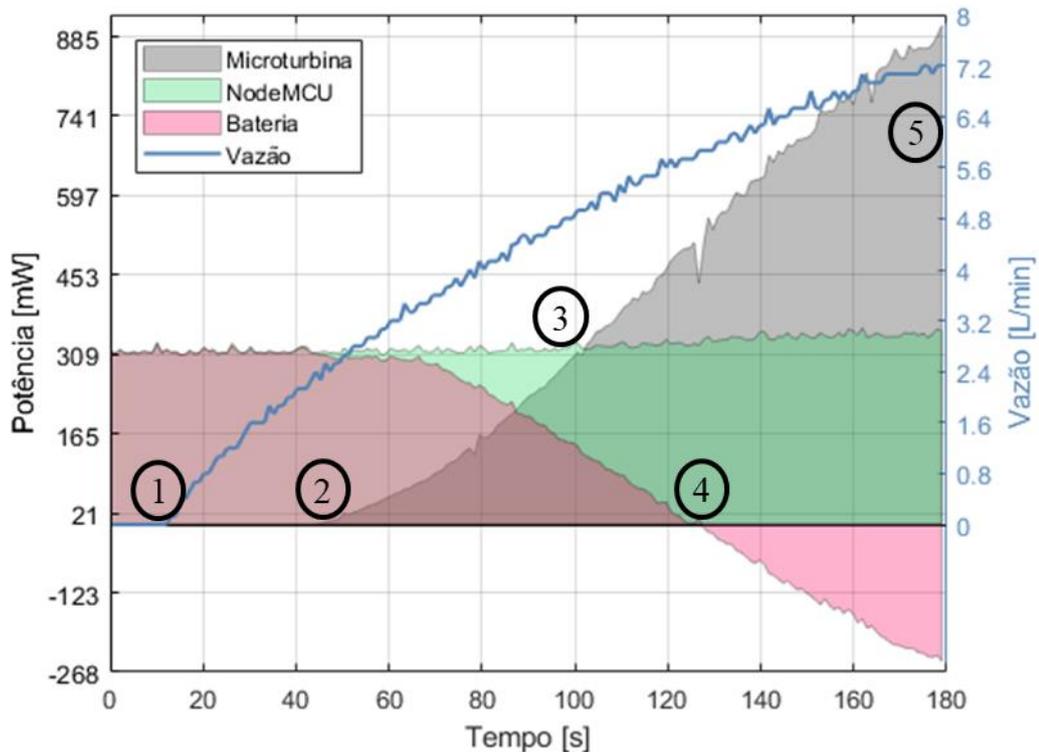
- *Light Sleep* – o Wi-Fi é desligado e a CPU entra em modo de suspensão, não sendo possível realizar operações. O módulo pode ser acordado através de um sinal externo para um pino definido em *software*. O consumo de corrente quando operando nesse modo é da ordem de $400\mu\text{A}$ (*datasheet*).
- *Deep Sleep* – praticamente todo o módulo fica desligado: Wi-Fi, CPU, *clock* – apenas o RTC (*Real Time Clock*) permanece ativado, uma vez que este é o responsável por acordar o dispositivo. O consumo médio de corrente é da ordem de $20\mu\text{A}$ (*datasheet*).

5 ANÁLISE DO POTENCIAL DE COLHEITA DE ENERGIA

5.1 ENSAIOS DE VERIFICAÇÃO DO POTENCIAL DE COLHEITA DE ENERGIA

O sistema desenvolvido foi submetido a diversos testes iniciais e sua resposta foi verificada pelos dados obtidos no software LabVIEW. Os ensaios foram sistematizados de modo a coletar dados ao longo de intervalo de 180 s a uma taxa de 1 kHz, os quais foram tabelados e convertidos em gráficos que ilustram a potência consumida pelo módulo ESP8266 NodeMCU, a potência fornecida pela microturbina a partir da geração hidrelétrica e a potência gerada e consumida pela bateria do sistema de nobreak. Os testes foram realizados com o NodeMCU transmitindo dados via Wi-Fi para a plataforma Thinger.io e alimentado em conjunto pela bateria do sistema de nobreak e energia gerada pela microturbina. O comportamento do sistema de acordo com tais parâmetros podem ser observado na Figura 25.

Figura 25: Gráfico da potência gerada, consumida e vazão do sistema



Fonte: Autoria Própria

Observa-se os seguintes pontos em destaque na Figura 25:

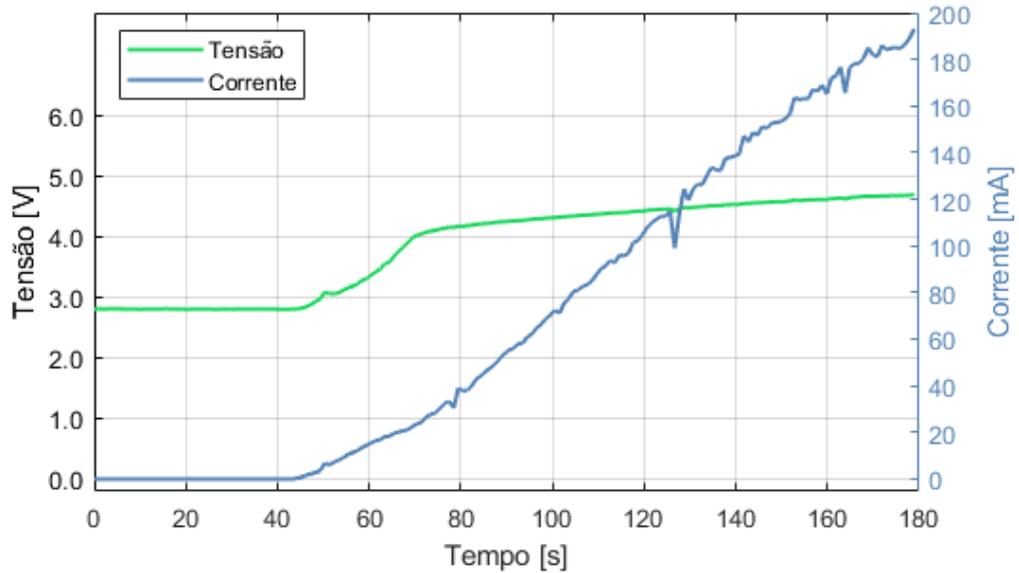
1. Em 10 segundos inicia-se a rampa de aumento gradual da vazão, contudo sem geração de energia pela micro turbina;
2. Em aproximadamente 40 segundos inicia-se a geração de energia pela microturbina;
3. Em aproximadamente 100 segundos a geração de energia pela micro turbina supre totalmente a energia consumida pelo dispositivo NodeMCU, com uma vazão aproximada de 4,9 L/min. Porém devido às perdas nos circuitos auxiliares, a bateria mantém-se suprindo um residual de energia.
4. Em aproximadamente 125 segundos a bateria para seu fornecimento de energia e passa a ser carregada pela microturbina. Neste instante a vazão é de aproximadamente 5,8 L/min.
5. Neste ponto é alcançada a máxima geração de energia de 890 mW e com vazão de aproximadamente 7,2 L/min.

A vazão da bomba aumentou gradualmente até um valor máximo de aproximadamente 7,2 L/min, com uma zona inicial em que o sensor YF-S201 não é capaz de realizar a coleta dos dados de vazão, pois esta possui valores muito baixos e inferiores ao mínimo requerido pelo hardware. Conforme esperado, a vazão aumenta muito semelhante a uma rampa e, aproximadamente nos 40 s iniciais, a vazão inferior a 2,5 L/min não é capaz de gerar energia através da microturbina.

Nesse período inicial, referente a uma “zona morta” para a geração de energia, o NodeMCU é alimentado integralmente por energia oriunda da bateria do sistema de nobreak. Porém, no momento em que a microturbina passa a gerar energia, a potência consumida da bateria diminui gradativamente conforme a vazão aumenta e, em um valor de cerca de 4,4 L/min, a microturbina passa a ser a fonte de alimentação principal para o NodeMCU. Quando a vazão atinge valores próximos a 5,4 L/min, a bateria para de fornecer energia ao NodeMCU e este passa a ser alimentado integralmente pela energia proveniente de microgeração com a energia excedente sendo é direcionada a recarga da bateria.

A Figura 26 apresenta o comportamento para as variáveis de tensão e corrente fornecidos pela microturbina durante os 180 s de um dos testes realizados. É possível verificar uma tensão máxima de aproximadamente 4,7 VCC e uma corrente de aproximadamente 200 mA fornecida pela microturbina. Portanto, verifica-se que a microturbina é capaz de gerar em torno de 1 W de energia elétrica nas configurações e condições de uso estabelecidas no experimento.

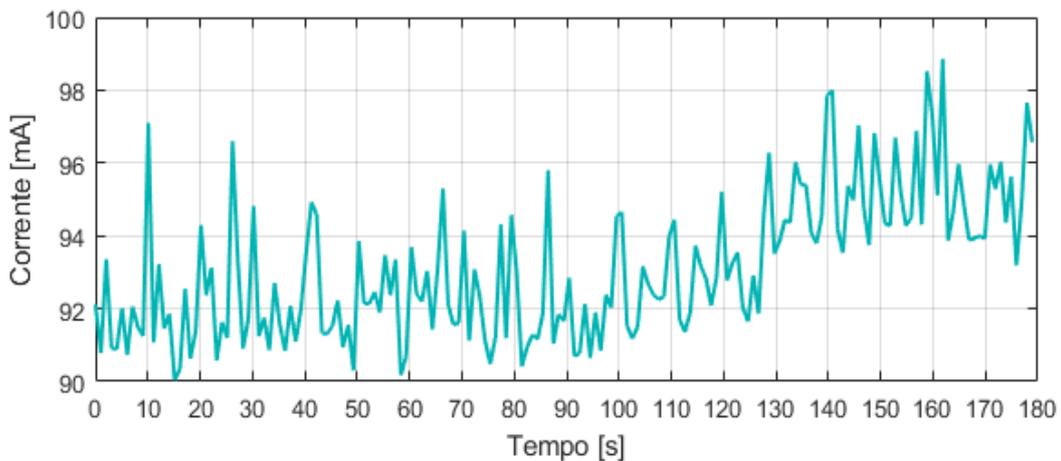
Figura 26: Gráfico de tensão terminal e corrente fornecida pela microturbina



Fonte: Autoria Própria

A Figura 27 apresenta o comportamento da corrente consumida pelo NodeMCU. Observa-se um valor médio de aproximadamente 93 mA e diversos picos de consumo ocasionados pela transmissão de dados via Wi-Fi para a plataforma Thingier.io. Um aumento no consumo de corrente ocorre devido à geração de energia pela microturbina no estágio final do teste.

Figura 27: Gráfico da corrente consumida nos picos devido à comunicação.



Fonte: Autoria Própria

A comprovação da viabilidade da solução pode ser efetuada a partir da metodologia de testes proposta. A microturbina tem potencial de geração energética para sustentar o sistema de

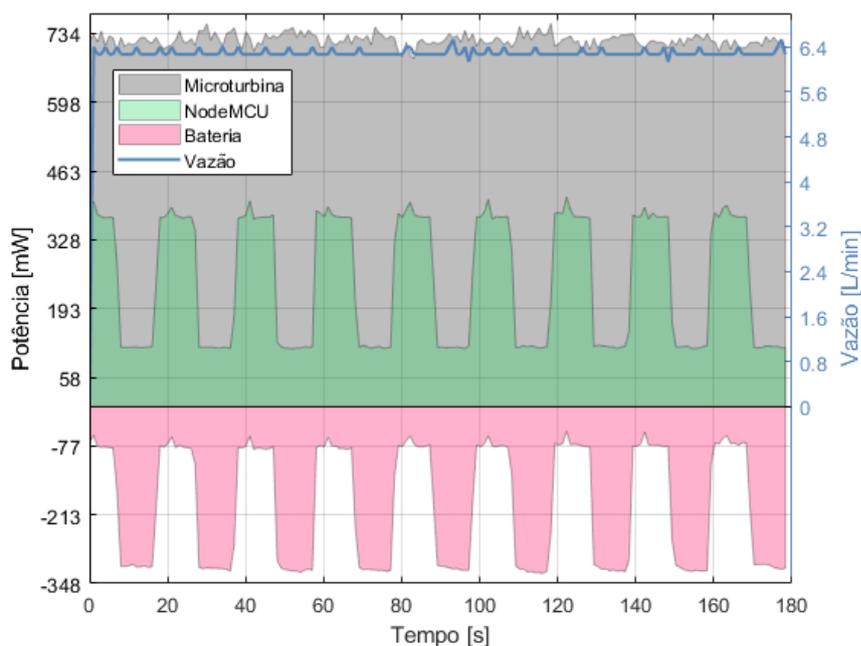
medição de vazão e integração deste à IoT Industrial. É importante, porém, buscar otimizar a solução a partir da diminuição do consumo energético do NodeMCU.

5.2 REDUÇÃO DO CONSUMO DE ENERGIA - OPERAÇÃO EM MODEM SLEEP

Para a solução desenvolvida de medição de vazão autossustentável, o único modo de economia passível de ser utilizado é o *Modem Sleep*. O medidor de vazão deve coletar dados de maneira ininterrupta, uma vez que a resultante do montante volumétrico escoado através da tubulação é calculada de maneira gradual. Os modos *Light Sleep* e o *Deep Sleep* inabilitam a utilização da CPU, impedindo a realização de operações pelo microcontrolador. A metodologia de testes empregada, neste momento, utiliza também comunicação através da plataforma em nuvem *Thinger.io*.

O algoritmo comunica-se com o *Thinger.io* durante 10 s e então entra em modo de economia de energia *Modem Sleep*. O módulo permanece dormindo por 10 s e então conecta-se novamente à rede Wi-Fi e volta a realizar comunicação com a plataforma em nuvem. O perfil de geração-consumo obtido para este teste é ilustrado na Figura 28.

Figura 28 : Gráfico da potência gerada, consumida e vazão obtido em ensaio experimental

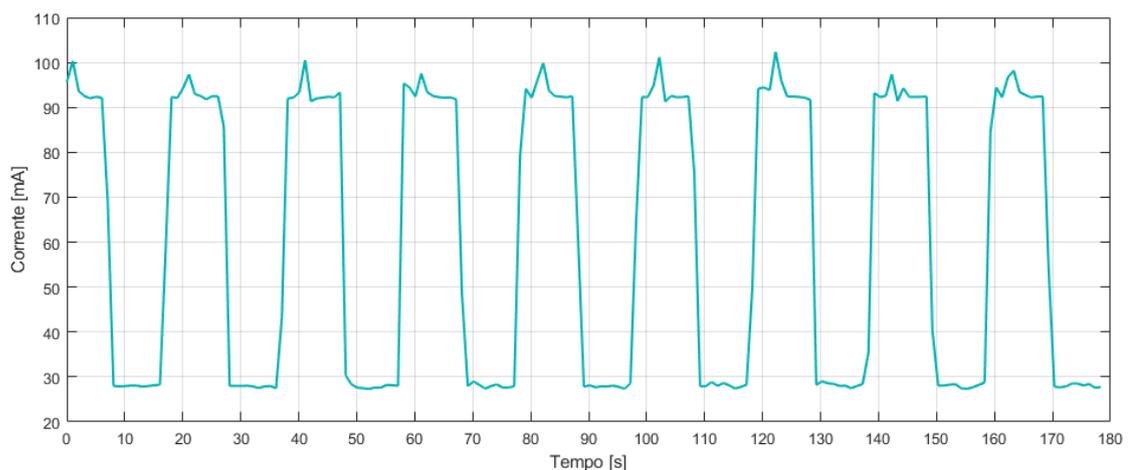


Fonte: Autoria Própria

Para a realização dos ensaios em economia de energia, a vazão da bomba foi mantida em torno de 6,3 L/min – valor no qual a microturbina já é capaz de sustentar concomitantemente a alimentação do módulo ESP8266 NodeMCU e recarregar a bateria do sistema de nobreak. Como observado, a potência gerada pela microturbina nesse valor de vazão varia em torno de 700 mW. Através do resultado obtido conforme Figura 28, é possível verificar a queda de consumo energético a cada 10 s; durante esse período, a energia sobressalente gerada pela microturbina é redirecionada para recarregar o sistema de nobreak. Tal fato indica uma melhoria na eficiência do sistema, uma vez que a manutenção do sistema de nobreak é realizada de maneira mais intensiva em intervalos de tempo cujo circuito de comunicação não é utilizado.

Para testes realizados nas mesmas condições, porém sem utilização do *Modem Sleep*, a potência média consumida pelo módulo ESP8266 NodeMCU é de 329,21 mW. Para os testes efetuados com *Modem Sleep*, tal potência é de 248,09 mW, o que implica em 81,12 mW de energia economizados pelo NodeMCU e armazenados na forma de recarga na bateria. Tal valor corresponde a 24,6% da potência média consumida sem a utilização do *Modem Sleep*. A redução do consumo de corrente pelo módulo também foi verificada: valor mínimo de 27,3 mA e um valor médio de 61,2 mA quando comparado ao teste sem o modo de economia com valor mínimo de 83,3 mA, com uma média de 84,2 mA. O comportamento da corrente para o teste de economia de energia é ilustrado na Figura 29.

Figura 29: Gráfico da corrente consumida nos picos devido à comunicação.



Fonte: Autoria Própria

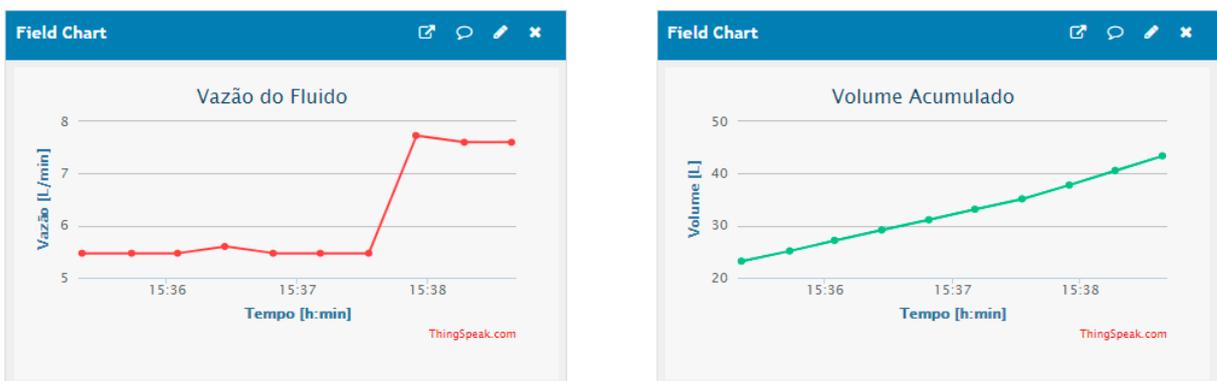
5.3 VERIFICAÇÃO DO CONSUMO DE ENERGIA COM COMUNICAÇÃO IOT

Até o momento em questão, não há integração entre o algoritmo desenvolvido para a comunicação com o sensor de fluxo YF-S201 e o algoritmo desenvolvido para os primeiros testes de comunicação Wi-Fi. Nesse sentido, buscou-se elaborar um algoritmo capaz de realizar a coleta de dados de forma ininterrupta, realizar a cada 1 s o cálculo da vazão instantânea e do volume acumulado e então enviar tais dados para a plataforma de Internet das Coisas (IoT).

Optou-se, então, pela comunicação com a plataforma *ThingSpeak*. Além da integração direta com o MATLAB, a documentação referente a utilização da plataforma é mais completa e permite a utilização de requisições HTTP sem a necessidade de utilizar uma biblioteca específica para a plataforma, flexibilizando a manutenção do algoritmo.

O algoritmo desenvolvido respondeu muito bem aos testes efetuados e a dashboard construída na plataforma *ThingSpeak* pode ser observada na Figura 30.

Figura 30: Dashboard construída na plataforma ThingSpeak: vazão (L/min) e Volume acumulado (L)



Fonte: Autoria Própria

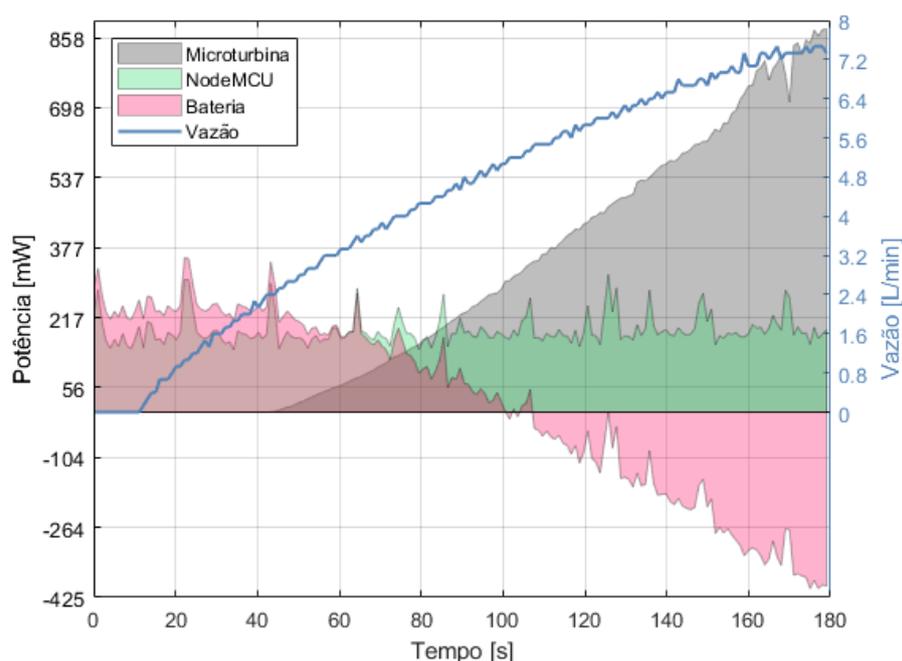
Foram desenvolvidas duas versões para o código em questão. Na primeira, o módulo ESP8266 NodeMCU permanece constantemente conectado à rede Wi-Fi, realiza a coleta ininterrupta de dados do sensor YF-S201 e, em intervalos de tempo pré-determinados, informações são enviadas ao ThingSpeak. Na segunda versão, o módulo ESP8266 NodeMCU

coleta os dados do sensor de fluxo de maneira ininterrupta enquanto permanece em *Modem Sleep*, isto é, com o Wi-Fi desativado, e, em intervalos de tempo pré-determinados, é realizado o wakeup do módulo, que se conecta à rede Wi-Fi e envia as informações ao *ThingSpeak*.

Assim pode-se realizar a análise comparativa da redução de consumo energético do módulo para a utilização da configuração de baixa potência *Modem Sleep* e, alterando o tamanho do intervalo de transmissão dos dados, verificar a redução do consumo médio de potência pelo módulo. A plataforma *ThingSpeak* possui uma taxa de recepção de dados limitada em um dado a cada 15 s. Assim, o primeiro ensaio realizado consiste na transmissão de dados de vazão instantânea e volume na taxa mínima de comunicação da plataforma.

Na Figura 31 apresenta o resultado obtido para o ensaio de 180 s com aplicação de uma variação de 0 ~ 5VCC em forma de rampa no circuito de controle de vazão da bomba. O módulo ESP8266 NodeMCU permanece conectado à rede Wi-Fi durante todo o ensaio, isto é, sem entrar em modo de baixo consumo de potência.

Figura 31: Potência gerada e consumida e vazão no sistema para dados a cada 15 s via rede Wi-Fi.



Fonte: Autoria Própria

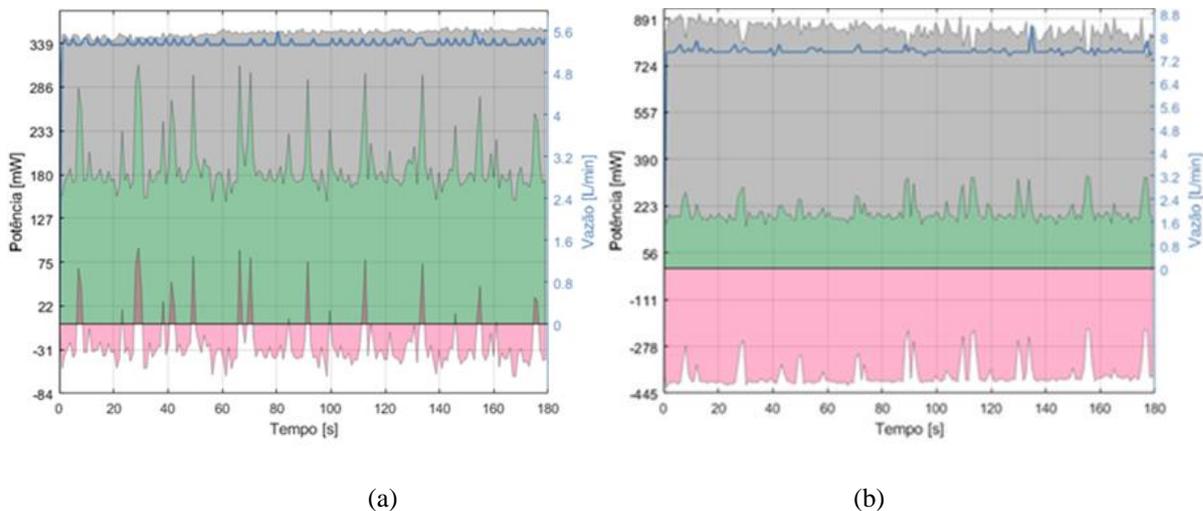
O consumo médio de potência pelo módulo ESP8266 NodeMCU é de 200 mW para a configuração em questão. Observa-se que a microturbina passa a gerar energia em

aproximadamente 44,27 s, o que corresponde a uma vazão de 2,4 L/min. Para a mesma configuração de transmissão, tem-se na Figura 32(a) a resposta do sistema para uma vazão fixa em aproximadamente 5,34 L/min e na Figura 32(b) a resposta do sistema para a vazão máxima passível de ser aplicada à plataforma de testes construída.

É importante verificar, na Figura 32(a), que a bateria está sendo carregada em grande parte do ensaio realizado; nos picos de transmissão de comunicação, a microturbina não é capaz de alimentar completamente o módulo e, nesses momentos, observa-se solicitação de potência da bateria para o módulo. Apesar da potência gerada pela microturbina ser superior à exigida pelo NodeMCU durante todo o ensaio, o gráfico não contabiliza as perdas energéticas no circuito de energia dimensionado.

Para a Figura 32(b) observa-se a geração média de 853,56 mW em vazão máxima, energia a qual é suficiente para manter a bateria em modo de carregamento durante todo o ensaio, fornecendo a ela uma potência média de aproximadamente 378,84 mW.

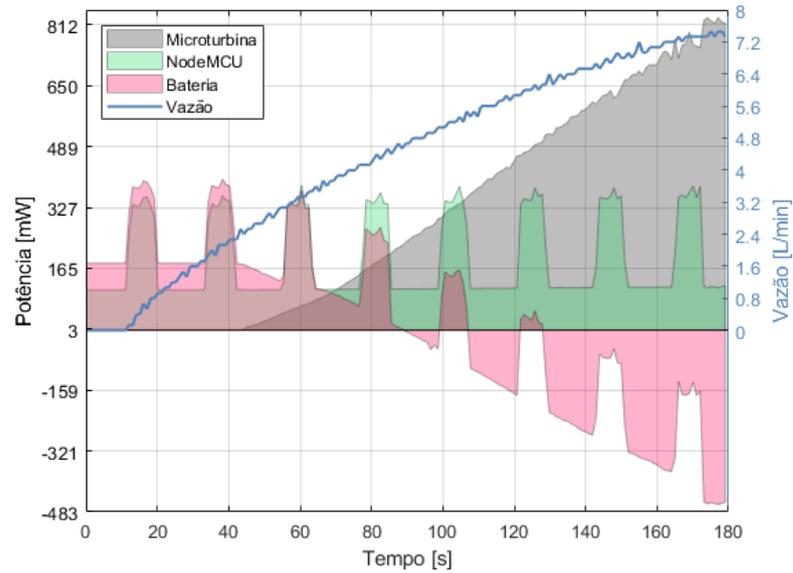
Figura 32: Potência gerada, consumida em (a) aproximadamente 5,34 L/min e (b) aproximadamente 7,46 L/min.



Fonte: Autoria Própria

A Figura 33 apresenta o comportamento do sistema para a execução do algoritmo que utiliza o modo de baixo consumo *Modem Sleep*. A transmissão ocorre, tal como anteriormente, a cada 15 s; porém, durante os intervalos de transmissão o Wi-Fi é desligado. Tem-se um consumo médio pelo módulo ESP8266 NodeMCU de 194,25 mW e a microturbina passa a gerar energia em 43,31 s, cuja vazão correspondente é de 2,53 L/min.

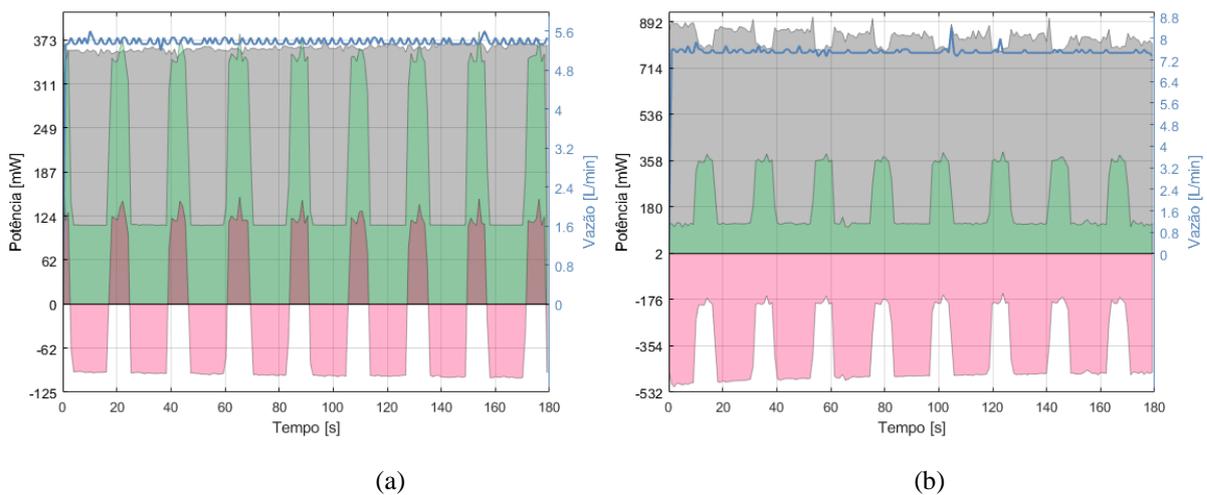
Figura 33 : Potência gerada e consumida no sistema para dados a cada 15s e *Modem Sleep*.



Fonte: Autoria Própria

Tem-se na Figura 34(a) a resposta do sistema na configuração em questão para uma vazão média de 5,35 L/min e na Figura 34(b) a resposta do sistema em vazão máxima.

Figura 34 : Potência gerada, consumida: (a) aproximadamente 5,35 L/min e (b) aproximadamente 7,45 L/min.



(a)

(b)

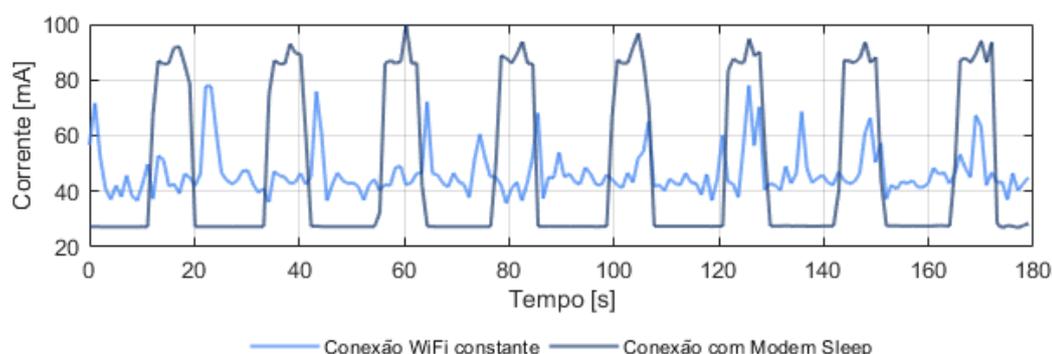
Fonte: Autoria Própria

É possível observar de forma mais clara, na Figura 34(a), a alternância entre o fornecimento e solicitação de potência pela bateria, cujo valor médio é de 21,11 mW no sentido negativo, isto é, a bateria mantém-se, na média, tendo sua carga carregada pela energia da microturbina. Já na Figura 34(b), verifica-se um fenômeno anteriormente não constatado: a potência gerada pela microturbina varia conforme o módulo ESP8266 NodeMCU solicita potência; tal fenômeno será tratado mais à frente.

O resultado esperado para os ensaios descritos é a diminuição da potência média consumida pelo módulo ESP8266 NodeMCU quando utilizado o modo de baixo consumo *Modem Sleep* em comparação com o algoritmo que mantém o módulo conectado ininterruptamente à rede Wi-Fi. A potência média obtida experimentalmente para ambos os casos foi, porém, bastante similar – o modo de baixo consumo apresentou um valor 0.62% maior em relação ao algoritmo de conexão constante.

Tal fato pode ser explicado pelos picos de maior intensidade na potência consumida pelo NodeMCU observados na Figura 33, referente ao modo de baixo consumo, em comparação aos picos observados na Figura 31. O valor máximo de consumo de potência do módulo para o caso de conexão constante é de 315,30 mW, enquanto para a configuração de *Modem Sleep*, tem-se um consumo máximo de 386,09 mW. Assim, verifica-se que há um esforço maior de corrente ao reiniciar constantemente a conexão Wi-Fi do módulo, o que pode ser observado na Figura 35.

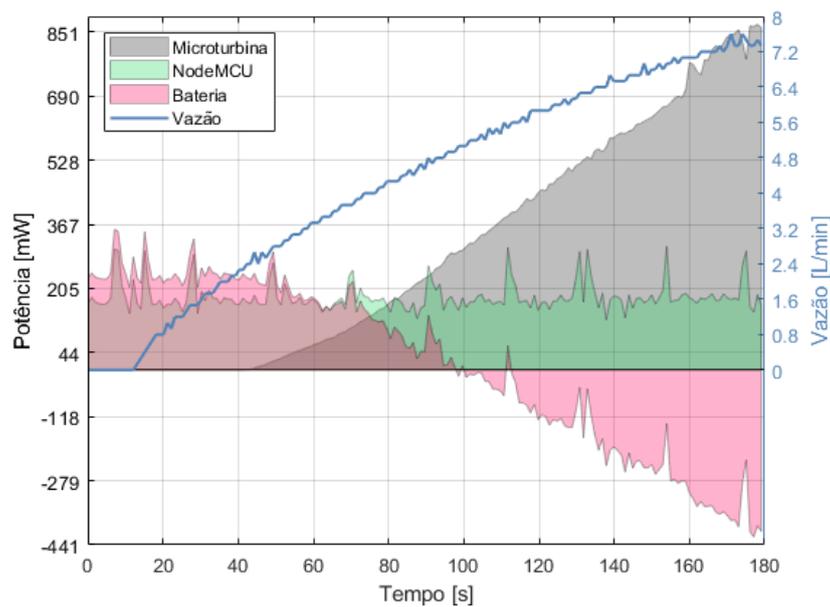
Figura 35 : Corrente solicitada pelo módulo NodeMCU, conexão Wi-Fi e baixo consumo *Modem Sleep*.



Fonte: Autoria Própria

Assim, para que a utilização do *Modem Sleep* seja efetiva é necessário aumentar o tempo entre cada envio de dados ao *ThingSpeak*. Dessa forma, é possível superar a desvantagem dos picos de corrente anteriormente descritas. Para verificar a influência do intervalo de transmissão, realiza-se um primeiro teste com o dispositivo comunicando-se a cada 20 s com a plataforma, mantendo o Wi-Fi conectado de forma constante, como pode ser observado na Figura 36.

Figura 36 : Potência gerada, consumida pelo NodeMCU, com dados a cada 20 s e conexão à rede Wi-Fi.

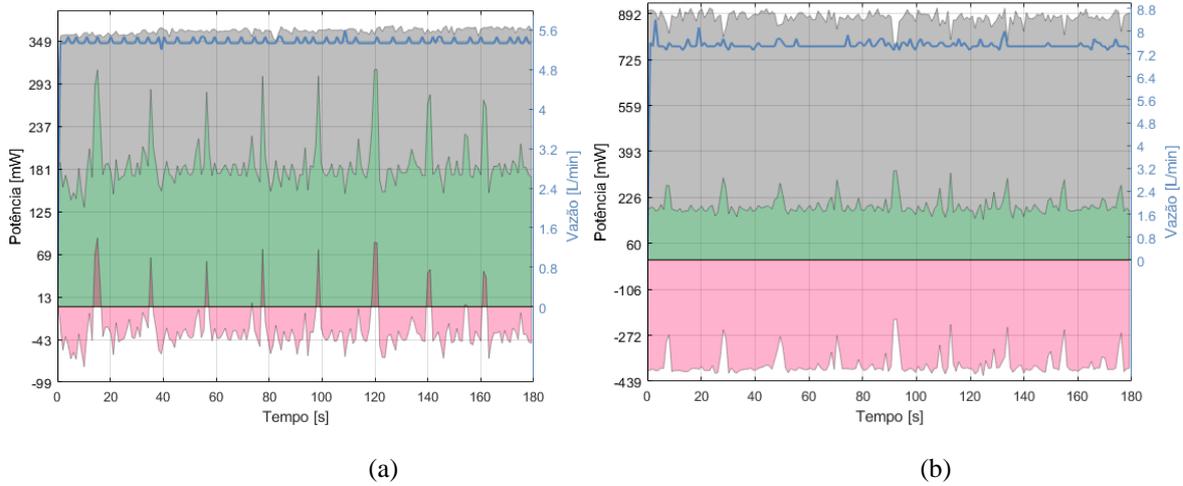


Fonte: Autoria Própria

Para a configuração em questão, obteve-se um consumo médio de potência pelo NodeMCU de 188,96 mW. A microturbina passa a gerar energia aos 43,31 s, correspondente à vazão de 2,40 L/min. Como esperado, a solicitação de potência pelo módulo é inferior à obtida para a configuração de transmissão a cada 15 s. Novamente, tem-se na Figura 37 a resposta do sistema para dois valores de vazão constantes.

Na Figura 37(a) é possível verificar com clareza a solicitação de potência da bateria nos momentos em que ocorrem os picos referentes à transmissão de dados e fornecimento de energia da microturbina para a bateria nos demais momentos. A potência média resultante é fornecida para a bateria e tem valor aproximado de 30,27 mW. Em vazão máxima, configuração ilustrada na Figura 37(b), a potência média gerada pela microturbina é de 877,98 mW e a bateria é carregada com uma potência média de 378,83 mW.

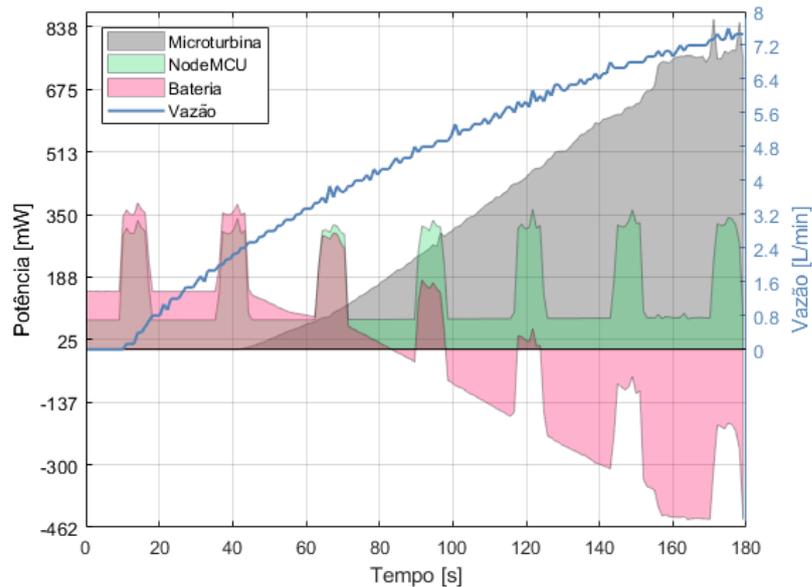
Figura 37: Potência gerada, consumida pelo NodeMCU para vazão fixa em (a) 5,33 L/min e (b) 7,47 L/min.



Fonte: Autoria Própria

A Figura 38 mostra o comportamento do sistema para a transmissão a cada 20 s e utilização de *Modem Sleep* nos intervalos de transmissão. A potência média consumida pelo módulo ESP8266 NodeMCU para a configuração em questão é de 148,29 mW, o que representa uma diminuição de 21,52% no consumo em relação ao caso anteriormente tratado, isto é, sem a utilização do *Modem Sleep*. O comportamento de geração da microturbina continua similar aos obtidos, com a geração iniciando em 43,33 s, na vazão de 2,40 L/min.

Figura 38: Potência gerada, consumida no sistema para transmissão a cada 20 s e Modem Sleep.



Fonte: Autoria Própria

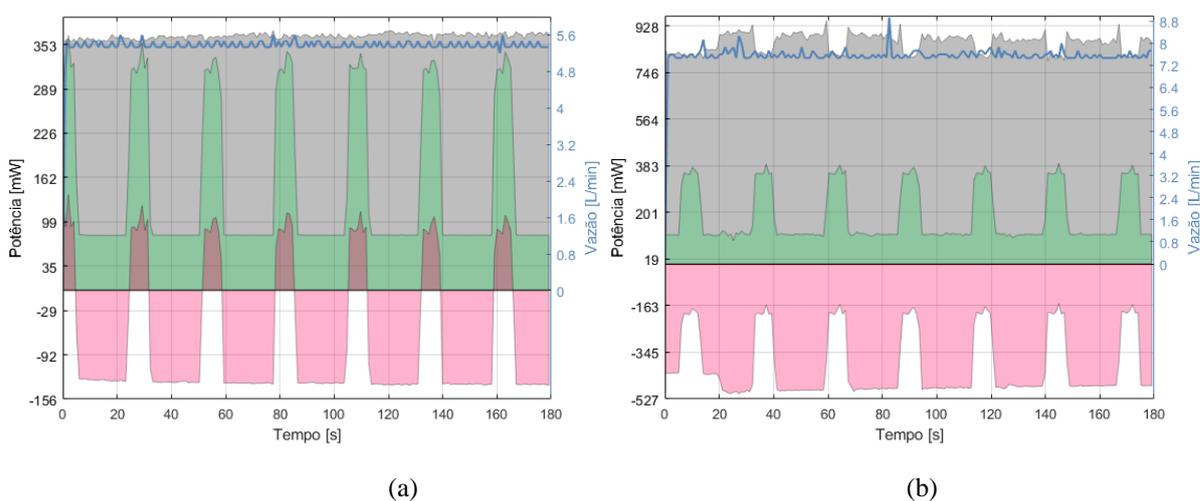
A

Figura 39 apresenta o comportamento do sistema para vazão constante. Para a vazão fixa representada na

Figura 39(a) tem-se uma geração média pela microturbina de 364,84 mW e a bateria recebe 69,97 mW de potência média para seu carregamento. Para o caso de vazão máxima, ilustrado na

Figura 39(b), tem-se uma geração média de 862,10 mW pela microturbina e a bateria é carregada com uma potência média de 394,30 mW. Novamente, observa-se o fenômeno de variação na potência gerada de acordo com a solicitação da carga.

Figura 39: Potência gerada, consumida pelo NodeMCU com vazão fixa em (a) 5,35 L/min e (b) 7,53 L/min

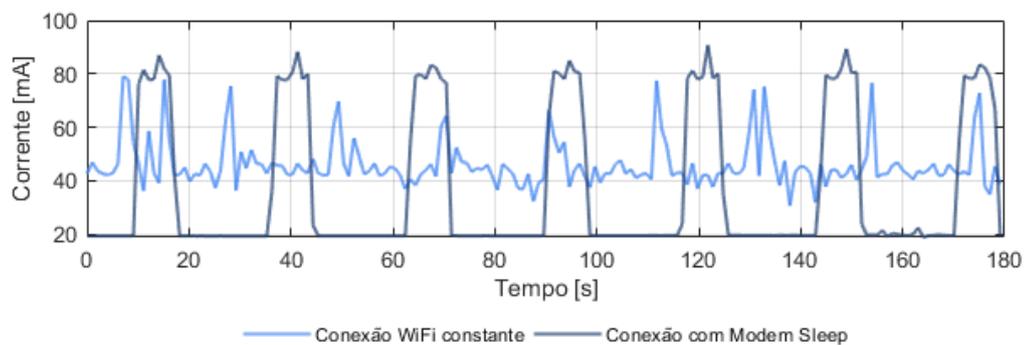


Fonte: Autoria Própria.

Com relação ao sistema comunicando-se com a plataforma *ThingSpeak* a cada 15 s, o aumento de tal intervalo para 20 s e utilização do modo de baixo consumo *Modem Sleep* representou uma economia de aproximadamente 23,66% na potência consumida pelo módulo ESP8266 NodeMCU. Verifica-se que, tal como observado anteriormente, a corrente solicitada durante a transmissão de dados é maior para o caso utilizando o modo de baixo consumo, uma vez que é necessário o *wakeup* e reconexão do Wi-Fi constantemente.

Para o módulo operando sem *Modem Sleep*, o máximo valor de corrente observado durante o ensaio realizado é de 79,02 mA. Para o caso de baixo consumo tem-se um valor máximo de 90.76 mA. O comportamento descrito pode ser verificado na

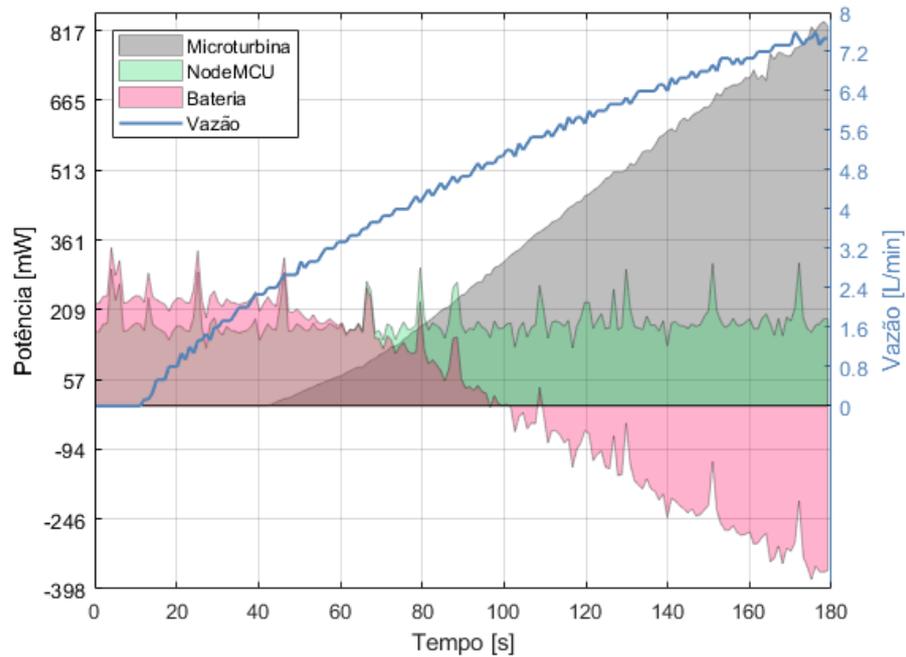
Figura 40.

Figura 40: Corrente solicitada pelo módulo NodeMCU, conexão Wi-Fi ininterrupta e *Modem Sleep*.

Fonte: Autoria Própria.

Realiza-se um aumento no intervalo de comunicação para 60 s e sem o emprego do modo de baixo consumo, obtendo-se o resultado da Figura 41. Para o intervalo selecionado, espera-se uma redução considerável no consumo energético, mantendo uma taxa aceitável de comunicação com a plataforma de IoT. Observa-se que, apesar do aumento na taxa de transmissão, a conexão Wi-Fi constante continua solicitando diversos picos de consumo.

Figura 41: Potência gerada, consumida e vazão de fluido, transmissão a cada 60 s e conexão Wi-Fi.

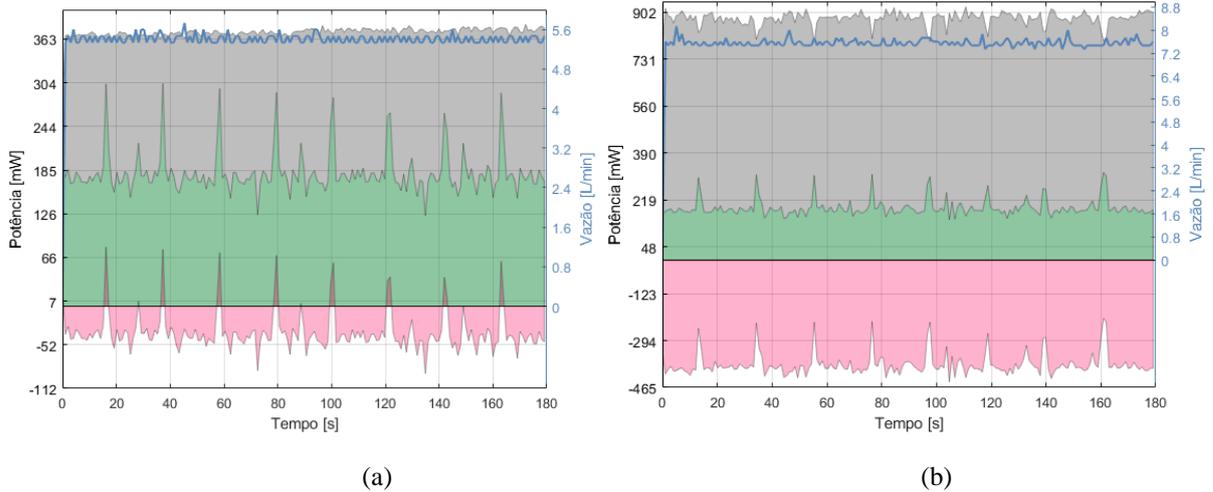


Fonte: Autoria Própria.

Para a configuração descrita, tem-se um consumo médio pelo módulo ESP8266 NodeMCU de aproximadamente 182,52 mW e a microturbina passa a gerar energia em 42,32 s, correspondente à vazão de 2,40 L/min. Para o Wi-Fi constantemente conectado, verifica-se que o consumo de energia pelo módulo diminuiu pouco com relação ao aumento do intervalo de transmissão comparando os casos supracitados. Além disso, o comportamento obtido para a geração pela microturbina foi bastante comum entre todos os ensaios efetuados até o momento.

Para verificação dos parâmetros de comportamento da bateria e geração máxima da microturbina, tem-se os ensaios em vazão constante ilustrados na Figura 42. Para a Figura 42(a), tem-se uma potência média de 359,34 mW no sentido de carregamento da bateria; em vazão máxima, tem-se 879,61 mW médios gerados pela microturbina. Ainda que menos intenso, o comportamento de variação da geração pode ser notado também na Figura 42(b).

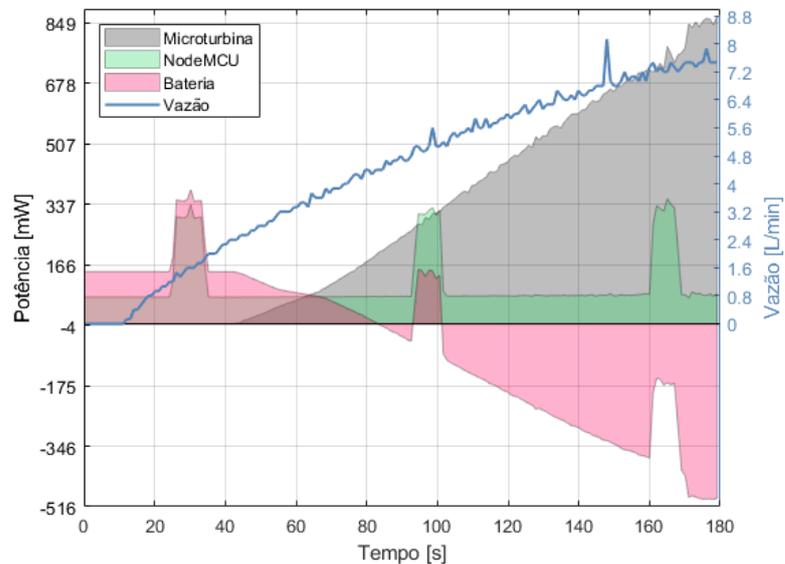
Figura 42: Potência gerada, consumida para vazão fixa em (a) 5.38 L/min e (b) 7.51 L/min.



Fonte: Autoria Própria.

A Figura 43 apresenta o comportamento do sistema com o módulo NodeMCU utilizando o modo de baixo consumo *Modem Sleep*. Tem-se, agora, um consumo médio para o módulo de 108.90 mW, cerca de 40,33% menor do que o caso anteriormente citado com a mesma taxa de transmissão e sem utilização do modo de baixo consumo. O início da geração se dá em 43,30 s na vazão de 2,40 L/min, consolidando esta como a vazão mínima necessária para a geração de energia pela microturbina.

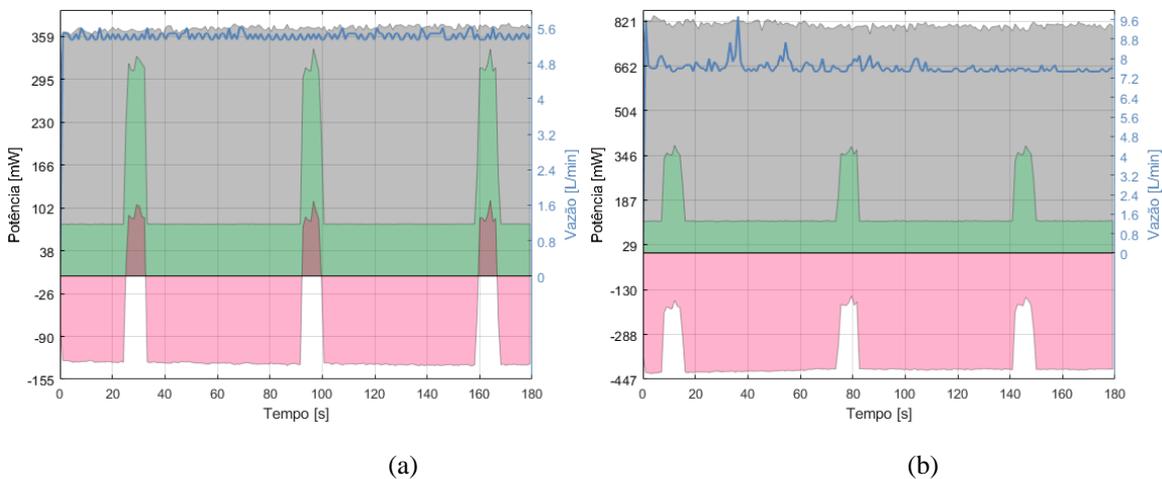
Figura 43: Potência gerada, consumida no sistema, transmissão a cada 60 s e *Modem Sleep*.



Fonte: Autoria Própria.

Os ensaios em vazão constante, ilustrados na Figura 44, apresentam um comportamento bastante semelhante ao obtido para as configurações anteriores. Na Figura 44(a), pode-se verificar a solicitação de potência da bateria pelo módulo ESP8266 NodeMCU nos momentos de transmissão. A taxa de comunicação com intervalos maiores permite, para este valor de vazão, um maior fornecimento de potência da microturbina para a bateria. A potência média resultante é fornecida a bateria e tem valor aproximado de 102,37 mW, o qual é 189,67% maior do que o obtido para a mesma taxa de transmissão e sem utilização de Modem Sleep. Na vazão máxima, ilustrada na Figura 44(b), tem-se 807,25 mW de potência média gerada.

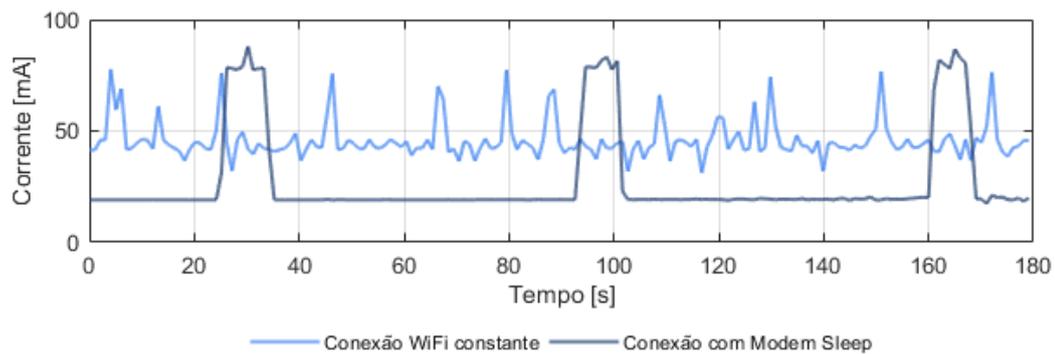
Figura 44: Potência gerada, consumida para vazão fixa em (a) 5,38 L/min e (b) 7,60 L/min.



Fonte: Autoria Própria.

Tem-se, na Figura 45, o comportamento da corrente solicitada pelo módulo ESP8266 NodeMCU para as duas configurações anteriormente tratadas. Para o caso de conexão Wi-Fi constante, tem-se um consumo médio de corrente de 45,89 mA. Por outro lado, a corrente média para a configuração com *Modem Sleep* é de 27,25 mA, o que representa uma diminuição de 40,62% em relação à operação com Wi-Fi conectado constantemente. Apesar dos picos com valores mais expressivos durante o *wakeup* do módulo (cujo valor máximo mensurado é de 87,89 mA) com relação ao comportamento com Wi-Fi constante (máximo de 77,74 mA), o intervalo de tempo em *sleep* garante um consumo bem menor na maior parte do tempo.

Figura 45: Corrente solicitada pelo módulo NodeMCU, conexão Wi-Fi e configuração *Modem Sleep*



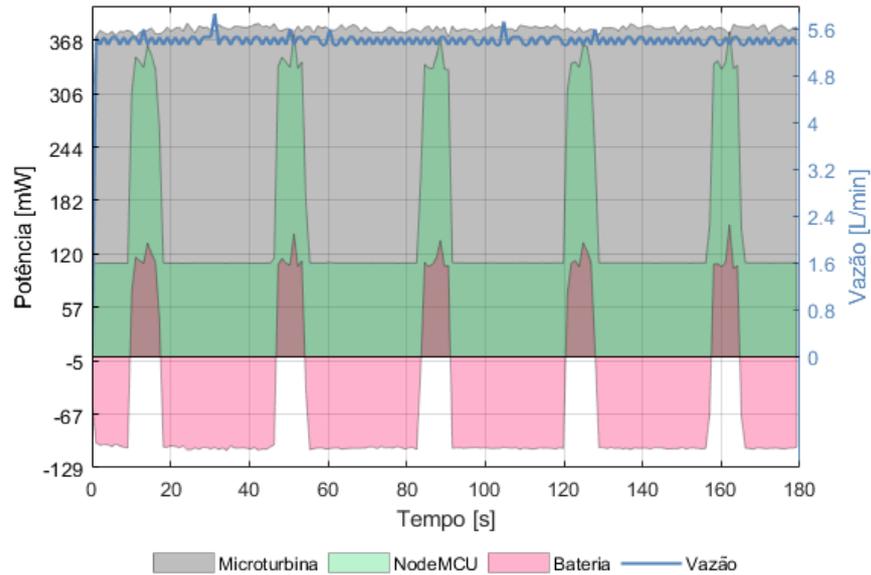
Fonte: Autoria Própria.

5.4 ANÁLISE DA GERAÇÃO DE ENERGIA PARA A CARGA DA BATERIA

Com relação à questão mencionada anteriormente sobre a variação da geração de potência pela microturbina em vazão fixa, levanta-se a hipótese de que a bateria esteja em torno do ponto máximo de carregamento e, dessa forma, a microturbina gera menos energia, uma vez que a carga exige menor potência. Para verificação dessa hipótese foram realizados dois ensaios.

No primeiro, realizou-se a descarga da bateria para valores em torno de 0,6V abaixo da tensão máxima que ela pode fornecer (valor nominal de 4,20V) e então aplicou-se uma tensão fixa de 3,0 V no controle de vazão da bomba. No segundo teste, realizou-se anteriormente a carga completa da bateria e então aplicou-se a mesma tensão no controle da bomba. A Figura 46 apresenta o comportamento descrito pelo primeiro teste.

Figura 46: Potência gerada, consumida e vazão de fluido para transmissão a cada 30 s, e Modem Sleep.

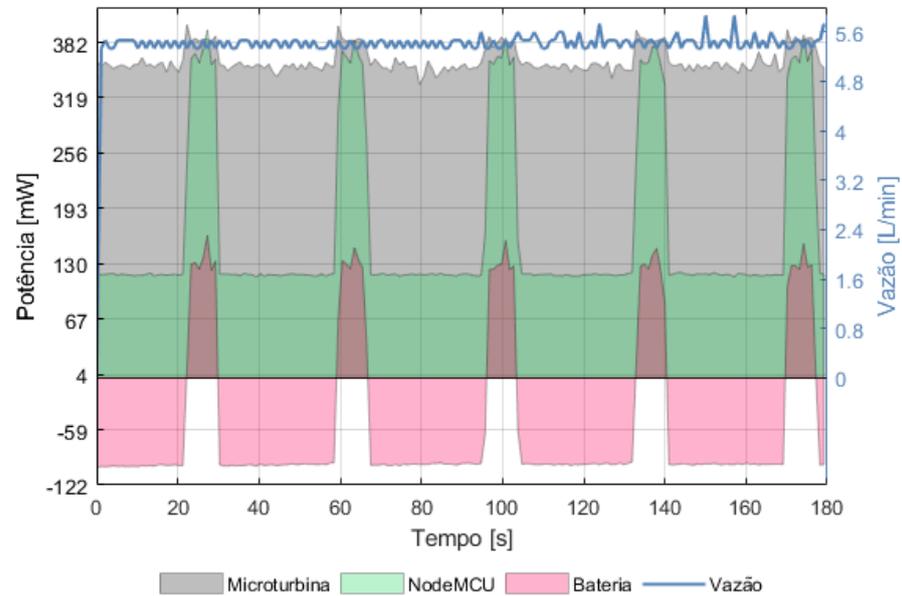


Fonte: Autoria Própria.

Como pode ser observado na Figura 46, o algoritmo utilizado realiza a comunicação com a plataforma *ThingSpeak* a cada 30 s. Esta taxa de transmissão foi escolhida por permitir a visualização de uma quantidade adequada de pulsos de comunicação. Tem-se que a potência média resultante para a bateria é negativa (indicando que a bateria recebeu em maior quantidade energia da microturbina do que forneceu energia de sua carga ao NodeMCU) e tem valor aproximado de 60,39 mW. A vazão média do ensaio é de aproximadamente 5,38 L/min e, nesse valor, a microturbina gera uma potência média de 380,97 mW. Observa-se que não houve grandes variações na potência gerada – com relação ao valor médio calculado, o máximo valor de potência medido foi apenas 7,09 mW maior.

A Figura 47 apresenta o comportamento do sistema para o segundo ensaio realizado, agora com a bateria completamente carregada. No ensaio descrito na Figura 47, cuja transmissão ocorre também a cada 30 s e cuja vazão média é de 5,39 L/min, tem-se também que a potência média é fornecida à bateria e possui valor aproximado de 51,38 mW, cerca de 14,92% menor do que o valor obtido para o ensaio realizado com a bateria com carga incompleta. Além disso, verifica-se a variação da potência gerada pela microturbina, agora bem mais acentuada se comparada ao comportamento da Figura 46.

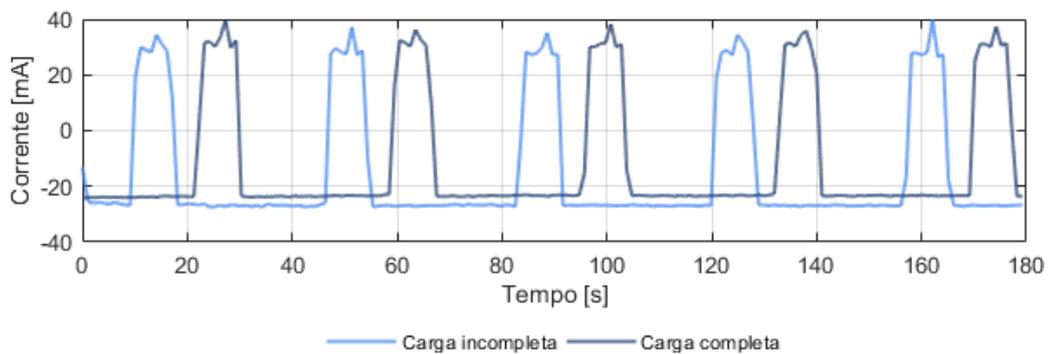
Figura 47: Potência gerada, consumida no sistema, transmissão a cada 30 s, *Modem Sleep* e bateria com carga completa.



Fonte: Autoria Própria.

Na Figura 47, a potência média gerada pela microturbina é 361,35 mW, cerca de 5,15% menor do que a obtida anteriormente. O valor máximo encontrado para potência gerada é de 401,79 mW, o que representa uma variação com relação ao valor médio de 40,44 mW, cerca de 5,70 vezes maior do que a variação encontrada para o caso em que a bateria está com carga incompleta. Pode-se, então, verificar o comportamento da corrente drenada/fornecida pela bateria para os ensaios anteriormente descritos, ilustrado na Figura 48.

Figura 48: Corrente fornecida/solicitada pela bateria para carga incompleta da bateria.



Fonte: Autoria Própria.

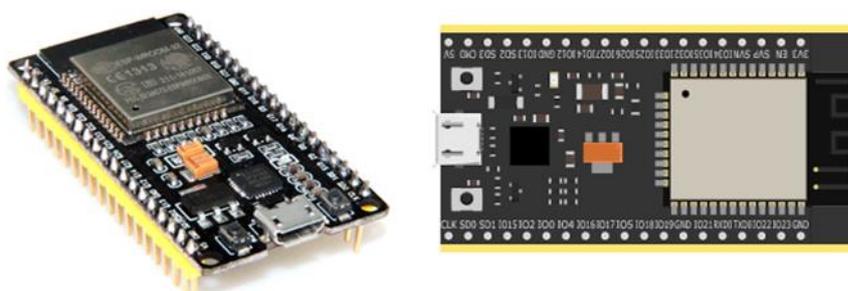
Verifica-se que os picos de corrente relativos à transmissão de dados possuem amplitudes bastante próximas para ambos os casos. Quando em *Modem Sleep*, porém, verifica-se que a corrente no caso de carga completa mantém-se em um patamar mais elevado com relação ao caso de carga incompleta, indicando o menor consumo de corrente, o que é esperado de acordo com as análises realizadas anteriormente.

Para o caso de carga incompleta, tem-se uma corrente média de 15,14 mA (no sentido negativo – isto é, a bateria drena corrente fornecida pela microturbina) e um valor mínimo (também no sentido de carregamento da bateria) de 27,57 mA; para o caso de carga completa, tem-se uma média de 12,06 mA e um valor mínimo de 24,23 mA, ambos negativos.

6 ATUALIZAÇÃO DO HARDWARE DO DISPOSITIVO

A fim de aumentar o potencial de aplicação da solução de medição de vazão autossustentável, permitindo que a mesma possa ser utilizada em aplicações com vazões menores e com maior variabilidade no tempo, é necessária uma maior redução do gasto energético da solução desenvolvida. Para obtenção do objetivo de reduzir ainda mais o consumo energético da solução, principalmente durante os intervalos de transmissão de dados, propõe-se a substituição do módulo ESP8266 pelo módulo ESP32. O ESP8266 e o ESP32 são ambos fabricados pela *Espressif Systems*, multinacional responsável pelo desenvolvimento de soluções em comunicação Wi-Fi, Bluetooth e em Internet das Coisas, com foco em baixo consumo de potência. O ESP32, ilustrado na Figura 49, apresenta as mesmas funcionalidades do ESP8266 e diversas outras ferramentas adicionais.

Figura 49: Imagem do chip Devkit contendo o módulo ESP32



Fonte: <https://www.filipeflop.com/blog/esp32-um-grande-aliado-para-o-maker-iot/>

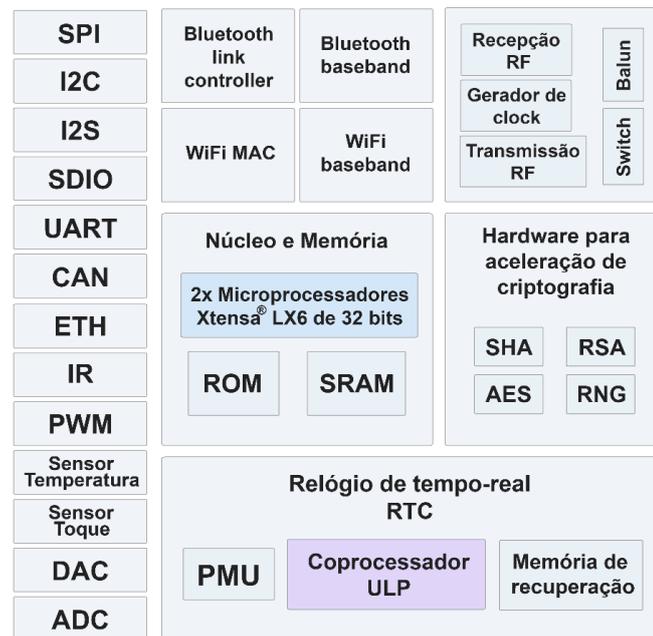
O ESP32 é um microcontrolador de 32 bits que apresenta dois núcleos (diferentemente do ESP8266), contando com 500 kBytes de SRAM, superior aos 160 kBytes do ESP8266. O módulo conta com Bluetooth de baixo consumo energético integrado, um sensor de temperatura capaz de monitorar o aquecimento do microcontrolador, um sensor de toque e um conversor digital-analógico. O módulo apresenta suporte para comunicação SPI, UART, I2C, Infravermelho, SDIO, CAN, Ethernet.

6.1 ESTUDO DO ULP PARA REDUÇÃO DO CONSUMO ENERGÉTICO

O recurso presente no ESP32 de maior interesse para a solução de medição de vazão autossustentável desenvolvida é a presença do coprocessador ULP, sigla que remete a Ultra Low Power. O ULP é um processador simples de baixíssimo consumo de energia, capaz de

realizar processamentos simples com o módulo operando em modo de baixo consumo de energia. Adicionalmente, o módulo ESP32 apresenta uma memória RTC slow SRAM de 8 KiB, utilizada para acesso do ULP quando em modo de baixo consumo, e memória RTC fast SRAM, utilizada para armazenamento de dados e uso da CPU em boot de RTC em modo de baixo consumo. A arquitetura interna do módulo, contendo os principais componentes e protocolos de comunicação, pode ser observada na **Erro! Fonte de referência não encontrada.**

Figura 50: Arquitetura interna funcional do módulo ESP32.



Fonte: Datasheet do ESP32

O módulo ESP32 apresenta os modos de baixo consumo idênticos aos do módulo ESP8266, tratado no presente trabalho: Modem Sleep, Light Sleep e Deep Sleep. Porém, diferentemente do ESP8266, quando acionado o modo mais eficiente de baixo consumo – Deep Sleep – o microcontrolador não fica sem capacidade de processamento. Neste momento pode ser habilitada a utilização do coprocessador ULP. Este coprocessador deve ser programado, porém, em linguagem de baixo nível (Assembly) e utiliza instruções para acessar endereçamento de memória e manipular seus quatro registradores de propósito geral (de 16 bits) (ESPRESSIF, 2019).

Dessa forma, o projeto tem continuidade com a construção de um algoritmo para o ESP32 capaz de colocar o módulo em Deep Sleep durante os intervalos entre os momentos de

comunicação Wi-Fi e manter o coprocessador ULP coletando dados do sensor YF-S201 de forma contínua e armazená-los na memória RTC slow SRAM.

Essa memória RTC é compartilhada entre os três processadores, isto é, os dois principais (dois núcleos) e o ULP, permitindo que os dados armazenados em modo de baixo consumo sejam acessados pelo programa principal. O programa principal, construído em C/C++ com framework Arduino, deve realizar a conversão do número de pulsos em vazão e volume acumulado. Posteriormente, o módulo deve se conectar à rede Wi-Fi e transmitir tais dados. Uma vez realizada a transmissão, o módulo deve entrar novamente em Deep Sleep. É importante ressaltar que, durante todo o processo descrito, a coleta de dados do sensor de fluxo é ininterrupta. No entanto, é necessário avaliar a capacidade de processamento (coleta de dados de pulsos) do coprocessador ULP, pois o mesmo possui capacidade reduzida em vista da necessidade de garantir um baixo consumo de energia.

A disponibilidade de uso do módulo ESP32 com múltiplos processadores e capacidade de armazenamento de dados em memória compartilhada permite a utilização de outras estratégias para tornar solução de medição de vazão autossustentável mais inteligente, ou seja, permite que sejam adotadas estratégias mais complexas para aumentar a eficiência energética do sistema. Pode-se, por exemplo: trabalhar com taxas de transmissão (comunicação sem fio da vazão) variáveis de acordo com a aplicação desejada; o módulo pode identificar longos períodos de vazão nula e assim aumentar o intervalo de tempo entre as transmissões, garantindo menor consumo energético da bateria. Ressalta-se que tais estratégias podem ser executadas de forma dinâmica, uma vez que o coprocessador ULP possui a funcionalidade de realizar o wakeup do(s) processador(es) principais do módulo caso seja verificado algum evento específico.

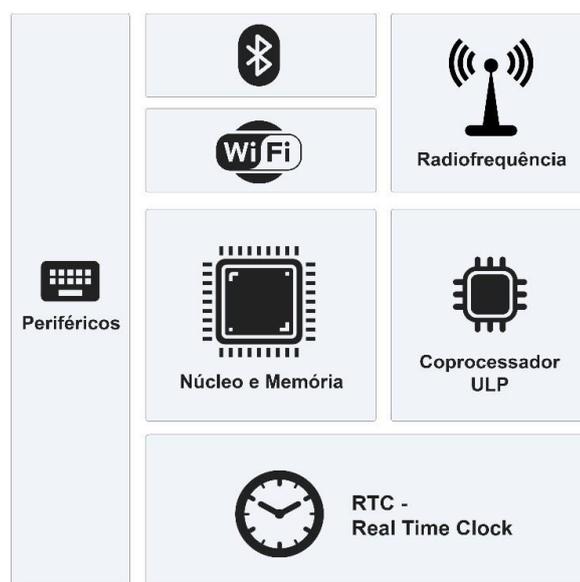
Por se tratar de um microcontrolador com mais recursos (ex: dois processadores), o ESP32 tende a consumir maior energia. Nesse sentido, o estudo detalhado do módulo e aprendizado do uso de recursos (e desativação dos não usados) é essencial de modo que a solução possa garantir um comportamento energético ainda mais eficiente do que o obtido com o ESP8266.

6.2 PERFIS DE CONSUMO DE ENERGIA DO ESP32

Dada a gama de novas possibilidades de implementação utilizando o ESP32, é extremamente importante estudar e documentar detalhes de sua operação considerando os

diversos perfis de consumo e trabalho suportados pelo módulo. Para tal, pode-se resumir a arquitetura interna do ESP32 através do diagrama esquemático da Figura 51.

Figura 51: Diagrama esquemático simplificado do módulo ESP32



Fonte: Datasheet do ESP32

Através do diagrama, pode-se indicar os componentes do módulo que estão ativos ou desativados em cada um dos perfis de consumo disponíveis. O gerenciamento de energia do módulo permite a utilização de cinco perfis de consumo: Modo Ativo, *Modem Sleep*, *Light Sleep*, *Deep Sleep* e Modo de Hibernação; cada perfil possui características próprias e, principalmente, aplicações a que se destinam – essa flexibilidade de gerenciamento energético é um recurso extremamente importante para aplicações de Internet das Coisas, uma vez que as dinâmicas de comunicação entre dispositivos são variadas e podem, assim, ser otimizadas para garantir maior durabilidade de baterias.

✓ *Modo Ativo* – é o perfil padrão do ESP32, onde todos os recursos estão ativos. É natural que, nesse perfil, o consumo assuma valores bastante elevados; uma vez que o Wi-Fi e o Bluetooth estão ligados o tempo todo, a corrente de operação é de cerca de 160~240 mA e picos de 800 mA podem ser atingidos durante instantes de comunicação. A Figura 52 ilustra os componentes ativos nesse modo.

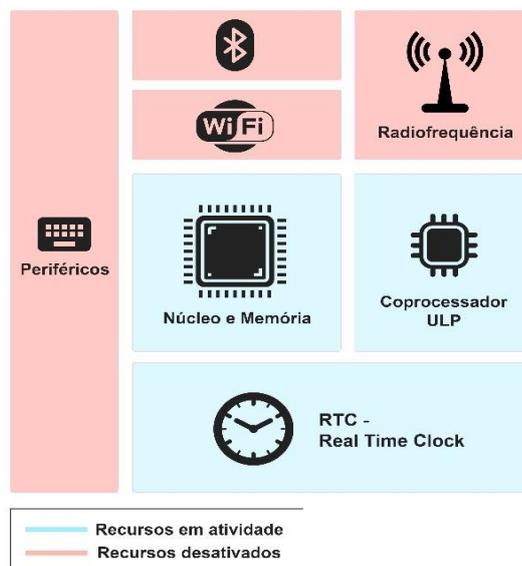
Figura 52: Diagrama esquemático simplificado do módulo ESP32 – modo ativo.



Fonte: Datasheet do ESP32

✓ *Modem Sleep* – nesse perfil, o Wi-Fi, Bluetooth, rádio e periféricos ficam inativos; os dois processadores principais, o coprocessador ULP e o RTC ficam ativos, ocasionando em um consumo médio que varia entre 3 mA e 20 mA (para operação em frequência elevada). Tem-se, na Figura 53, o diagrama ilustrativo do estado dos recursos do módulo nesse modo de operação onde é possível criar padrões de conexão automáticos, onde o módulo desperta em intervalos de tempo determinados para realizar comunicação.

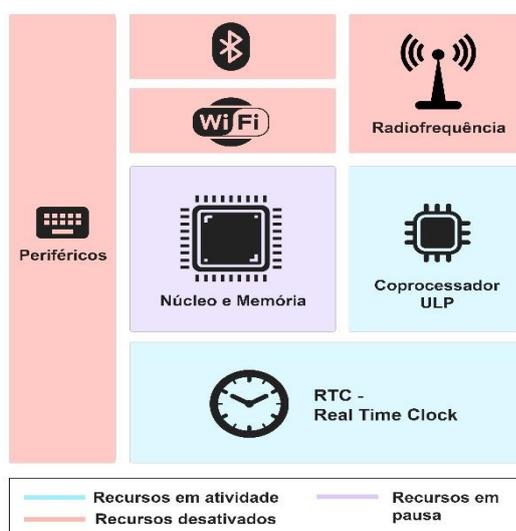
Figura 53 : Diagrama esquemático simplificado do módulo ESP32 – modo modem sleep



Fonte: Datasheet do ESP32

✓ *Light Sleep* – esse perfil de consumo é semelhante ao *Modem Sleep*; a CPU, porém, é pausada através do chaveamento de seus pulsos de clock. Desligando os pulsos de clock, não é necessário que o circuito lógico altere seu estado e, dessa forma, economiza-se energia e, ao mesmo tempo, é mantido o estado do dispositivo armazenado em memória e as informações não são perdidas durante o *sleep*. Nesse perfil, o módulo apresenta um consumo médio de 0,8 mA e o estado dos componentes podem ser observados na Figura 54.

Figura 54 : Diagrama esquemático simplificado do módulo ESP32 – modo light sleep



Fonte: Datasheet do ESP32

✓ *Deep Sleep* – nesse perfil, conforme Figura 55, os periféricos, recursos de comunicação e até mesmo a CPU são desligados. Em atividade, tem-se apenas os recursos relacionados ao RTC – *Real Time Clock*, como o controlador RTC, as memórias *RTC slow SRAM* e *RTC fast SRAM*, e o coprocessador ULP. Durante o *sleep*, juntamente com a CPU, a memória principal do módulo é desativada, seus dados são apagados e ela não pode ser acessada enquanto não for realizado o *wakeup*; porém, como anteriormente citado, a memória RTC permanece em operação e, dessa forma, pode-se armazenar dados que não devem ser voláteis durante a execução do algoritmo – a obtenção do volume, por exemplo, é realizada indiretamente através de medições sucessivas e cumulativas de vazão, o que implica na necessidade de dados históricos para obtenção da medição atual. O *wakeup* do módulo pode ser realizado através de temporização, interrupções externas e pelo próprio coprocessador ULP, o qual compartilha, também, dos dados armazenados na memória RTC. Nesse perfil, o consumo de corrente do módulo pode chegar a valores inferiores a 0,15 mA.

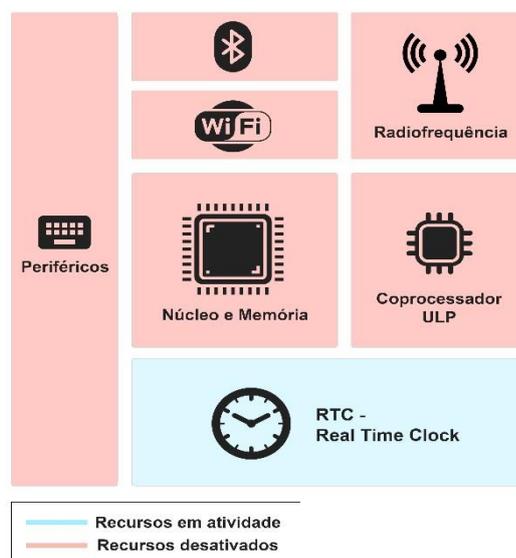
Figura 55 : Diagrama esquemático simplificado do módulo ESP32 – modo deep sleep



Fonte: Datasheet do ESP32

▪ *Modo de Hibernação* – nesse modo, ilustrado na Figura 56, até mesmo o coprocessador ULP é desativado, bem como a memória RTC e o oscilador interno. Dessa forma, não é possível armazenar dados de forma não volátil e todos os dados são perdidos durante o *sleep*. Apenas alguns recursos do RTC continuam operantes com o objetivo de realizar o wakeup do módulo; esse modo apresenta aplicações específicas devido às limitações de processamento, porém o consumo médio de corrente nesse perfil pode chegar a valores em torno de 2,5 μA .

Figura 56 :Diagrama esquemático simplificado do módulo ESP32 – modo hibernação



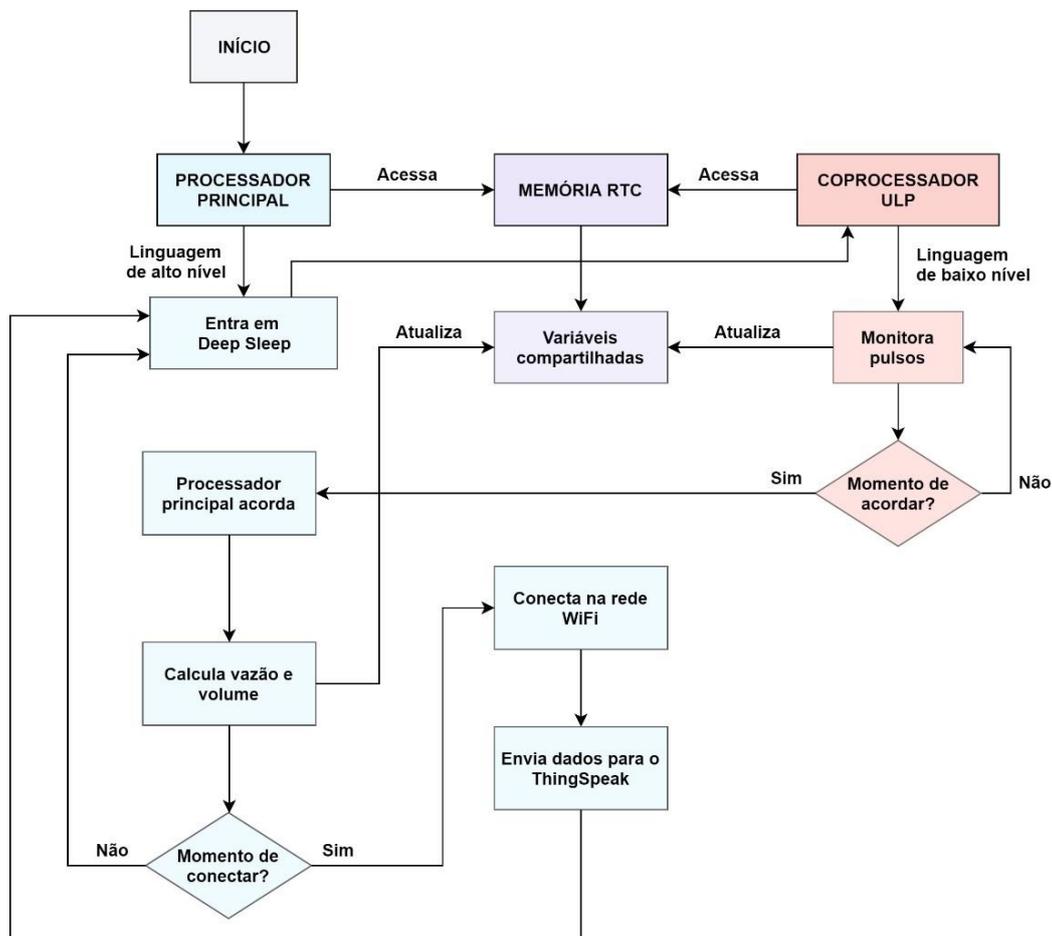
Fonte: Datasheet do ESP32

6.3 ALGORITMO DE MEDIÇÃO PARA OTIMIZAÇÃO DE CONSUMO

Dadas as novas possibilidades de implementação, faz-se necessário construir um algoritmo que seja capaz de realizar a coleta de dados do sensor de vazão YF-S201, calcular de maneira correta os valores de vazão e volume, realizar a conexão na rede Wi-Fi e enviar os dados para a plataforma *ThingSpeak*; e o mais importante: realizar todas as tarefas considerando eficiência energética e otimização do consumo de energia do módulo. É natural que, dadas as possibilidades de utilização de um coprocessador de baixo consumo e memória não-volátil compartilhada entre os processadores, o algoritmo torne-se mais complexo em comparação ao já desenvolvido para o módulo ESP8266 NodeMCU.

Nesse sentido, tem-se, na Figura 57, um fluxograma que apresenta as principais funções do algoritmo construído, bem como as dinâmicas de acesso à memória não-volátil e as tarefas realizadas pelo coprocessador ULP.

Figura 57 :Fluxograma do algoritmo para medição de vazão no modo deep sleep



Fonte: Autoria própria

O início do algoritmo se dá com a inclusão de bibliotecas responsáveis por permitir a conexão Wi-Fi e operação em *Deep Sleep* com utilização do coprocessador ULP; então, são definidos alguns parâmetros constantes para utilização durante a execução e os parâmetros responsáveis pela conexão com a rede Wi-Fi e com o *dashboard* no ThingSpeak. Por fim, são definidas as variáveis armazenadas na memória RTC, as quais podem ser manipuladas e alteradas pelo processador principal e pelo coprocessador ULP.

O algoritmo do processador principal foi construído inteiramente através de funções – isso torna o fluxo de execução mais flexível e simplifica manutenções que eventualmente podem ser realizadas. As principais funções construídas foram: conexão com a rede Wi-Fi, transmissão de dados para a plataforma ThingSpeak, inicialização das variáveis e configurações do coprocessador ULP, execução do coprocessador ULP e cálculo de vazão/volume. Definidas tais rotinas, é habilitado o *wakeup* através de temporização, é inicializado o coprocessador ULP e então inicia-se o *Deep Sleep*.

O coprocessador ULP é responsável apenas pelo monitoramento do GPIO onde o sensor de vazão YF-S201 está conectado. O sensor opera com pulsos proporcionais à vazão de fluido que passa por sua estrutura interna; assim, é necessário realizar a contagem desses pulsos em um intervalo de tempo pré-definido – nesse caso, o tempo de *sleep* do módulo. O coprocessador ULP possui uma lista de instruções bastante reduzido e permite apenas operações simples com o módulo – além disso, cabe ressaltar que sua programação é inteiramente realizada em linguagem de baixo nível, o que agrega certa complexidade na construção das rotinas necessárias para uma correta aquisição de dados. O coprocessador realiza o monitoramento na porta e, realizando um *debouncing* através de um filtro de média, compara o estado atual com o anterior – assim, realiza a identificação de pulsos e incrementa uma variável presente na memória RTC.

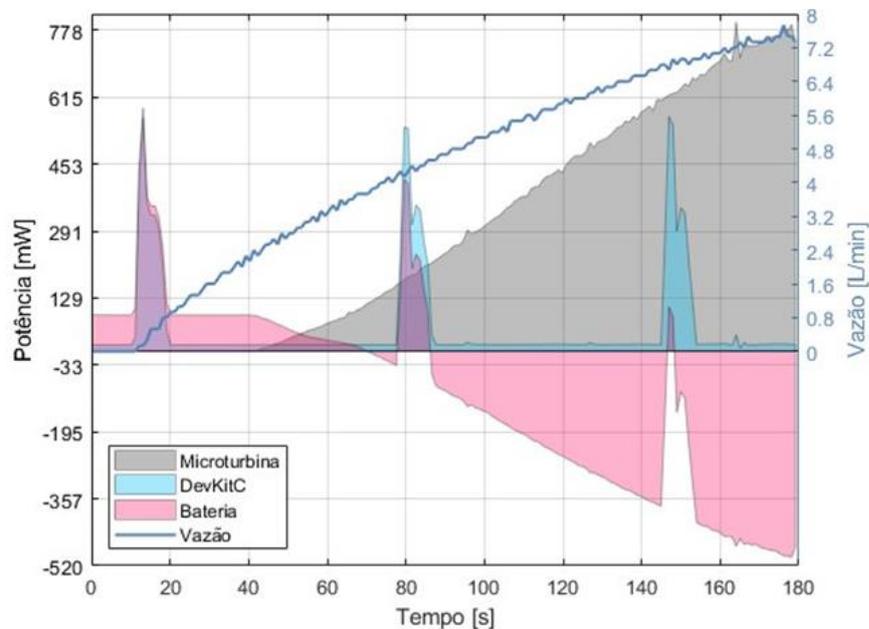
É realizado, então, o *wakeup* do processador principal. Em seguida, ele realiza a conversão da contagem de pulsos e tempo de *sleep* (inicialmente definido como 1 s) em dados de vazão/volume. É verificado, então, se foi atingido o intervalo de tempo pré-configurado para transmissão dos dados para a plataforma ThingSpeak; caso ainda não tenha sido atingido, o processador retorna ao modo de *sleep*, enquanto o ULP continua sua rotina de monitoramento do GPIO. Caso seja momento de realizar a transmissão de dados, o processador principal executa a rotina de conexão à rede Wi-Fi e transmite os dados em formato adequado para a *dashboard* da plataforma em nuvem.

6.4 ENSAIOS PARA VALIDAÇÃO DO POTENCIAL DE REDUÇÃO DE CONSUMO DO ESP32

Para verificar o potencial de melhoria para a solução de medição de vazão autossustentável, inicialmente foram realizados alguns ensaios experimentais para verificação do comportamento do módulo ESP32 na plataforma de testes. Na Figura 58, verifica-se o consumo energético do módulo para um intervalo entre a transmissão de dados de vazão de 60 s. Foi aplicado um sinal em rampa no controle da vazão, variando-a até a máxima vazão fornecida pela bomba.

O algoritmo implementado nesses testes iniciais apenas envia um dado fixo para a plataforma *ThingSpeak* (não realiza coleta de dados do sensor de vazão e nem utiliza o coprocessador ULP). Além disso, nos intervalos entre cada transmissão, o módulo ESP32 entra em modo de baixo consumo *Deep Sleep*, reduzindo significativamente seu consumo de energia.

Figura 58 : Potência gerada e consumida pelo módulo ESP32 Devkit e bateria com transmissão de dados a cada 60 s e utilização de Deep Sleep



Fonte: Autoria própria

Observa-se, na Figura 58, que apesar de haver uma maior solicitação de energia no momento do wakeup do módulo e da transmissão de dados, durante o *Deep Sleep*, o consumo

do ESP32 é significativamente inferior àquele apresentado na solução obtida com o ESP8266 para o mesmo ensaio. O uso do modo *Deep Sleep* do ESP32 Devkit resultou em um consumo médio durante o período de sleep de 16,3 mW, enquanto o Modem Sleep do ESP8266 NodeMCU apresentou um consumo médio de 76, 2 mW. Outro fator importante é corrente consumida durante o sleep que, para o ESP8266 NodeMCU foi de aproximadamente 19.3 mA; para o ESP32 Devkit foi obtido um consumo de 3,9 mA.

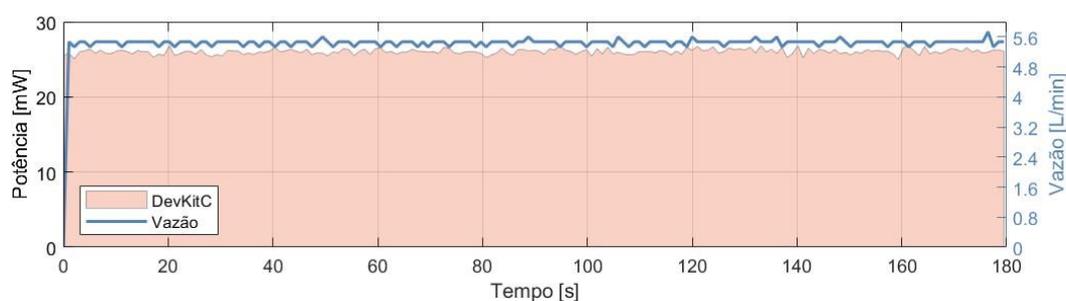
Para o ensaio em questão, tem-se um consumo médio para o ESP32 Devkit de 59,57mW. Enquanto isso, para o mesmo ensaio, o ESP8266 NodeMCU apresentou um consumo médio de 108,90 mW. Assim, verifica-se uma redução de aproximadamente 45,30% no consumo energético ao empregar o módulo ESP32, demonstrando o elevado potencial de melhoria para a solução desenvolvida.

6.5 ENSAIOS DE VALIDAÇÃO DO ALGORITMO PARA MEDIÇÃO DE VAZÃO EM MODO DEEP SLEEP

Os primeiros testes nesta fase e considerando uma maior complexidade da aplicação, alguns testes iniciais de validação com apenas trechos da rotina completa do algoritmo do medidor foram realizados. Dessa forma, é possível identificar com maior facilidade as falhas lógicas na construção do algoritmo proposto inicialmente e ainda pontos de melhoria quanto à precisão da medição e, principalmente, quanto à redução de consumo energético.

Um primeiro ensaio foi realizado considerando apenas a operação do coprocessador ULP, ilustrado na Figura 59. Dessa forma, é possível constatar o valor médio da menor quantidade de energia que será necessária para operação do ESP32 Devkit na solução proposta.

Figura 59 : Potência consumida pelo ESP32 Devkit durante Deep Sleep e com operação apenas do coprocessador ULP realizando aquisição de dados para vazão constante em aproximadamente 5.41L/min.

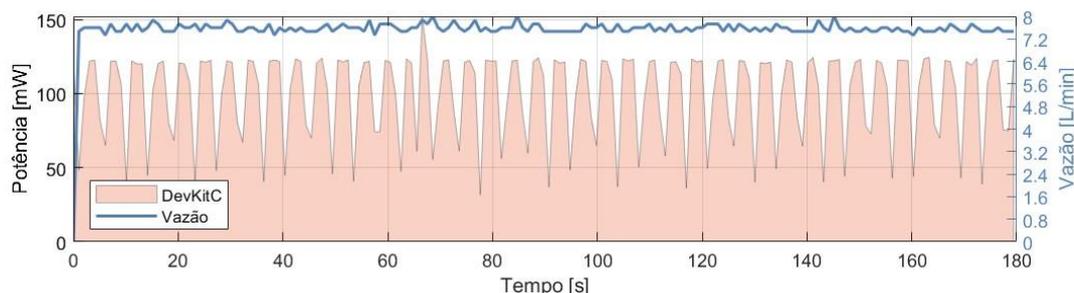


Fonte: Autoria própria

O consumo médio de corrente do Devkit, nessas condições, foi de 5,92 mA. Uma vez que está sendo utilizado um kit de desenvolvimento e não puramente o módulo ESP32, é natural que o consumo seja superior àquele estipulado pela documentação (ESP32, 2016). O consumo médio de potência pelo Devkit foi de 26,03 mW, muito abaixo do obtido para a solução realizada empregando o ESP8266 NodeMCU. Verifica-se, então, que a operação de entrada em modo de baixo consumo e operação do coprocessador ULP está funcionando adequadamente.

Na Figura 60, verificar se o módulo é capaz de despertar do *sleep* em intervalos de tempo pré-determinados, dinâmica que é fundamental para o cálculo de vazão/volume. Ressalta-se que a taxa de despertar do módulo, para manter boa precisão na medição, deve ser reduzida – inicialmente, utiliza-se como tempo de sono o intervalo de 1 s, garantido pelo temporizador que realiza o wakeup do módulo.

Figura 60 : Potência consumida pelo ESP32 Devkit durante Deep Sleep em intervalos de tempo de 1 s e aquisição de dados com ULP para vazão de 7.53 L/min.



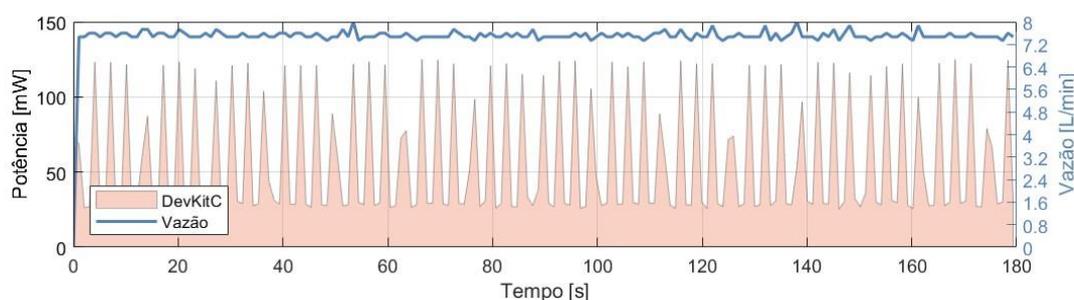
Fonte: Autoria própria

Para o ensaio em questão, obteve-se um consumo médio de corrente de 21,31 mA; o menor valor de potência requerida pelo Devkit foi de 31,57 mW e o valor máximo de potência foi de 152,20 mW, conferindo ao ensaio um valor médio de consumo de cerca de 99,82 mW – o valor obtido é elevado para a solução em objetivo, principalmente ressaltando que ainda não está sendo realizada a conexão Wi-Fi e comunicação. Constata-se que, devido aos recursos extras presentes no módulo, este necessita de maior energia para despertar.

Assim, verifica-se um ponto de atenção para otimização da solução – é necessário ajustar o tempo de sleep de modo a garantir um baixo consumo de potência e, ao mesmo tempo, manter precisão adequada às medições de vazão. Para verificar o potencial de melhoria nesse quesito, ainda não considerando a convergência das duas premissas supracitadas, realizou-se o ensaio da Figura 61, considerando a operação com sleep durante 3 s – essa abordagem

permitiu a obtenção de um consumo médio de 57.06 mW de potência, cerca de 57.16% menor do que o anteriormente obtido.

Figura 61: Potência consumida pelo ESP32 Devkit durante Deep Sleep em intervalos de tempo de 3 s e aquisição de dados com ULP para vazão de 7.48 L/min.



Fonte: Autoria própria

Dessa forma, faz-se ainda necessário verificar a manutenção da precisão do instrumento de medida para a estratégia supracitada e, ainda, validar de maneira mais abrangente o consumo energético para outros intervalos de tempo de aquisição. Realizada tal melhoria, pode-se, então, mensurar os esforços energéticos para que o módulo ESP32 Devkit conecte-se à rede Wi-Fi – tais esforços compreendem o despertar do módulo e a conexão com um ponto de acesso. Por fim, considerando ainda a análise de consumo energético, deve-se realizar a comunicação com a plataforma ThingSpeak e enviar, em formato adequado, os dados de vazão e volume. Tem-se, novamente, uma oportunidade de melhoria de consumo – essa, já utilizada no estudo com o módulo ESP8266, refere-se à variação dos intervalos de conexão e envio de dados.

Conforme já observado com o ESP8266, os momentos de comunicação são aqueles que demandam maior quantidade de corrente e, conseqüentemente, energia para correta operação do módulo; dessa forma, a tratativa desses resultados é bastante crítica no que tange a análise de viabilidade da utilização do ESP32 em detrimento do ESP8266. Uma vez já realizado o estudo aprofundado dos perfis de consumo energético do ESP32, pode-se adotar estratégias de consumo inteligente adaptáveis às aplicações em que a medição de vazão e os dados armazenados serão inseridos, com variações dos tempos de comunicação. Cabe, então, estudar tais possibilidades e verificar se estas são aplicáveis e se a viabilidade conduz a uma otimização significativa na solução proposta.

7 VERIFICAÇÃO DO CONSUMO DE ENERGIA NO MODO DEEP SLEEP

Nesta etapa considera-se a integração entre o algoritmo desenvolvido para a obtenção dos dados de vazão do sensor de fluxo YF-S201 e o algoritmo do ESP 32 – *Deep Sleep* com a comunicação Wi-Fi com a plataforma *ThingSpeak* variando em intervalos de tempo pré-determinados. O algoritmo integrado realiza a coleta de dados de vazão de forma ininterrupta através do ULP que permanece ativo durante todo o tempo no modo Deep Sleep. Após o período de tempo ajustado para envio de dados, faz-se o “wakeup” do ESP 32 para a execução dos cálculos de vazão e comunicação com a plataforma de Internet das Coisas (IoT) ThingSpeak.

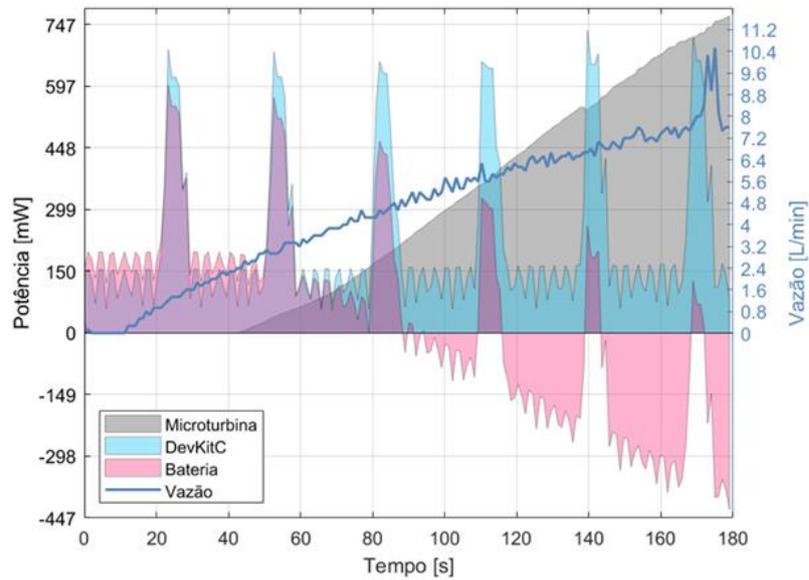
Nos primeiros testes integrados verificou-se a necessidade de um reforço de bateria e utilização de um capacitor de 4400 uF para efetivamente alcançar e manter um nível de tensão suficiente na ocorrência dos picos de consumo nos momentos de efetiva comunicação de dados, nos intervalos de tempo pré-definidos.

7.1 VERIFICAÇÃO DO CONSUMO DE ENERGIA COM TRANSMISSÃO DE DADOS A CADA 15 S

Assim como anteriormente executado com o dispositivo ESP8266, foram executados os testes para verificação da geração e consumo de energia com a transmissão dos dados de vazão a cada 15 s, quando o dispositivo ESP 32 “acorda” e realiza o envio dos dados de vazão coletados. As etapas de testes englobam os ensaios com uma variação da vazão em rampa de zero até a vazão máxima. Em uma segunda etapa adota-se uma vazão constante ajustada para aproximadamente 5,7 L/min. e finalmente com uma vazão constante máxima obtida.

Na Figura 62 pode-se verificar as potências geradas e consumidas com uma variação de vazão desde zero até a vazão máxima, mantendo-se um intervalo de amostra de 180 s. Neste ensaio é possível verificar as mesmas de geração de energia a partir de uma vazão de aproximadamente de 2,53 L/min. O consumo médio calculado do módulo ESP 32 é de 222,38 mW, com uma geração média de 282,37 mW e energia consumida pela bateria de 37,64 mW.

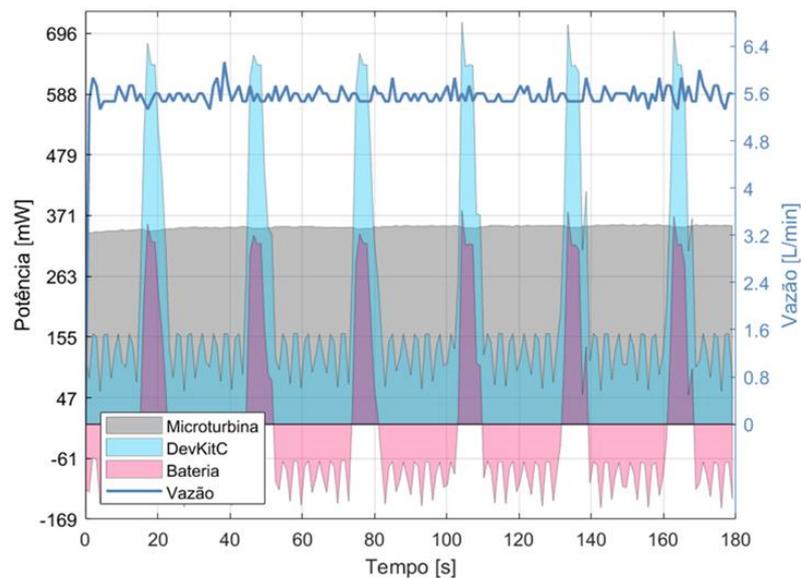
Figura 62: Potência gerada e consumida pelo ESP32 Devkit no modo Deep Sleep, comunicação a cada 15 s e vazão variável.



Fonte: Autoria própria

A Figura 63 apresenta o comportamento do sistema com uma vazão constante de 5,7 L/min, obtendo-se neste ensaio uma geração média de 350,94 mW, consumo do ESP 32 de 223,99 mW e potência da bateria de -14,86 mW. Neste caso o valor negativo de potência significa o efetivo carregamento da bateria

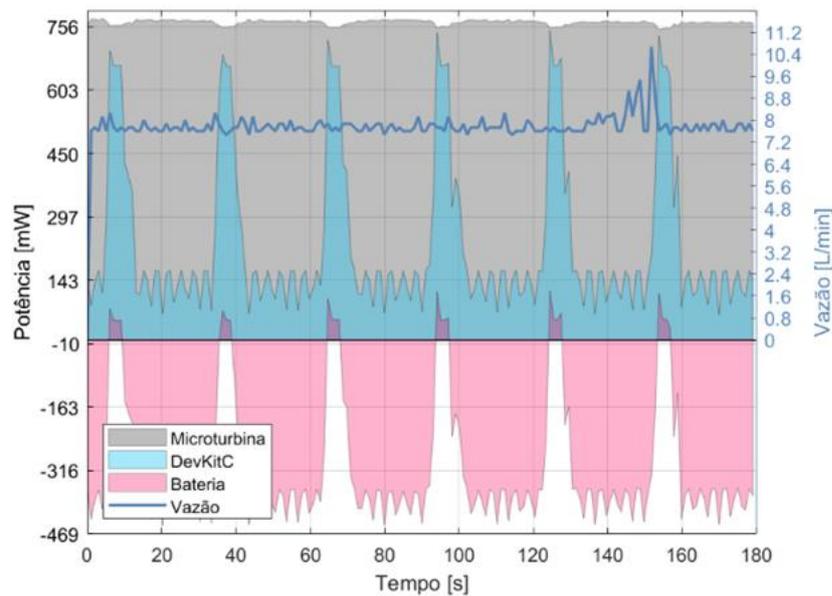
Figura 63: Potência gerada e consumida pelo ESP32 Devkit no modo Deep Sleep, comunicação a cada 15 s e vazão fixa média de 5,7 L/min.



Fonte: Autoria própria

A Figura 64 apresenta o comportamento do sistema com uma vazão máxima em torno de 8,2 L/min, obtendo-se neste ensaio uma geração média de 766,42 mW, consumo do ESP 32 de 234,78 mW e potência da bateria de -305,25 mW. Neste caso o valor negativo de potência da bateria demonstra o carregamento desse componente na maior parte do tempo, com sua utilização como provedora de energia apenas nos picos de transmissão de dados.

Figura 64: Potência gerada e consumida pelo ESP32 Devkit no modo Deep Sleep, comunicação a cada 15 s e vazão fixa máxima de 7,8 L/min.



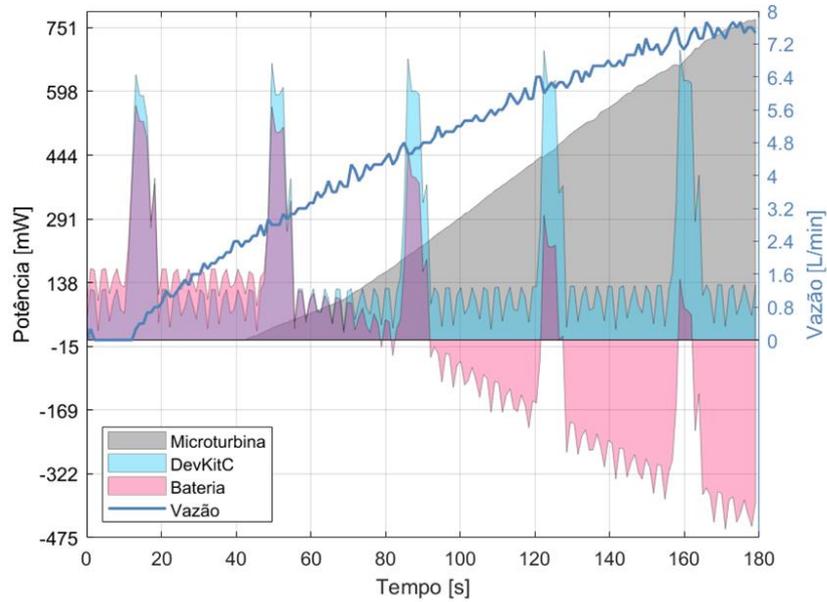
Fonte: Autoria própria

7.2 VERIFICAÇÃO DO CONSUMO DE ENERGIA COM TRANSMISSÃO DE DADOS A CADA 20 S

Nos ensaios a seguir foi alterado o tempo de transmissão de dados para 20 s a fim de verificar o consumo de energia e estabelecer níveis comparativos com os ensaios do mesmo tipo realizados no dispositivo ESP 8266. Novamente foram realizadas três etapas de verificação com vazão variável em rampa, vazão fixa de aproximadamente 5,7 L/min e vazão máxima em torno de 7,8 L/min.

Na Figura 65 são apresentados os resultados com vazão variando em rampa, transmissão de dados a cada 20 s, onde pode-se verificar a geração média da microturbina em 284,70 mW, potência consumida do ESP 32 de 173,87 mW e potência da bateria de -2,99 mW (bateria em carregamento).

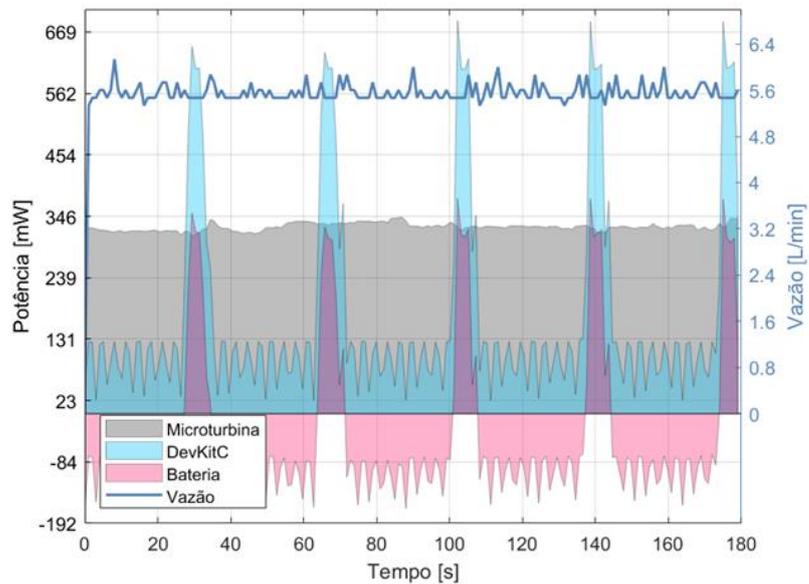
Figura 65: Potência gerada e consumida pelo ESP32 Devkit no modo Deep Sleep, comunicação a cada 20 s e vazão variável



Fonte: Autoria própria

Na Figura 66 são apresentados os resultados com vazão fixa de 5,7 L/min, transmissão de dados a cada 20 s, com geração média em 328,03 mW, potência consumida do ESP 32 de 172,62 mW e potência da bateria de - 38,80 mW (bateria em carregamento).

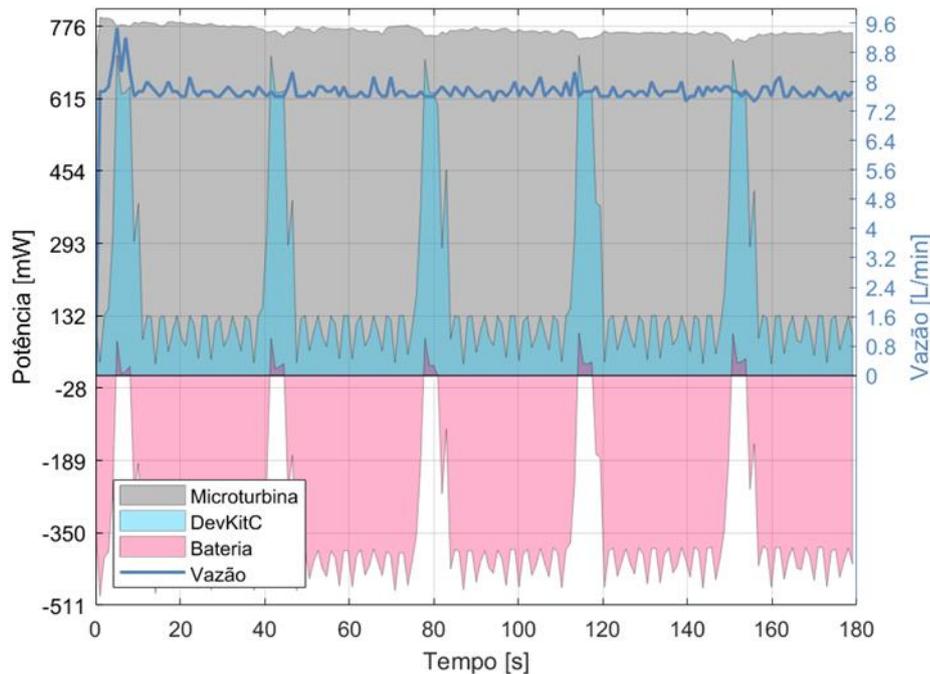
Figura 66: Potência gerada e consumida pelo ESP32 Devkit no modo Deep Sleep, comunicação a cada 20 s e vazão fixa média de 5,7 L/min.



Fonte: Autoria própria

A Figura 67 apresenta o comportamento do sistema com uma vazão máxima em torno de 7,8 L/min, obtendo-se neste ensaio uma geração média de 766,14 mW, consumo do ESP 32 de 182,37 mW e potência da bateria de -347,09 mW (bateria em carregamento).

Figura 67: Potência gerada e consumida pelo ESP32 Devkit no modo Deep Sleep, comunicação a cada 20 s e vazão fixa máxima de 7,8 L/min.

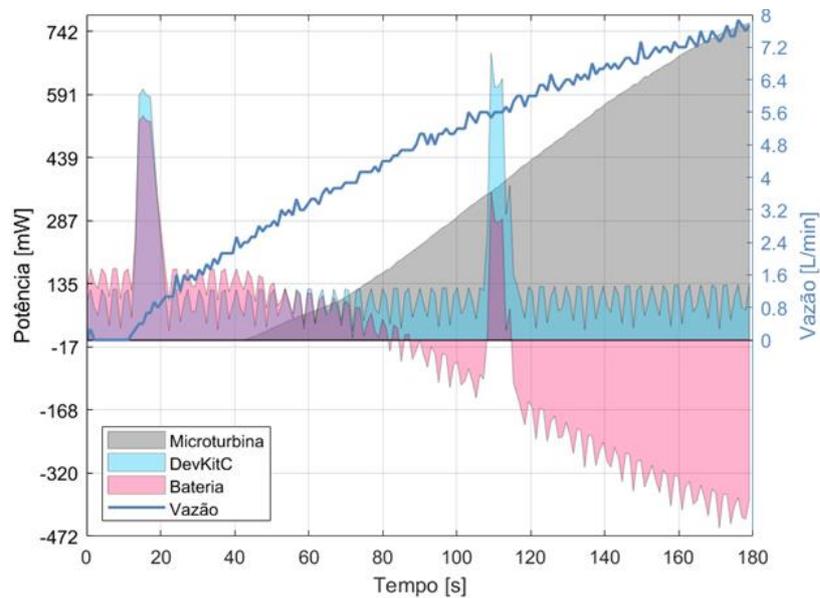


Fonte: Autoria própria

7.3 VERIFICAÇÃO DO CONSUMO DE ENERGIA COM TRANSMISSÃO DE DADOS A CADA 60 S

A Figura 68 apresenta o comportamento do sistema com o módulo ESP 32 utilizando o modo de baixo consumo *Deep Sleep*, vazão variável e transmissão de dados de 60 s. Verifica-se agora um consumo médio para o módulo de 125,22 mW, cerca de 27,98% menor do que os teste anterior e geração de energia de 284,71 mW, com a bateria com potência média de -41,76 mW (em carregamento).

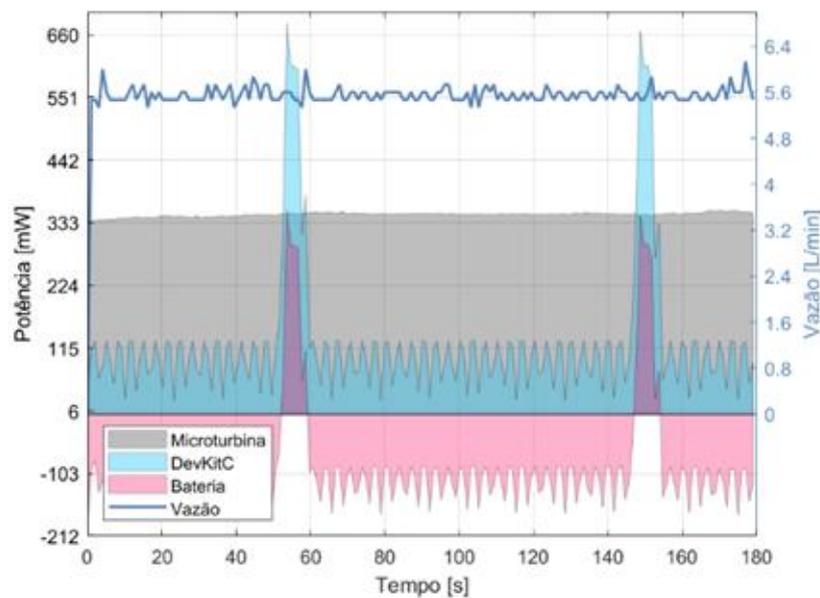
Figura 68: Potência gerada e consumida pelo ESP32 Devkit no modo Deep Sleep, comunicação a cada 60 s e vazão variável



Fonte: Autoria própria

Na Figura 69 são apresentados os resultados com vazão fixa de 5,7 L/min, transmissão de dados a cada 60 s, com geração média em 347,49 mW, potência consumida do ESP 32 de 125,66 mW e potência da bateria de - 91,30 mW (bateria em carregamento).

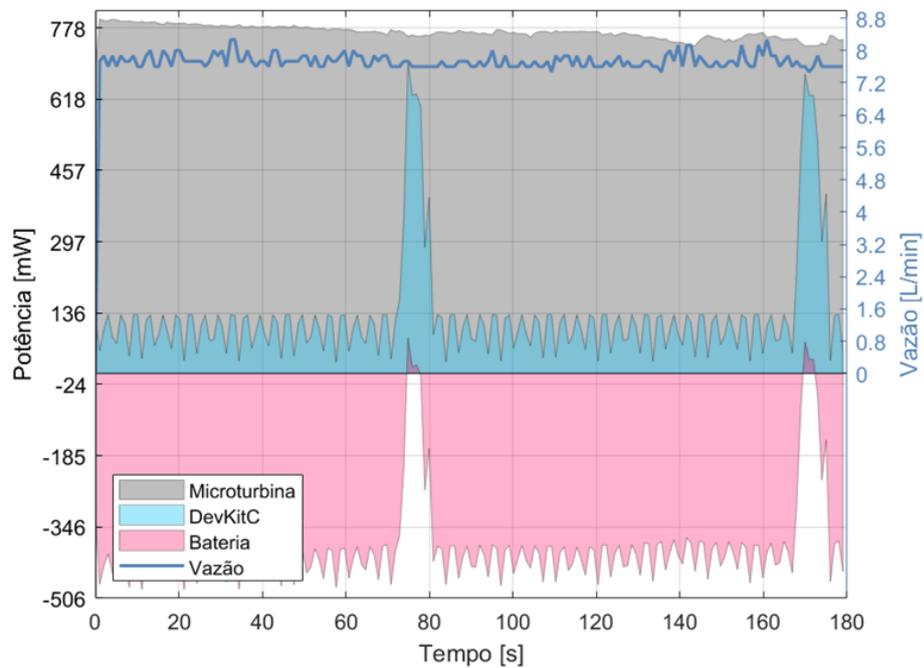
Figura 69: Potência gerada e consumida pelo ESP32 Devkit no modo Deep Sleep, comunicação a cada 60 s e vazão fixa média de 5,7 L/min.



Fonte: Autoria própria

Na Figura 70 são apresentados os resultados com vazão máxima e intervalos de transmissão de dados de 60 s. Neste contexto a geração média é de 768,40 mW, com energia consumida de 131,69 mW no ESP 32 e potência média na bateria de $-389,74$ mW significando o carregamento maximizado do componente durante o período de teste.

Figura 70: Potência gerada e consumida pelo ESP32 Devkit no modo Deep Sleep, comunicação a cada 60 s e vazão fixa máxima de 7,8 L/min



Fonte: Autoria própria

8 RESULTADOS E DISCUSSÕES

A sequência de trabalhos desde a montagem do protótipo, integração do ambiente LabVIEW e escolha inicial dispositivo IoT NodeMCU ESP8266 propiciou a verificação do comportamento do sistema na medição de vazão em várias situações e a validação do modelo proposto para geração de energia em pequenas quantidades a partir da vazão de água como opção para auto sustentabilidade energética de dispositivos de IoT em operação e envio de dados via comunicação wireless para equipamentos remotos de supervisão. No decorrer da evolução dos testes verificou-se a geração de energia suficiente para manter o dispositivo IoT em funcionamento, ao mesmo tempo que, obteve-se energia suficiente para o recarregamento das baterias de suporte, confirmando a viabilidade de implementação em outras escalas.

Na primeira parte dos trabalhos com a utilização do dispositivo NodeMCU ESP8266 obteve-se importantes variações de consumo através do uso do modo de funcionamento Modem Sleep, com variações de até 43% na redução do consumo na operação e envio de dados para a plataforma *ThingSpeak* de aquisição dos dados de vazão do sistema.

O resultado obtido da potência média consumida pelo dispositivo de medição quando utilizado o modo de baixo consumo Modem Sleep em comparação com o modo normal, para a operação em rampa de vazão, foi similar com uma redução de 3% (de 200 mW para 194 mW). Tal fato pode ser explicado pelos picos de maior intensidade na potência consumida pelo dispositivo quando o NodeMCU sai do modo *Modem Sleep*, reativa e reconecta na rede Wi-Fi para transmissão dos dados. Assim, verifica-se que há um esforço maior de potência ao entrar/sair do modo Modem Sleep e reiniciar a conexão Wi-Fi do módulo.

Portanto, para que a utilização do dispositivo de medição no modo *Modem Sleep* seja efetiva é necessário aumentar o tempo entre envio de dados para a nuvem (*ThingSpeak*). Dessa forma, é possível superar a desvantagem dos picos de potência anteriormente descritas.

Na parte final dos testes com a utilização do Devkit ESP32, com maiores possibilidades de programação e envio de dados através do modo de funcionamento *Deep Sleep*, foi possível aprofundar as conclusões sobre o comportamento e evolução dos dispositivos IoT.

Conforme analisado anteriormente, o resultado esperado seria a diminuição da potência média consumida pelo dispositivo de medição quando utilizado o modo de baixo consumo *Deep Sleep* em comparação com o modo *Modem Sleep* e modo normal do NodeMCU ESP 8266. No entanto, a potência média obtida foi maior, apresentando um aumento de 5,6% (de 194 mW para 205 mW). Tal fato pode ser explicado principalmente por dois motivos.

Primeiramente, devido aos grandes picos de intensidade na potência consumida quando o Devkit ESP32 reativa e reconecta na rede Wi-Fi para transmissão dos dados para a nuvem. Assim, verifica-se que o ESP32 demanda maior potência para transmissão de dados que o ESP8266. E também devido à necessidade de o Devkit ESP32 ter de periodicamente chavear a operação entre o modo *Sleep Mode* (com a operação do ULP e contagem dos pulsos) e o modo normal (para cálculo da vazão, escrita na memória e cálculo do volume), como pode ser visto nas Figuras 61 a 69 apresentadas.

Apesar de a potência de consumo do dispositivo de medição com o Devkit em *Deep Sleep* ser de 50 mW, valor bem menor que a potência de consumo de 100 mW obtida com o NodeMCU em *Modem Sleep*, o chaveamento do modo *Deep Sleep* para o modo normal resulta em picos de intensidade na potência de consumo maiores que 200 mW. Como esse chaveamento entre os modos *Deep Sleep* e normal na operação do dispositivo de medição com o Devkit tem que ser periódico e realizado diversas vezes entre intervalos de transmissão dos dados para a nuvem, isso acaba aumentando o valor da potência média consumida pelo dispositivo e resultando no aumento de 5,6% (de 194 mW para 205 mW) informado anteriormente. Em consequência disso, estima-se que o aumento do tempo entre envio de dados para a nuvem (*ThingSpeak*) pelo dispositivo de medição não ofereça vantagem em relação à sua redução do consumo de energia.

Os principais resultados na geração e consumo de energia nos testes com vazões variáveis e a utilização dos diferentes dispositivos de IoT, NodeMCU ESP 8266 e Devkit ESP32 podem ser comparados conforme as Tabelas 2, 3 e 4.

Tabela 2 - Comparativo de geração e consumo de energia dos dispositivos ESP8266 e ESP32 para intervalos de transmissão de 15 s.

Testes Realizados – Intervalo de Comunicação de 15 s	Modem Sleep (ESP 8266)		Deep Sleep (ESP 32)	
	Potência Gerada pela Microturbina (mW)	Potência consumida (mW)	Potência Gerada pela Microturbina (mW)	Potência consumida (mW)
Vazão variável em rampa	299,96	194,25	295,52	205,05
Vazão fixa: 5,7 L/min	351,16	196,25	365,16	203,93
Vazão fixa máxima: 7, 8 L/min-	853,56	201,58	797,32	220,31

Fonte: Autoria própria

Tabela 3 - Comparativo de geração e consumo de energia dos dispositivos ESP8266 e ESP32 para intervalos de transmissão de 20 s.

Testes Realizados – Intervalo de Comunicação de 20 s	Modem Sleep (ESP 8266)		Deep Sleep (ESP 32)	
	Potência Gerada pela Microturbina (mW)	Potência consumida (mW)	Potência Gerada pela Microturbina (mW)	Potência consumida (mW)
Vazão variável em rampa	300,38	148,29	292,82	175,05
Vazão fixa: 5,7 L/min	361,28	186,55	358,51	183,44
Vazão fixa máxima: 7, 8 L/min-	877,98	195,67	771,13	196,28

Fonte: Autoria própria

Tabela 4 - Comparativo de geração e consumo de energia dos dispositivos ESP 8266 e ESP 32 para intervalos de transmissão de 60 s.

Testes Realizados – Intervalo de Comunicação de 60 s	Modem Sleep (ESP 8266)		Deep Sleep (ESP 32)	
	Potência Gerada pela Microturbina (mW)	Potência consumida (mW)	Potência Gerada pela Microturbina (mW)	Potência consumida (mW)
Vazão variável em rampa	306,68	108,90	284,71	155,22
Vazão fixa: 5,7 L/min	372,45	181,23	347,49	125,66
Vazão fixa máxima: 7, 8 L/min-	879,60	189,82	768,40	131,69

Fonte: Autoria própria

Os resultados apresentados nas Tabelas 2, 3 e 4 demonstram que a desvantagem de uso do dispositivo de medição em modo *Deep Sleep*, em comparação ao *Modem Sleep*, aumenta significativamente com o aumento do tempo entre transmissões de dados para a nuvem (de 5,6% para 15 s para 42,2% para 60 s). Quanto maior for o tempo entre transmissões de dados para a nuvem aliado ao fato de o dispositivo de medição necessitar sair periodicamente do modo *Deep Sleep* para o modo normal para cálculo da vazão e volume praticamente inviabiliza seu uso quando o foco é a redução do consumo de energia.

O aumento do tempo de transmissão de 15 até 60 segundos acarreta redução no consumo de energia do dispositivo de medição com o NodeMCU, de 9% no modo de operação normal, até 56% no modo *Modem Sleep*, evidenciando que grande parte do consumo de energia do dispositivo está relacionada à transmissão de dados sem fio. Por fim, na comparação entre a operação do dispositivo de medição no pior caso (15 s em modo normal) e no melhor caso (60 s em modo *Modem Sleep*), os resultados da Tabela I comprovam que os desenvolvimentos deste trabalho permitiram uma redução de até 54,5% no consumo de energia médio. Se o dispositivo

fosse utilizado numa aplicação com tempo de transmissão muito maior, por exemplo de hora em hora ou até poucas vezes ao dia, seria possível reduzir o consumo médio para praticamente o valor de 100 mW obtido quando em modo *Modem Sleep*.

É importante destacar que neste trabalho o dispositivo de medição desenvolvido é composto por microturbina, sensor de vazão, módulo microcontrolador e circuito eletrônico adicional. Dessa forma, os resultados apresentados são relacionados a essa composição. No caso dos módulos NodeMCU e Devkit usados, os mesmos representam placas de desenvolvimento que possuem, respectivamente, os chips ESP8266 e ESP32. Ainda que os dados de catálogo desses chips tragam informações de consumo de energia muito menores que os obtidos nos experimentos, é importante esclarecer que é conhecido que outros componentes desses módulos são responsáveis pelo consumo de grande parte da energia requerida pelo módulo, principalmente o regulador de tensão e conversor serial para programação. No caso do circuito eletrônico adicional, responsável por compatibilizar os diferentes níveis de tensão elétrica dos componentes do dispositivo, como geração elétrica da microturbina, bateria e alimentação do módulo microcontrolador, este também é responsável por uma perda de energia não desprezível, conforme visto nas análises anteriores.

Dessa forma, para diminuir mais o consumo de energia e melhorar a eficiência energética do dispositivo de medição desenvolvido, seria indicado a remoção dos módulos NodeMCU e uso do chip ESP8266 direto, com uma significativa redução do consumo de energia do dispositivo.

9 CONCLUSÕES

A colheita de energia em pequenas quantidades através do fluxo hidráulico com a medição de vazão e com a utilização da energia recuperada para alimentar eletricamente um dispositivo IoT determinaram as premissas para construção de um protótipo para a avaliação do potencial de recuperação de energia.

O desenvolvimento de uma prova de conceito de um dispositivo de medição de vazão e volume com capacidade de colheita de energia hidráulica e comunicação de dados em nuvem validaram o desenvolvimento da plataforma de testes, comprovando a viabilidade de colheita de energia em pequenas quantidades a partir da vazão de água como opção para autossustentabilidade energética de dispositivos de IoT.

Diversos experimentos foram realizados com a medição da potência consumida pelo dispositivo e potência gerada pela microturbina e relação à vazão na tubulação do protótipo. Uma capacidade de colheita de energia de até 1 W foi obtida, demonstrando ser suficiente para alimentar o dispositivo de medição, ao mesmo em que se obteve energia suficiente para o recarregamento da bateria de suporte.

No decorrer dos experimentos buscou-se levantar o potencial de utilização da energia recuperada de maneira prática e considerando as várias condições de operação e funcionamento, através da análise dos dados obtidos em diferentes montagens da plataforma construída, a escolha dos materiais e suas especificações, a capacidade de vazão de uma moto bomba, circuitos eletrônicos necessários para o tratamento das grandezas elétricas de geração e consumo e o uso de diferentes dispositivos IoT, seguindo a evolução desses componentes.

Com a evolução dos resultados iniciais, pode-se aprimorar o aprendizado obtido e a inserção das ferramentas de software exigidas ao longo do trabalho, como a coleta e o armazenamento de dados utilizando LabVIEW e a posterior análise através de gráficos e dados numéricos extraídos, trazendo acuracidade aos resultados obtidos.

Os dispositivos dimensionados para a construção do circuito de energia mostraram-se adequados para obtenção de um sistema eficiente energeticamente. Foi necessário manter a menor quantidade possível de periféricos para realização da conversão de energia e, nesse sentido, os módulos escolhidos para conversão de nível lógico e regulação do nível de tensão não apresentaram consumo energético expressivo para dificultar a implementação do sistema de alimentação através da microturbina. Esta também se mostrou eficiente na geração de

energia, onde o valor mínimo de vazão necessário para iniciar o fornecimento de potência ao sistema é bastante abaixo dos padrões médios de vazão verificados em aplicações. O sensor de fluxo YF-S201 apresentou-se como uma alternativa simples, porém bastante eficiente, para a medição de vazão.

Para construção dos algoritmos responsáveis pela coleta de dados e comunicação, foram empregadas diversas ferramentas presentes na arquitetura de um microcontrolador. A principal delas utilizada no desenvolvimento do projeto é a interrupção em uma entrada digital. Esta foi empregada para conversão do sinal de saída do sensor de fluxo (na forma de pulsos digitais) na variável de vazão (em L/min). Além disso, através da medição contínua de vazão foi possível estimar o volume de água acumulado que já percorreu a tubulação em que o sistema está inserido.

Uma vez que o foco do projeto desenvolvido foi a análise da colheita de energia do fluxo hidráulico com a medição de vazão geradora da energia cinética, optou-se como elemento final de uso dessa energia, o emprego de uma estrutura de Internet das Coisas simplificada empregando a comunicação Wi-Fi. Para tal, foram utilizadas ferramentas de código aberto que apresentaram resultados bastante satisfatórios. Apesar de inicialmente proposta a utilização da plataforma Thinger.io, durante o desenvolvimento do projeto verificou-se o potencial da utilização da plataforma *ThingSpeak* e optou-se por empregá-la nos testes referentes à análise de viabilidade, uma vez que esta possui uma documentação mais completa e integração com o software MATLAB.

Em uma análise quantitativa, observou-se uma redução gradativa no consumo energético do sistema após a aplicação de cada um dos métodos propostos para sustentabilidade da solução. Ressalta-se que a utilização da taxa mínima de comunicação do *ThingSpeak* – referente a um dado a cada 15 s – em comparação com a comunicação a cada 1 s da plataforma Thinger.io resultou em uma redução no consumo de aproximadamente 41.36%. Importante ressaltar que a redução energética foi obtida mantendo-se uma taxa de transmissão relativamente alta e o módulo com o processador operando ininterruptamente.

O sistema de medição de vazão proposto se mostrou viável com os resultados obtidos, demonstrando que sua aplicação, tanto urbana como industrial são inúmeras e vão muito além do monitoramento do escoamento de fluidos nas instalações. Para a aplicação proposta, os modos de operação em baixa potência do módulo ESP8266 NodeMCU não puderam ser explorados por completo. Optou-se então para a utilização do dispositivo NodeMCU ESP32 com o modo operacional *Deep Sleep* mantendo o microcontrolador consumindo a menor

quantidade de energia possível e uma coleta contínua de dados, os quais tornam-se disponíveis para envio remoto em intervalos de tempo programáveis, buscando assim a melhoria contínua dos níveis de consumo da solução.

Nas etapas finais de testes com o dispositivo ESP 32, algumas adaptações foram necessárias, como por exemplo, o aumento da potência das baterias de suporte para os momentos iniciais de operação, porém compensados plenamente pelo recarregamento obtido neste componente.

Através da utilização de microcontroladores nos dispositivos com diferentes capacidades de processamento e modos de operação, os testes permitiram buscar a maximização da eficiência energética. Os resultados demonstraram que a operação do dispositivo com o Devkit ESP32 em modo *Deep Sleep* consome mais energia que com o NodeMCU Esp8266 em modo *Modem Sleep*. Os desenvolvimentos do trabalho permitiram a redução de até 54,5% do consumo de energia do dispositivo de medição desenvolvido, em relação à sua configuração inicial.

Por fim, os resultados obtidos e os comparativos dos resultados de geração e consumo de energia, e, ao mesmo tempo, mantendo-se a aquisição dos dados de vazão confiáveis, permite afirmar que o desenvolvimento e aplicação de uma solução visando a auto sustentabilidade no universo de IoT e IIoT são plenamente atingíveis e que podem tornar-se realidade a médio prazo.

10 REFERÊNCIAS BIBLIOGRÁFICAS

ABNT. (2017). “Norma ABNT NBR 12218:2017: Projeto de rede de distribuição de água para abastecimento público.” <https://www.abntcatalogo.com.br/norma.aspx?ID=37093>”.

ABNT. (1998). “Norma ABNT NBR 5626:1998: Instalação predial de água fria.” <https://www.abntcatalogo.com.br/norma.aspx?ID=4274>.

ALAREFI, S. A. S.; WALKER, S. D. (2015). “*Intelligent low-cost micro-hydro power emulator for domestic applications*”. Proceedings of the 3rd International Renewable and Sustainable Energy Conference (IRSEC), Marrakech, 2015, pp. 1-5.

BECKER P, FOLKMER B, GOEPFERT R, HOFFMANN D, WILLMANN A, MANOLI Y. (2013). “*Energy Autonomous Wireless Water Meter with Integrated Turbine Driven Energy Harvester*”.

CASINI, M. (2015). “*Harvesting energy from in-pipe hydro systems at urban and building scale*”. Department of Planning, Design, and Technology of Architecture (PDTA), Sapienza University of Rome, Via Flaminia 72, 00196 – Rome, Italy.

CHEN J, YANG H X, LIU C.P. (2013). “*A novel vertical axis water turbine for power generation from water pipelines*”. The Hong Kong Polytechnic University, Kowloon, Hong Kong, China, Water Supplies Department, The Government of the Hong Kong SAR, Chi.

CHU, S, GOLDEMBERG, J. ARUNGU-OLENDE, S.; EL-ASHRY, M.; DAVIS, G.; NAKICENOVIC, N., (2007). “*Lighting the Way: Toward a Sustainable Energy Future. Amsterdam*”:InterAcademyCouncil.Disponível em:<<http://www.interacademycouncil.net/File.aspx?id=24548>>.

CUNEFARE, K A; SKOW, E; ERTURK, A; SAVOR J; VERMA N; CACAN M R. (2013). “*Energy harvesting from hydraulic pressure fluctuations*”. G W Woodruff School of Mechanical Engineering, Georgia Institute of Technology, Atlanta, GA 30332-0405, USA - Smart Mater. Struct. 22.

EDNA. ELETRONIC DEVICES& NETWORK ANNEX. (2018). “*Energy Harvesting Technologies for IoT Edge Devices*”. *4E’s Electronic Devices and Networks Annex (EDNA)*. <https://edna.iea-4e.org/>

ENEOCEAN, (2015) – *White Paper Internet_ of_ Things_ EnOcean*. www.enocean.com. https://www.enocean.com/fileadmin/redaktion/pdf/white_paper/White_Paper_Internet_of_Things_EnOcean.pdf

ESP8266. (2017) Wikipédia. Disponível em: <<https://pt.wikipedia.org/wiki/ESP8266>>. Acesso: janeiro, 2017.

DARYI, WK; YANG HX, TONG JCK. (2016).” *Novel in-building hydro energy recovery turbine in operation.*” The 7th Greater Pearl River Delta Conference on Building Operation and Maintenance (2016)

FILHO, MF, MOELLWALD M.C.E. (2016).” *Internet das coisas*”. Livro digital; Universidade do Sul de Santa Catarina.: Unisul Virtual - Palhoça, (2016).

GILBERT, J M; BALOUCHI F. (2008). Comparison of Energy Harvesting Systems for Wireless Sensor Networks Department of Engineering, University of Hull, Hull HU6 7RX, UK October 2008, 334-347 DOI: 10.1007/s11633-008-0334-2.

HARVEY, C. (2014). 35 *Open Source Tools for the Internet of Things*. IT Bussines Edge. <<http://www.datamation.com/open-source/35-open-source-tools-for-the-internet-of-things-1.html>>.

HOFFMANN, D; WILLMANN, A; GOPFERT, P; BECKER, P; FOLKMER, B and MANOLI, Y. (2013). “*Energy Harvesting from Fluid Flow in Water Pipelines for Smart Metering Applications*.” Journal of Physics: Conference Series.

HUANG, J., MENG, Y., GONG, X., LIU, Y., & DUAN, Q. (2014). “*A novel deployment scheme for Green Internet of Things*”. IEEE Internet of Things Journal, 1(2), 196–205.

KAMALINEJAD, P.; MAHAPATRA, C.; SHENG, S.; MIRABBASI, S.; LEUNG, V. C. M.; GUAN, Y.L. (2015). “*Wireless energy harvesting for the Internet of Things*”. IEEE Communications Magazine, vol. 53, no. 6, pp. 102-108, June 2015.

KAYAALP, F; ZENGIN, A; KARA, R. (2014).” *Water Powered Sensors In Water Distribution Systems*”. Article · January 2014 <https://www.researchgate.net/publication/271646446>.

LIU, C. H.; FAN, J.; BRANCH, J.W.; LEUNG, K.K. (2014). “*Toward QoI and Energy-Efficiency in Internet-of-Things Sensory Environments*”. IEEE Transactions on Emerging Topics in Computing, vol. 2, no. 4, pp. 473-487, Dec. 2014.

LYDON, B. (2014).” *Internet of Things Industrial automation industry exploring and implementing IoT*”. Cover story, ISA. <<https://www.isa.org/standards-and-publications/isa-publications/intech-magazine/2014/mar-apr/cover-story-internet-of-things/>>.

MORAIS, R; MATOS, S; FERNANDES M; VALENTE A; SOARES S; FERREIRA, P, REIS. (2008). “*Sun, wind and water flow as energy supply for small stationary data acquisition platforms*”.Computers and electronics in agriculture 64 (2008) 120–132 <http://sweet.ua.pt/pjf/pdf/ferreira2008d.pdf>.

NODE-RED. *Ambiente de desenvolvimento para Internet das Coisas*. Disponível em: <http://nodered.org/>. Acesso: março, 2017.

PEREIRA, C.E.P; CARVALHO, F.V. (2017).” *A Internet das Coisas (IoT): Cenário e Perspectivas no Brasil e Aplicações Práticas*”. VII SRST – Seminário de Redes e Sistemas de Telecomunicações - INATEL ISSN 2358-1913

PORTINARI N. (2017) “*Sabesp planeja instalar mini- hidrelétricas em canos de água.*” Imprensa AESBE, Folha de São Paulo 06/10/2017. Acesso: março 2019.

SHAIKH, FK; ZEADALLY, S. (2015). “*Energy harvesting in wireless sensor networks: A comprehensive review.*” Renewable and Sustainable Energy Reviews 55 (2016) 1041–1054.

STANKOVIC, J.A. (2014). “*Research Directions for the Internet of Things*”. Internet of Things Journal, IEEE, vol.1, no.1, pp.3-9.

SUNDMAEKER, H; GUILLEMIN, P; FRIESS, P; WOELFFÉ, S. (2010).” *Vision and Challenges for Realising the Internet of Things*”. https://docbox.etsi.org/zArchive/TISPAN/Open/IoT/CERP-IOT_Clusterbook_2009.pdf

TAN YK, PANDA S K. (2010). “*Review of Energy Harvesting Technologies for Sustainable WSN*”. Sustainable Wireless Sensor Networks, Yen Kheng Tan (Ed.), ISBN: 978-953-307-297-5, InTech.

THINGER. *Plataforma para IoT*. Disponível em: <<https://thinger.io/>>. Acesso em: março, 2017.

THINGSPEAK. *Internet of Things Apps*. Disponível em: < <https://thingspeak.com>>. Acesso em: fevereiro, 2018.

YAN, T. C.; IBRAHIM, T.; NOR, N. M. (2011). “*Micro hydro generator applied on domestic pipeline*”. Proceedings of the 2011 International Conference on Electrical Engineering and Informatics, Bandung, 2011, pp. 1-6.

YFS201. *Datasheet do Sensor de Fluxo de Água 1/2*”. Disponível em: < http://img.filipeflop.com/files/download/Datasheet_YF-S201.pdf>. Acesso em: março, 2017.

WANG T., ZHANG Y. (2018). “*Design, Analysis, and Evaluation of a Compact Eletromagnetic Energy Harvester from Water Flow for Remote Sensors*”. Ocean College, Zhejiang University, Zhoushan 316000, China; mecontrol@126.com. Energies 2018, 11, 1424; doi:10.3390/en11061424