

Marcos José Rangel Gonçalves Junior

**Desenvolvimento de um medidor eletrônico de vazão sob o
paradigma da internet das coisas**

Campos Dos Goytacazes/RJ

2019

Marcos José Rangel Gonçalves Junior

**Desenvolvimento de um medidor eletrônico de vazão sob o
paradigma da internet das coisas**

Dissertação de Mestrado apresentada ao programa de pós-graduação do Instituto Federal de Educação, Ciência e Tecnologia Fluminense, no curso de Mestrado em Sistemas Aplicados à Engenharia e Gestão, como parte dos requisitos necessários à obtenção do título de mestre em Sistemas Aplicados à Engenharia e Gestão.

Instituto Federal de Educação Ciência e Tecnologia Fluminense

Programa de Pós-Graduação em Sistemas Aplicados à Engenharia e
Gestão

Orientador: Prof. D. Sc. Rogerio Atem de Carvalho

Campos Dos Goytacazes/RJ
2019

Biblioteca Anton Dakitsch
CIP - Catalogação na Publicação

G643d Gonçalves Junior, Marcos José Rangel
Desenvolvimento de um medidor eletrônico de vazão sob o paradigma da internet das coisas / Marcos José Rangel Gonçalves Junior - 2019.
43 f.: il. color.

Orientador: Rogerio Atem de Carvalho

Dissertação (mestrado) -- Instituto Federal de Educação, Ciência e Tecnologia Fluminense, Campus Campos Centro, Curso de Mestrado Profissional em Sistemas Aplicados à Engenharia e Gestão, Campos dos Goytacazes, RJ, 2019.
Referências: f. 42 a 42.

1. Medidor Eletrônico de Vazão. 2. Gestão de Consumo. 3. ESP8266. 4. Internet das Coisas. 5. IoT. I. Atem de Carvalho, Rogerio, orient. II. Título.

Marcos José Rangel Gonçalves Junior

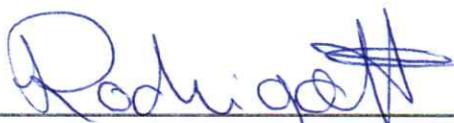
Desenvolvimento de um medidor eletrônico de vazão sob o paradigma da internet das coisas

Dissertação de Mestrado apresentada ao programa de pós-graduação do Instituto Federal de Educação, Ciência e Tecnologia Fluminense, no curso de Mestrado em Sistemas Aplicados à Engenharia e Gestão, como parte dos requisitos necessários à obtenção do título de mestre em Sistemas Aplicados à Engenharia e Gestão.

Trabalho aprovado. Campos dos Goytacazes/RJ, 24 de março de 2019:



Prof. DSc. Rogério Atem de Carvalho
Instituto Federal de Educação, Ciência e Tecnologia Fluminense
(Orientador)



Prof. Dsc. Rodrigo Martins Fernandes
Instituto Federal de Educação, Ciência e Tecnologia Fluminense



Dsc. Marcelo Patrício
Doutor em Engenharia Naval e Oceânica, USP

AGRADECIMENTOS

Gostaria de agradecer aos meus pais pelo amor e apoio incondicional em mais esta etapa.

Gostaria de agradecer especialmente a minha irmã, Jéssica, pela inconsequente ideia de me seguir pelo caminho da engenharia, mas principalmente pelo apoio técnico neste trabalho, sem ela este resultado não seria possível.

Gostaria de agradecer ao meu orientador, Prof. Rogerio Atem de Carvalho, pelas reuniões de orientação na cozinha durante o café, sem elas nada disso seria possível. Aos professores do programa, que foram importantes nesta jornada acadêmica e meu crescimento profissional.

Aos amigos professores e bolsistas do laboratório, agradeço a todos vocês pelo caos que tornaram alguns dos meus dias, mas principalmente pelas ideias que me acenderam uma luz em momentos de desespero criativo.

A toda equipe do Polo de Inovação Campos dos Goytacazes que continuou me acolhendo e me permitiram evoluir ainda mais.

A todos que direta ou indiretamente fizeram parte desta etapa da minha vida.

Resumo

É crescente a preocupação mundial com a maneira como a sociedade consome recursos naturais. Esta preocupação vai de encontro com temas atuais como cidades inteligentes e Internet das Coisas, uma vez que com todos os aspectos da vida em sociedade sendo monitorados e estando conectados torna-se mais fácil o gerenciamento do consumos desses recursos naturais.

Um dos recursos mais importantes e que está ligado a vida em sociedade e a qualidade de vida do ser humano é a água. Ela é usada desde a agricultura que produz os alimentos necessários, passando pelas indústrias que produzem os bens que necessitamos e desejamos e por fim chegando ao consumo em si pelo ser humano.

O Brasil, mesmo tendo posição de destaque quanto a disponibilidade de água doce, experimentou em 2014 os efeitos da falta deste importante recurso, evidenciando mais uma vez a necessidade de evitar desperdícios e gerenciar bem seu consumo.

Visando fornecer uma ferramenta de controle economicamente atrativa e de fácil integração, este trabalho se propõe a desenvolver um medidor eletrônico do consumo de água capaz de fornecer dados de consumo de maneira remota e autônoma. Sendo desenvolvida no Polo de Inovação Campos dos Goytacazes, esta ferramenta é baseada em sistemas anteriormente desenvolvidos pelo próprio Polo de Inovação mas tendo como diferencial ser baseado no paradigma da Internet das Coisas e seu custo de implementação significativamente menor, viabilizados pela utilização do microcontrolador ESP8266. Com este trabalho fornecendo ao final um dispositivo funcional baseado em protocolos de comunicação abertos e de fácil utilização.

Palavras-Chave: Medidor Eletrônico de Vazão, Gestão de Consumo, ESP8266, Internet das Coisas, IoT.

Abstract

Concern about the way how society consumes natural resources is growing worldwide. This concern runs counter to current themes such as smart cities and the internet of things, since with all aspects of society life being monitored and connected, it becomes easier to manage the consumption of these natural resources.

One of the most important resources that is linked to life in society and the life quality of the human being is water. It is used from the agriculture that produces the necessary foods, passing through the industries that produce the goods that we need and desire and finally arriving at the consumption itself by the human being.

Brazil, even having a prominent position regarding the availability of fresh water, experienced in 2014 the effects of the lack of this important resource, once again evidencing the need to avoid waste and manage its consumption well.

Aiming to provide an economically attractive and easy to integrate control tool, this paper proposes to develop an electronic water consumption meter capable of providing consumption data remotely and autonomously. Being developed in the Campos dos Goytacazes Innovation Hub, this tool is based on systems previously developed by the Innovation Hub itself, but based on the Internet of Things paradigm and its significantly lower implementation cost, made possible by the use of the ESP8266 microcontroller. With this work providing at the end a functional and friendly device based on open communication protocols.

Keywords: Electronic Flow Meter, Consumer Management, ESP8266, Internet of Things, IoT.

Lista de ilustrações

Figura 1: Protótipo do medidor de vazão original	18
Figura 2: Módulo SoC ESP-12E	20
Figura 3: Sensor de vazão digital de $\frac{3}{4}$ "	21
Figura 4: Placa de desenvolvimento NodeMCU.	23
Figura 5: Módulo RTC DS1307.	24
Figura 6: Módulo I2C LCD.	25
Figura 7: Ambiente de desenvolvimento em protoboard	25
Figura 8: Esquemático elétrico da PCB do medidor.	33
Figura 9: PCB do medidor sem os componentes.	33
Figura 10: Segundo protótipo do medidor sem o RTC soldado	34
Figura 11: Módulo RTC soldado a PCB do medidor	36
Figura 12: Gravação do firmware (MicroPython) no medidor.	37
Figura 13: Primeiro protótipo fechado do medidor	38
Figura 14: Sistema instalado junto a uma máquina de lavar	39

Lista de quadros

Quadro 1: Especificações dos sensores utilizados	21
--	----

Lista de tabelas

Tabela 1: Exemplo de dados coletados pelo medidor	39
---	----

Lista de abreviaturas e siglas

I2S	Integração e Inteligência para sustentabilidade
PICG	Polo de inovação Campos dos Goytacazes
IoT	Internet of things
RTC	Real Time Clock
MCU	Microcontrolador
GPIO	General purpose input output
SRAM	Static Random Access Memory
UART	Universal Asynchronous Receiver/Transmitter
SPI	Serial Peripheral Interface
I2C	Inter-Integrated Circuit
IR	Infrared
ADC	Analog to digital converter
PWM	Pulse width modulation
ROM	Read only memory
PCB	Printed circuit board
I/O	Input/output
CI	Circuito integrado
MQTT	Message Queuing Telemetry Transport
NTP	Network Time Protocol
EEPROM	Electrically-Erasable Programmable Read-Only Memory
FIFO	First in first out
RF	Radiofrequência

Sumário

1. Introdução	12
1.1. Contextualização	12
1.2. Objetivo	14
1.2.1. Objetivo Geral	14
1.2.2. Objetivos Específicos	14
1.3. Estrutura do Trabalho	14
2. Revisão Patentária	16
3. Metodologia	17
3.1. Descrição do smart meter original	17
3.2. Descrição do Novo Sistema	18
3.3. Escolha do hardware base	19
3.4. Escolha do elemento sensor	21
3.5. Criação do ambiente de desenvolvimento	22
3.6. Desenvolvimento dos códigos embarcados	26
3.7. Testes de Bancada do Sistema	30
3.8. Desenvolvimento da placa de circuito	31
3.9. Desenvolvimento do invólucro de proteção	34
3.10. Montagem do protótipo	35
3.11. Testes de Campo do sistema	38
4. Resultados e Considerações Finais	40

1. Introdução

1.1. Contextualização

O Brasil é um dos países com maior disponibilidade de água doce do mundo. Apesar da posição privilegiada, a crise hídrica de 2014 expôs a necessidade de se gerenciar melhor este recurso, obrigando os consumidores a se reeducar e mudar seus hábitos. Essa reeducação fez com que a população reduzisse o consumo de forma instintiva, pois na maioria dos consumidores de médio e pequeno porte não existe uma estrutura que permita o controle extensivo do volume de água, porém, o consumo de água no Brasil aumentou em 80% nos últimos 20 anos e a tendência é que aumente em mais 30% até 2030 (ANA, 2017).

A importância de gerenciar de forma eficiente os recursos hídricos não se limita a disponibilidade em si de água para o consumo. Mesmo no campo energético, com o Brasil sendo considerado o país com a matriz energética mais limpa do mundo, mais de 70% de sua produção proveniente de fontes como recursos hídricos (EOZAWA, 2010).

Observando esta demanda, os medidores inteligentes de vazão e energia elétrica desenvolvidos pelo Laboratório de Energias Renováveis, Eficiência Energética e Biocombustíveis são dois dispositivos desenvolvidos originalmente para operar como soluções simples de gerenciamento de consumo com o objetivo de dar a gestores ferramentas que viabilizassem esse gerenciamento em tempo real de forma intuitiva (GONÇALVES JUNIOR, 2016).

Estes medidores tinham como plataforma base o Raspberry Pi, que é um single board computer, que possibilita aos medidores terem como principal diferencial a sua integração com o Google Drive, plataforma online e gratuita onde os dados ficam disponíveis em forma de planilhas e gráficos. Outros diferenciais dos medidores eram a possibilidade de fazer backup local do histórico de consumo em banco de dados, além da visualização local através de um display LCD (GAMA e RANGEL, 2017).

Os dois sistemas desenvolvidos foram inicialmente projetados para operarem isolados, sem a necessidade de nenhuma outra infraestrutura de suporte,

porém existe potencial para que esses dispositivos se integrem em sistemas maiores de gerenciamento, como o I2S, por exemplo.

O I2S (Integração e Inteligência para Sustentabilidade) é um projeto de integração, que visa tornar o campus do Polo de Inovação de Campos dos Goytacazes (PICG) em um campus inteligente utilizando a tecnologia. O sistema do I2S será responsável por coletar e monitorar os dados de subsistemas instalados no PICG, integrando os dados referentes a geração e consumo de recursos, fazendo seu tratamento e disponibilizando esses dados em um painel de monitoramento instalado no campus, além de armazenar os dados coletados em um banco de dados semântico para que futuramente seja possível a realização de pesquisas com os dados gerados, como análise de tendências, eventos específicos e suas correlações (RIBEIRO, CAVADAS, CARVALHO e VIANNA, 2017).

No I2S todos os subsistemas de geração e consumo serão conectados a uma central de tratamento de dados, que terá a função de fazer a aquisição dos dados dos subsistemas e tratá-los de modo a enviá-los para a central de monitoramento.

Um dos subsistemas do I2S é o de monitoramento de consumo de recursos, este subsistema é responsável pelo monitoramento do consumo de energia elétrica e água (RIBEIRO, 2018). O monitoramento neste subsistema será feito pelos medidores inteligentes desenvolvidos pelo próprio polo de inovação.

Para um controle mais preciso, quanto maior a quantidade de pontos monitorados, melhor. Neste sentido, a principal dificuldade em adotar os medidores desenvolvidos é seu custo. O medidor de vazão desenvolvido em 2016 teria um custo de fabricação atualizado em 2018 de aproximadamente R\$ 390,00 por unidade.

Visto que os medidores já desenvolvidos precisarão de qualquer forma passar por adaptações para integrar-se ao I2S, vislumbrou-se então a possibilidade de desenvolver um novo dispositivo de medição que atendesse às necessidade do I2S, mas que tivesse também um custo de produção bem menor. Este dispositivo deveria ser inspirado no paradigma da Internet das Coisas (IoT - *Internet of Things*) e utilizar o protocolo de comunicação MQTT (*Message Queuing Telemetry Transport*).

MQTT é um protocolo de mensagens extremamente simples e leve de publicação / assinatura, projetado para dispositivos restritos e redes de baixa largura de banda, alta latência ou não confiáveis. Os princípios de design são minimizar a largura de banda da rede e os requisitos de recursos do dispositivo, ao mesmo tempo em que tentam garantir a confiabilidade e um certo grau de garantia de entrega. Esses princípios também tornam o protocolo ideal para o emergente mundo de dispositivos conectados, (Internet das Coisas) e para aplicações móveis onde a largura de banda e a bateria são preciosas (MQTT, [s.d.]).

O nome Internet das Coisas tem a origem atribuída a Kevin Ashton, que em 1999 fez uma apresentação com este mesmo nome a empresa Procter & Gamble (P&G) e é um conceito em que todos os objetos da vida cotidiana estariam conectados e sendo capazes de mensurar tudo e enviar estas informações em uma rede de computadores nos permitiria reduzir custos, perdas e desperdícios (ASHTON, 2009).

Isto posto, este trabalho propõe-se a desenvolver um novo dispositivo de medição do consumo de água, baseado no dispositivo já desenvolvido pelo PICG, visando integrar o mesmo ao I2S além de reduzir seu custo de produção a fim de facilitar sua utilização no próprio I2S em outros projetos, este projeto se justifica como uma evolução dos dispositivos desenvolvidos anteriormente.

1.2. Objetivo

1.2.1. Objetivo Geral

Desenvolver um novo sistema de monitoramento de água baseado nos sistemas desenvolvidos no PICG com um custo-meta de R\$ 100,00 de produção do protótipo por cada unidade.

1.2.2. Objetivos Específicos

- Desenvolver o novo *hardware* de medição e comunicação do medidor, capaz de medir e enviar os dados para um servidor remoto.

- Desenvolver um dispositivo capaz de se comunicar com rede via cabo e via WI-FI.
- Desenvolver o *software* de medição embarcado no sistema
- Construir 5 protótipos para testes
- Testar e validar o sistema operando integrado ao I2S

1.3. Estrutura do Trabalho

- Capítulo 1: Apresenta uma introdução do contexto em que a pesquisa será realizada, assim como os objetivos da pesquisa e sua justificativa.
- Capítulo 2: Compreende a revisão patentária das principais soluções desenvolvidas e presentes no mercado.
- Capítulo 3: Descreve a metodologia utilizada para a realização da pesquisa, detalhando todas as etapas envolvidas.
- Capítulo 4: Apresenta os resultados obtidos durante o desenvolvimento do projeto.
- Capítulo 5: Apresenta as referências bibliográficas utilizadas para a elaboração deste trabalho.

2. Revisão Patentária

A busca por patentes foi feita na base *Web of Science* e os documentos relevantes a este trabalho serão descritos abaixo nos moldes de uma solicitação de patente da forma como indica o INPI.

O termo utilizado na busca foi ((meter OR measure*) AND (flow* OR water*) AND (Cloud OR web OR 'remote server') AND 'turbine') refinando pela área de conhecimento (INSTRUMENTS INSTRUMENTATION) retornando um total de 25 documentos com os mais relevantes descritos abaixo.

O documento US2011307101-A1 depositado em 25/08/2011 e publicado em 15/12/2011 intitulado “*Method of altering operating condition of device e.g. smart appliance located at e.g. residential site, involves detecting availability of control action data at remote server in conjunction with communication of device data*”, refere-se a um método de conexão com dispositivos localizados em algum lugar enviando seus dados para um servidor remoto. O método consiste em um arranjo de sistema que se conecte a pelo menos um dispositivo de aquisição de dados que seja capaz de enviar seus dados adquiridos para um servidor remoto e de detectar ações de controle quando disponíveis. Portanto, o documento US2011307101-A1 não sugere nem antecipa o dispositivo em desenvolvimento.

O documento US2017059543-A1 depositado em 10/09/2016 e publicado em 02/03/2017 intitulado “*Water sensing assembly for measuring properties of water in water distribution system and managing measurement data in water utility companies, has electrical communication device mounted at top portion of adjustable top section*”, refere-se a um sistema e métodos de monitoramento de parâmetros da água em um sistema de distribuição, o sistema possui um elemento intermediário que comprime os dados e os disponibiliza através do acesso a uma interface WEB local. Portanto, o documento US2017059543-A1 não sugere nem antecipa o dispositivo em desenvolvimento.

Os demais documentos não possuíam qualquer relação com dispositivo proposto.

3. Metodologia

3.1. Descrição do *smart meter* original

O dispositivo de medição desenvolvido originalmente utilizava um sensor de vazão tipo turbina por efeito hall como elemento sensor que modulava a vazão em frequência, o componente principal do sistema era um mini PC Raspberry Pi 3 que era responsável pela aquisição do sinal do sensor, pela medição da frequência e cálculo da vazão. O sistema ainda possuía uma outra placa de circuito responsável pela alimentação do sistema, RTC externo, ajuste do contraste do display LCD e integração de todos os periféricos no Raspberry Pi. O último componente era um display LCD para apresentação local dos dados e auxílio na configuração.

Funcionando no Raspberry Pi ainda existia um banco de dados MySQL e um servidor WEB Apache 2. O princípio de funcionamento era relativamente simples, o Raspberry Pi era responsável pela medição da vazão e cálculo do volume acumulado e a cada intervalo de tempo previamente configurado o sistema armazenava a vazão média e o volume no banco de dados interno. Paralelamente funcionavam outras duas rotinas, uma que enviava esses dados colhidos para um serviço em nuvem (Google Drive) usando o banco de dados como um buffer e outro que removia os dados do banco de dados e armazenava-os em arquivos CSV separados por dia sendo todas essas funções implementadas em Python.

Por último, existia o servidor WEB por onde era possível configurar o sistema e ter acesso direto aos histórico de dados salvos nos arquivos e no banco de dados e uma última rotina que controlava o que era mostrado no display LCD.

A Figura 1 mostra o medidor de vazão original baseado no Raspberry Pi.

Figura 1: Protótipo do medidor de vazão original



Fonte: Autor

3.2. Descrição do Novo Sistema

O novo medidor deve manter o maior número de funcionalidades possível sem estourar o custo-meta, sendo fundamentais dentre as existentes no sistema antigo a de medição instantânea, a de envio dos dados com *timestamp* e a de configuração. As demais funções seriam implementadas dependendo da capacidade do hardware escolhido, sendo consideradas desejáveis: cálculo e armazenamento do volume total acumulado, comunicação com a rede cabeada ou sem fio, display para apresentação local dos dados, capacidade de operar sem estar conectado à rede e interface remota de acesso dos dados.

Como a limitação deste projeto é seu custo final de produção, a concepção original envolvia uma solução modularizada de olho na maior redução do custo possível. Essa modularização seria caracterizada por um dispositivo básico que possuísse as funções essenciais citadas, sendo que o custo-meta (*target cost*) de R\$ 100,00 seria aplicado a este dispositivo básico, mas seriam também projetados módulos que adicionariam funções a esse dispositivo a medida que fossem necessárias. O *target cost* R\$ 100,00 se justifica pelo fato de um hidrômetro convencional custar na faixa dos R\$ 70,00, sendo este um dispositivo puramente mecânico e que somente permite medição in loco.

Contudo, conforme será apresentado nos resultados, esta ideia foi abandonada em função das capacidades operacionais e do custo do hardware escolhido e dos demais periféricos.

3.3. Escolha do hardware base

Como dito, o primeiro medidor tinha como base o Raspberry Pi na versão 3, que é um mini PC que executa um sistema operacional, possui um processador de 4 núcleos rodando a 1.2GHz e tem 1GB de memória principal, mas que tem custo muito alto e que encarece o dispositivo final.

O hardware escolhido para ser a base do novo sistema é módulo ESP-12 cujo componente principal é o MCU ESP8266. ESP8266 é um microcontrolador que possui conectividade WI-FI integrada com bom gerenciamento de energia, tendo sido projetado especialmente para aplicações de Internet das Coisas (ESPRESSIF SYSTEMS, 2018). Possui um processador Tensilica L106 Diamond de 32-bits e memória do tipo SRAM integrada, além de poder ser integrado com sensores e outros dispositivos através de suas GPIOs.

Abaixo, algumas características do ESP8266.

Especificações:

- Clock: 80 MHz ~ 160MHz
- Memória principal: 50kB
- Tensão de operação: 2.5V ~ 3.6V
- Corrente média: 80mA
- Temperatura de operação: -40°C ~ 125°C
- Interfaces de entrada e saída: UART, SPI, I2C, IR Remote Control, GPIO, ADC, PWM

Não existe uma memória ROM interna no ESP8266, o programa a ser executado deve ser gravado em uma memória Flash SPI externa.

Características do WI-FI:

- Padrão WI-FI: 802.11 b/g/n
- Frequência: 2.4G ~ 2.5G
- Taxa de transferência: Até 72.2 Mbps
- 2 x Interface WI-FI virtual
- Protocolos de rede: IPv4, TCP/UDP/HTTP

A Figura 2 exemplifica o módulo ESP-12E, componente principal do novo sistema em desenvolvimento.

Figura 2: Módulo SoC ESP-12E



Fonte: Autor

Como dito, o ESP8266 é o MCU presente no módulo ESP-12E que será usado neste projeto. O módulo ESP-12E é um módulo que integra além do ESP8266, uma memória Flash necessária para armazenamento do código, e tem ao PCB do módulo uma antena para a comunicação com a rede WI-Fi, reduzindo o esforço no projeto de projetar nas mesmas.

Como é baseado no ESP8266 suas especificações são as mesmas, a não ser pela memória Flash que no modelo 12E é de 4MB e pela antena WI-FI integrada de 2dBi (AI-THINKER TECHNOLOGY CO., 2017).

Outra informação importante para o desenvolvimento do projeto é que o ESP-12E é encontrado no Brasil na faixa dos R\$ 20,00, motivo principal da sua escolha.

3.4. Escolha do elemento sensor

O sensor de vazão escolhido é o mesmo utilizado na primeira versão do medidor e pode ser visto na Figura 3 na versão de ½". Trata-se de um sensor de vazão tipo turbina fabricado na China que modula vazão em frequência.

Figura 3: Sensor de vazão digital de ½"



Fonte: Seed Studio WIKI

A tabela abaixo demonstra as diferentes bitolas encontradas desse sensor e suas especificações.

Quadro 1: Especificações dos sensores utilizados

	G ½"	G ¾"	G 1"	G1 ¼"
Tensão mín. de funcionamento	DC 4.5V	DC 4.5V	DC 4.5V	DC 4.5V
Corrente de operação	15mA(DC 5V)	15mA(DC 5V)	15mA(DC 5V)	15mA(DC 5V)
Tensão de operação	5V~24V	5V~24V	5V~24V	5V~24V
Faixa de vazão	1~30L/min	1~60L/min	1~100L/min	1~120L/min
Temperatura de operação	≤80°C	≤80°C	≤80°C	≤80°C

Temperatura do Líquido	$\leq 120^{\circ}\text{C}$	$\leq 120^{\circ}\text{C}$	$\leq 120^{\circ}\text{C}$	$\leq 120^{\circ}\text{C}$
Pressão da água	$\leq 2.0\text{MPa}$	$\leq 2.0\text{MPa}$	$\leq 1.75\text{MPa}(\text{Max } 2\text{MPa})$	$\leq 2.0\text{MPa}$

Fonte: Seed Studio WIKI

Importante destacar que, o nível lógico de funcionamento do Raspberry Pi é 3,3V, da mesma forma que o ESP8266, e que o sensor usado modula a vazão em frequência mas com o nível lógico igual à tensão de alimentação. Apesar de as especificações indicarem que a tensão mínima de funcionamento destes sensores é de 4,5V, estes foram testados em 3,3V e seu funcionamento continua normal. Desta forma, nenhum tipo de adequação do sinal é necessária e o sensor pode funcionar ligado diretamente ao microcontrolador, da mesma forma que o medidor original (GONÇALVES JUNIOR, 2016).

Esta afirmação é confirmada pela análise construtiva do sensor, que possui apenas um ímã permanente preso diretamente ao eixo da turbina. O circuito eletrônico do sensor é constituído apenas de um sensor de efeito hall WSH130 que fica alinhado com o campo magnético do ímã do rotor e um resistor pull-up. A faixa de operação do WSH130 vai de 2,4V até 26V (©WINSON, 2017).

3.5. Criação do ambiente de desenvolvimento

Para o desenvolvimento do sistema, é necessário uma plataforma de desenvolvimento que permitisse o rápido uso do ESP-12E, sem que fosse necessário neste momento desenvolver circuitos de alimentação e programação com o módulo.

Para tanto foi escolhido o NodeMCU, visto na Figura 4, como plataforma de desenvolvimento. O NodeMCU é uma plataforma de desenvolvimento similar ao Arduino, mas tendo como diferencial ser baseado no módulo ESP-12 e, portanto, já ter comunicação WI-Fi integrada.

Figura 4: Placa de desenvolvimento NodeMCU.



Fonte: Autor

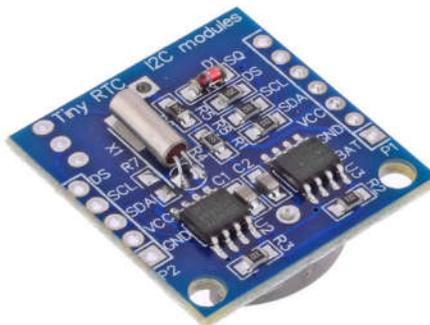
A disposição das I/Os do NodeMCU torna-o capaz de ser encaixado em uma protoboard, desta forma facilitando a elaboração do circuito do novo medidor e os testes de comunicação com os demais periféricos.

O ESP-12E já é por conta própria capaz de executar as funções de medição e envio, devido as suas I/Os e conectividade WI-FI, porém, entre a comunidade de desenvolvedores é sabido que o relógio interno do ESP8266 é muito impreciso, conforme um teste realizado que consistiu primeiramente em sincronizar o relógio interno do ESP8266 e o relógio do PC utilizado no teste. Em seguida o ESP8266 foi programado para ficar informando intervalos de 5 segundos.

O teste foi iniciado com os dois relógios marcando 08 horas e 07 minutos. O teste foi encerrado 13:35 no horário marcado pelo PC enquanto que no horário marcado pelo ESP8266 indicava 13 horas e 21 minutos, um atraso de 14 minutos em apenas 5 horas de teste.

Não existindo a possibilidade de contar apenas com o relógio interno do ESP8266, fez-se necessário a adição de circuito RTC (Real Time Clock) externo. Devido ao custo e a familiaridade com o RTC, dois CIs (circuitos integrados) foram pré escolhidos, o DS1303 e o DS1307 da Maxim Integrated, ambos com mesmo funcionamento, sendo RTCs de baixo consumo capazes de armazenar de segundo até anos com seu uso válido até 2100 (MAXIM INTEGRATED PRODUCTS, 2015) tendo como diferença apenas a interface de comunicação. No fim foi escolhido o DS1307 que utiliza I2C como interface de comunicação. Na figura 5 é possível ver um módulo RTC usado no desenvolvimento e baseado no DS1307.

Figura 5: Módulo RTC DS1307



Fonte: Autor

Devido ao baixo custo do sensor, na faixa dos R\$ 35,00, e do ESP-12E, na faixa dos R\$ 20,00, decidiu-se que a versão básica do sistema contaria com um display para visualização local dos dados.

O medidor original utilizava um display LCD de 16x2 caracteres que é um display que atendia as necessidades de visualização local dos valores de medição e para configuração inicial do sistema e custava pouco frente a outros displays possíveis de ser utilizado, encontrado por aproximadamente R\$ 16,00. Infelizmente esse display necessita de 6 portas digitais para funcionar conectado diretamente ao MCU, o que tornaria inviável utilizá-lo com o ESP-12E.

Felizmente, existe um módulo capaz de fazer a comunicação do display com o MCU utilizando a interface I2C, esse módulo pode ser visto na Figura 6 e como a I2C é uma interface de comunicação que funciona em barramento, nenhum I/O a mais seria necessária para utilizá-lo visto que já existiria um barramento I2C para o RTC. Este é um módulo fácil de encontrar no mercado e muitas vezes é vendido já integrado ao display LCD sem aumentar muito o preço final.

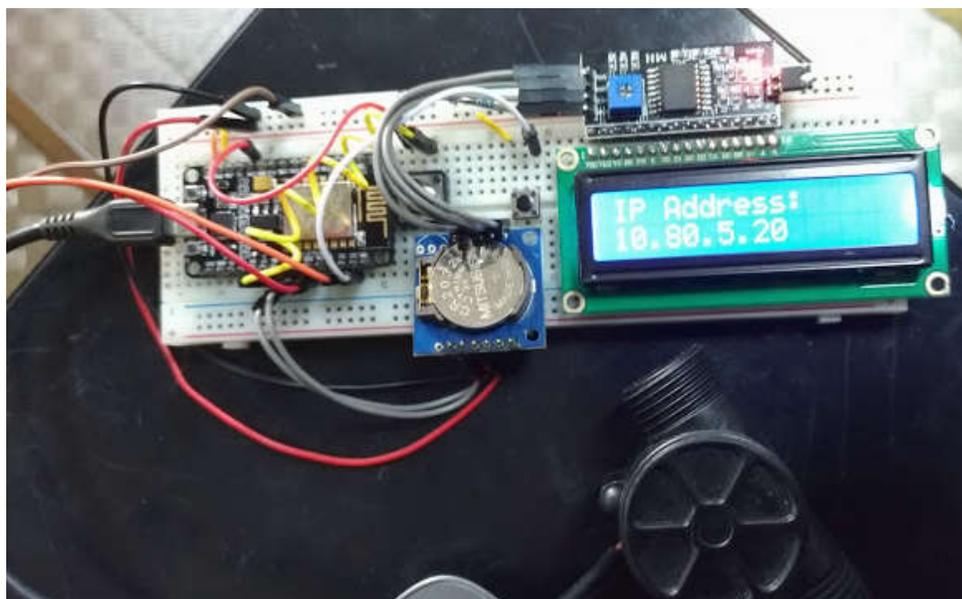
Figura 6: Módulo I2C LCD.



Fonte: Autor

Definidos os principais componentes de hardware, todos os periféricos foram conectados a placa de desenvolvimento NodeMCU por meio de uma protoboard e interligados por fios tal qual pode ser visto na Figura 7.

Figura 7: Ambiente de desenvolvimento em protoboard



Fonte: Autor

3.6. Desenvolvimento dos códigos embarcados

Existem algumas possibilidades de se programar o ESP8266, utilizando a IDE nativa disponibilizada pelo fabricante, utilizando Lua pela interface do NodeMCU, programar pela mesma IDE do Arduino utilizando C ou utilizar MicroPython.

MicroPython é uma implementação enxuta da linguagem de programação Python 3, possui algumas de suas bibliotecas padrão e é otimizada para ser embarcada em microcontroladores (GEORGE, [S.d.]).

Um dos MCUs para o qual o MicroPython foi otimizado é o ESP8266 e devido a familiaridade com a linguagem de programação, além da compatibilidade de código entre as duas versões do medidor, a original e a em desenvolvimento, o MicroPython foi a linguagem escolhida para o desenvolvimento deste trabalho.

Tal qual o Python comum, o MicroPython também é uma linguagem que utiliza a figura dos scripts, uma vez que, não é necessária recompilar o código embarcado a cada modificação. Com o Python, especificamente neste caso o MicroPython, existe um interpretador embarcado que acessa a memória Flash presente no módulo ESP-12E e executa os scripts, desta forma o interpretador só precisa ser carregado no ESP-12E apenas uma vez.

O carregamento do MicroPython no ESP-12E presente no NodeMCU pode ser feito através de um computador com Windows ou Linux e no desenvolvimento deste projeto foi utilizado um PC rodando o sistema Ubuntu na versão 16.04. Na documentação oficial online do MicroPython para o ESP8266 é indicado utilizar uma ferramenta chamada esptool (“Quick reference for the ESP8266 — MicroPython 1.9.4 documentation”, [S.d.]) que pode ser instalada utilizando o pip pelo comando:

```
~ pip install esptool
```

Uma vez baixado o arquivo binário do MicroPython, para instalá-lo é preciso primeiro apagar todo conteúdo da memória Flash antes da instalação, de acordo

com a própria documentação oficial. Para apagar a memória Flash e carregar o arquivo binário foram usados os seguintes comandos:

```
~ esptool.py --port /dev/ttyUSB0 erase_flash  
~ esptool.py --port /dev/ttyUSB0 --baud 115200 write_flash --flash_size=detect -fm  
dio 0 esp8266-20171101-v1.9.3.bin
```

A entrada e saída padrão do MicroPython é a interface UART 1 do ESP8266 e para ter acesso a ela é necessário algum conversor que faça o PC se comunicar via UART, como um conversor USB x Serial por exemplo. O NodeMCU utilizado no desenvolvimento dos scripts embarcados possui um desses conversores. O modelo varia de acordo com a versão do NodeMCU, mas o utilizado possui o conversor CP2101 integrado e para comunicar o PC com o ESP-12E basta utilizar um cabo micro-USB.

Com o NodeMCU ligado ao PC é necessário utilizar um monitor serial, neste caso o picocom, e através dele é possível ter acesso ao terminal do MicroPython sendo executado dentro do ESP8266, que é idêntico ao terminal do Python padrão.

O terminal é útil para a testes de pequenos trechos de códigos, mas a sua principal utilidade neste trabalho é para a execução e testes dos scripts armazenados na memória interna.

No desenvolvimento deste sistema os scripts ou parte deles que controlam cada periférico ou interface de comunicação foram desenvolvidos ou modificados a partir de versões abertas de código disponíveis na internet de forma que pudessem ser testados separadamente.

A primeira função implementada foi a de medição da vazão com o sensor ligado diretamente ao NodeMCU e alimentado com 3.3V, as ligações elétricas serão demonstradas na sequência deste trabalho com o detalhamento do desenvolvimento do diagrama elétrico do sistema, porém esta função não foi desenvolvida do zero, por se tratar do mesmo sensor e da mesma linguagem de programação, parte do código foi importado da primeira versão do medidor embarcado no Raspberry.

Trata-se de um script que define duas interrupções no processador do sistema, uma externa atrelada ao pino de sinal do sensor e outra por tempo

configurada pelo tempo de amostragem definido pelo usuário do sistema. A interrupção externa dispara uma função que apenas aumenta em uma unidade um contador referente ao volume passado pelo sensor. As informações encontradas online sobre outras aplicações deste sensor são feitas contando o tempo entre os pulsos enviados pelo sensor, calculando a frequência do sinal e conseqüentemente a vazão, mas por motivos diferentes nem o Raspberry Pi nem o ESP8266 possuem formas confiáveis de contar estes intervalos de tempo de forma que isso poderia influenciar nas medições e para mitigar essas possíveis interferências as interrupções apenas contam os pulsos e posteriormente outra parte do código controlada pela interrupção por tempo converte o valor deste contador para o valor de vazão média. As únicas modificações necessárias foram quanto às definições das interrupções e a interface física do sensor.

Em seguida foi codificada a comunicação com o servidor MQTT que foi o protocolo de comunicação escolhido para este projeto e que também atende às necessidades do I2S, devido às próprias finalidades para o qual o ESP8266 foi desenvolvido. Existe nativamente no MicroPython uma biblioteca para comunicação MQTT que facilitou o desenvolvimento, sendo necessário apenas um pequeno código que abra a comunicação com o Broker MQTT e que publique as mensagens contendo o timestamp, vazão média no último intervalo de medição e o volume total acumulado.

Os dois periféricos principais, o driver I2C-LCD e o RTC DS1307 possuem bibliotecas abertas disponíveis online e testadas com micropython e o ESP8266. A biblioteca do LCD não sofreu qualquer alteração enquanto a do RTC que foi escrita para funcionar com mais de um modelo de RTC foi simplificada para funcionar apenas com o DS1307.

Como este é um dispositivo que pode estar conectado à internet, optou-se por implementar uma pequena rotina que sincroniza o relógio interno tanto do ESP8266 quanto do DS1307 com o relógio global utilizando o protocolo NTP (*Network Time Protocol*), que é um protocolo que permite a sincronização dos relógios dos dispositivos de uma rede como servidores, estações de trabalho, roteadores e outros equipamentos à partir de referências de tempo confiáveis. Este script recebe como parâmetro o endereço do servidor NTP de onde se deseja

adquirir a data correta e o fuso horário, ele adquire a data e hora e as corrige segundo o fuso horário indicado sem considerar horário de verão, que deve ser ajustado manualmente.

Visto que um dos requisitos do sistema é funcionar mesmo sem estar conectado a um servidor para receber os dados, é necessário uma estrutura de dados que armazene os dados coletados mas não enviados. No medidor original esse mecanismo era gerenciado por um banco de dados instalado no Raspberry Pi, mas com o ESP8266 não existe esta possibilidade, então no lugar de um banco de dados um buffer em memória foi usado no lugar na forma de uma fila do tipo FIFO (*first in first out*). Apenas um buffer em memória não seria suficiente para armazenar dados de medição por muito tempo e o ideal seria utilizar a memória Flash como SWAP, no entanto, essas memórias Flash de 4MB utilizadas no módulo ESP-12E tem o ciclo de vida de 100.000 escritas e utilizar o SWAP de memória faria ela se deteriorar muito rápido e o tempo que levaria dependeria de quanto tempo o sistema ficasse “offline”. Disto isto, a memória Flash é usada apenas para armazenar os dados que estão em memória em intervalos de um minuto, apenas para que esses dados não sejam perdidos em caso de desligamento do sistema.

Pelo mesmo motivo que a memória Flash não é usada como SWAP o tempo inteiro, o volume total medido pelo sistema também não é armazenada na memória Flash, sendo armazenado no lugar da Flash do ESP-12 em uma outra memória EEPROM presente no módulo RTC. O CI AT24C32 é uma EEPROM de 32Kb presente no módulo RTC usado no desenvolvimento, é uma memória que utiliza a interface I2C ligada diretamente no barramento onde também está o DS1307 e diferente da Flash interna do ESP-12E tem um ciclo de reescritas de 1 milhão de vezes. Para comunicar o ESP8266 com a AT24C32 foi usado uma biblioteca aberta disponível online.

O script de medição é o script que todos os trechos de código e bibliotecas reunidos a fim de fazer o sistema funcionar, este funciona da seguinte forma: define as classes de configuração que acessa a Flash e carrega os parâmetros do sistema de um arquivo JSON; define a classe sensor onde estão o contador de pulsos e a função de conversão dos pulsos em volume; define a classe MQTT que trata da comunicação com o servidor MQTT; define a classe MCU que trata de todos os

parâmetros referentes a entradas, saídas e interrupções do ESP8266 bem como a comunicação WIFI; instância a classes definidas neste código e que tratam da fila FIFO; define as funções a serem executadas pelas rotinas de interrupção externa e por tempo; define a função principal do código que abre a comunicação com o display, recupera dados a serem enviados dos arquivos CSV caso existam, recupera o valor total acumulado medido, abre a comunicação WIFI, ajusta a data e hora usando o NTP e por fim inicializa o main loop que controla o que é mostrado no display e esvazia a fila de dados caso exista algum dado nela.

Todas essas funções implementados ocupam sozinhas bastante memória e ainda é necessário implementar uma interface de configuração para o usuário do sistema. o código chamado “main.py” que é o segundo código que por padrão é executado pelo MicroPython após o “boot.py” assumiu a função de gerenciar em qual modo de utilização o sistema é iniciado e utilizando os botões físicos presentes no sistema carrega na memória e inicializa o código de medição ou de configuração.

O script de configuração é um script que abre a interface WIFI do ESP8266 em modo AP, ou seja, é necessário se conectar diretamente ao dispositivo e mediante IP padrão acessar uma página WEB onde é possível alterar os parâmetros de funcionamento e zerar o buffer de volume.

Além destes script relacionados diretamente ao funcionamento do sistema existem mais dois scripts reservados para testes e reconfiguração dos mesmos durante os testes de bancada. Os scripts são: “config.py”, cuja função é a mesma da página de configuração, porém pode ser acessada diretamente pelo terminal e o script “erase.py”, que apaga os arquivos de configuração, os arquivos CSV e o valor de volume total e faz o sistema reiniciar e inicializar como se fosse o primeiro boot do sistema.

3.7. Testes de Bancada do Sistema

Após o sistema ser montado com o NodeMCU em protoboard e os scripts carregados no ESP-12E, testes preliminares foram feitos a fim de validar o circuito elétrico do sistema. O circuito montado em protoboard e com os scripts pré

carregados foi alimentado com uma fonte de bancada ajustada para 5V e um PC ao lado da bancada atuou com broker MQTT para recebimento dos dados.

O teste consistiu em deixar o sistema ligado enviando dados para o servidor MQTT ao mesmo tempo que mostrava os dados no display LCD sem qualquer interferência, os botões físicos de utilização do usuário foram testados posteriormente.

O teste seguinte consistiu em estressar o sistema com interrupções constantes geradas pelo sensor de vazão. Para isto forçou-se a turbina do sensor de vazão a rodar utilizando uma fonte de ar comprimido, elevando a frequência de saída a valores acima do indicado pelo fabricante do sensor primeiramente com o sistema conectado ao Broker e em seguida simulando uma falha de conexão desligando o servidor presente no PC de apoio.

Feitos os testes com o sistema funcionando corretamente em todos eles, iniciou-se a etapa de projeto de uma placa de circuito que permitisse a ligação de todos os periféricos diretamente ao ESP-12E. Primeiramente foi desenvolvido um adaptador que permitisse ligar o ESP-12E diretamente ao circuito montado na protoboard sem a necessidade do NodeMCU e os testes descritos anteriormente foram repetidos.

Após feito o adaptador verificou-se que o ESP-12E apresentou instabilidades, reiniciando o sistema de forma aleatória. Após testes verificou-se que o ESP-12E apresentava tal problema apenas quando alimentado por fontes chaveadas, mesmo com os filtros de alimentação idênticos aos presentes no NodeMCU, sendo testados 3 tipos diferentes de fontes chaveadas. O problema apenas deixava de se manifestar quando o ESP-12E era alimentado por um regulador linear e por este motivo decidiu-se manter o regulador AMS1117 presente no NodeMCU na nova placa circuito do sistema.

3.8. Desenvolvimento da placa de circuito

A placa de circuito é responsável por integrar todos os periféricos ao ESP-12E ao mesmo tempo que é mais barata que o próprio NodeMCU além de, por ter todos os componentes presentes apenas em uma placa, tornar o sistema mais

robusto. Esta placa deverá conter além do circuito de alimentação, os botões de uso e configuração, um conector para o conversor serial que a partir deste momento seria utilizado externamente, e um conector para um cabo que ligará o conversor do display nesta PCB e deverá conter também os pinos de alimentação.

Esta versão do sistema originalmente também contaria com o RTC integrado diretamente à PCB do sistema e por isso foi fabricada tendo lugar para soldar o DS1307 além do seu cristal oscilador. Posteriormente descobriu-se que o DS1307 sofre muita interferência quando posicionado perto de outros dispositivos emissores de radiofrequência, ocasionando que o RTC conte o tempo de forma mais rápida que o normal e conseqüentemente inviabilizando seu uso.

O circuito de alimentação é um circuito simples e foi projetado seguindo como referência o circuito do NodeMCU com apenas o regulador de tensão e um filtro capacitivo na entrada e na saída do regulador.

O botões são ligados diretamente ao ESP-12E e em paralelo com seus respectivos resistores de pull-down.

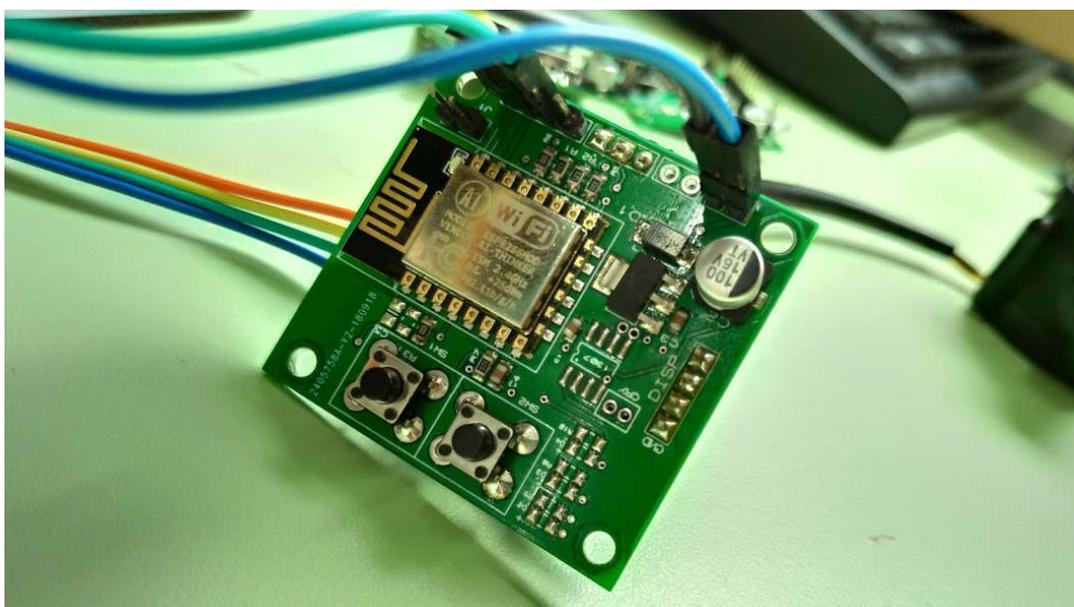
Os terminais de conexão I2C do display, o terminal de programação e o de conexão do sensor de vazão trata-se apenas de pinos ligados diretamente ao ESP-12E.

A placa conta ainda com o ESP-12E com seus resistores de pull-up e pull-down necessários para o seu funcionamento além de LEDs de alimentação e um de propósito geral. A presença de pads para esses LEDs não aumenta o custo do projeto.

O projeto da PCB foi feito em um software CAD específico e o esquemático elétrico pode ser visto na Figura 8.

Após a montagem do primeiro protótipo foi descoberto o problema da interferência da radiofrequência no RTC. Este problema foi contornado soldando o módulo RTC utilizado nos testes com o protoboard diretamente no terminal destinado a conexão com o display LCD, como ambos são I2C e a ordem dos pinos SDA, SCL, VCC e GND eram exatamente iguais essa adaptação foi possível. O display foi então conectado no outro terminal presente no módulo RTC e interligado ao mesmo barramento I2C. A Figura 10 ilustra o segundo protótipo do medidor montado já sem o DS1307 soldado.

Figura 10: Segundo protótipo do medidor sem o RTC soldado



Fonte: Autor

3.9. Desenvolvimento do invólucro de proteção

De forma que o sistema fosse testado em campo, um invólucro de proteção é necessário, uma vez que o sistema pode ficar exposto às intempéries climáticas. Este invólucro além de proteger, tem a função de auxiliar na montagem do dispositivo, já que o display não é diretamente acoplado diretamente a PCB do sistema.

O invólucro foi projetado para que os botões físicos do sistema pudessem ser utilizados por fora e que também houvesse uma abertura por onde é possível ver o display ainda que protegido por um visor de acrílico.

O invólucro foi dividido em duas partes: a parte inferior onde é montada toda a eletrônica e preso o cabo do sensor; e a parte superior com a abertura para os botões e para o display. Todas as partes foram projetadas utilizando software CAD Inventor da Autodesk e os protótipos foram fabricados utilizando impressoras 3D usando ABS como material.

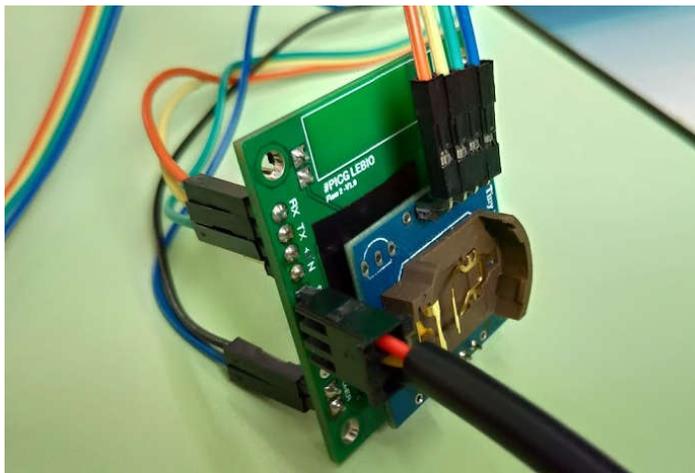
Foram desenvolvidas cinco versões do invólucro devido a facilidade de prototipar novas versões utilizando a impressora 3D e como os botões disponíveis para montagem eram de perfil baixo foi impresso também prolongadores para os botões para que estes pudessem ser acionados de fora do invólucro e sem a necessidade de inserir nenhum objeto dentro do mesmo.

3.10. Montagem do protótipo

Esta etapa descreve a montagem dos protótipos, desde a placa de circuito até o invólucro mecânico dos protótipos com objetivo de serem usados nos testes de campo.

As PCBs recebidas da fábrica chegam sem nenhum componente soldado e foram montadas utilizando ferramentas de soldagem manual, sendo a primeira montada manualmente utilizando uma estação de solda digital circuito por circuito após o teste de cada um seguindo o esquemático elétrico. Neste ponto foi descoberto o problema com o DS1307 e o dispositivo de RF (ESP-12E) como descrito anteriormente, sendo então soldado um módulo RTC idêntico ao usado nos testes em protoboard na PCB, conforme explicado anteriormente e visto na Figura 11.

Figura 11: Módulo RTC soldado a PCB do medidor



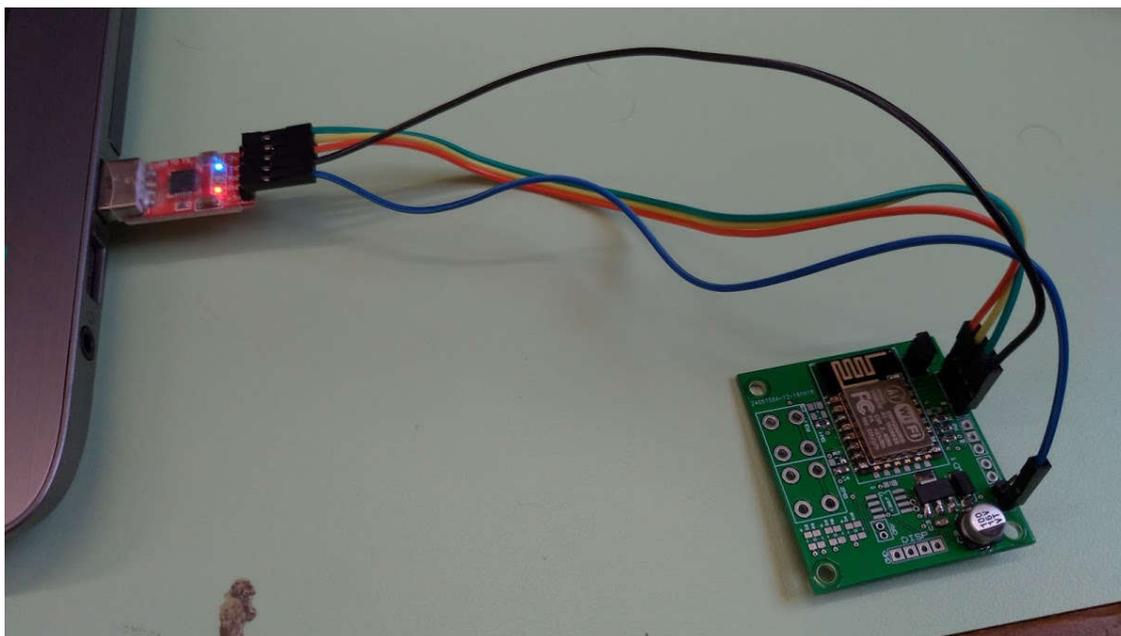
Fonte: Autor

Devido ao problema com o RTC o primeiro protótipo foi descartado sem dar continuidade ao processo de montagem com os trabalhos tendo continuado com o protótipo número dois.

Apesar de existir no mercado a opção de adquirir o display LCD já acoplado ao conversor I2C, no momento da montagem do dispositivo existiam disponíveis os dois componentes apenas separadamente de forma que foi necessário uni-los. O display já acoplado ao conversor foi ligado ao medidor por um cabo de 4 vias.

Com toda a eletrônica montada e o mesmo conectado ao sensor de vazão, os scripts finais foram carregados conectando um conversor USB serial no header disponível para esta finalidade e após isto um teste seco foi realizado com o medidor alimentado por uma fonte de bancada ajustada para 5V. O sistema foi deixado enviando zeros para um broke MQTT local durante 15 horas sem que o sistema parasse de funcionar. A Figura 12 exemplifica um protótipo durante a fase de montagem conectado a interface de programação.

Figura 12: Gravação do firmware (MicroPython) no medidor



Fonte: Autor

Testada a eletrônica em bancada, esta foi montada no invólucro de proteção. Primeiramente a PCB principal é parafusada aos quatro apoios mais baixos da parte inferior do invólucro usando parafusos auto atarrachantes de 2.5mm, o sensor de vazão fica posicionado do lado de fora do invólucro e seu cabo de três vias entra por uma abertura lateral e é conectado ao *header* específico do dispositivo.

O cabo do display é conectado ao *header* oposto ao que foi soldado no RTC mas que dividem o mesmo barramento I2C, o display é então parafusado nos dois apoios mais altos posicionando-o parcialmente sobre a PCB do medidor e com o cabo passando por baixo do display.

A princípio seria usado um suporte que apoiaria o outro lado do display e parafusado junto a PCB principal, porém reforçando os apoios que prendem o display diretamente a parte inferior do invólucro estes apoios foram descartados.

Um visor de acrílico foi colado dentro da parte superior do invólucro para proteger o display LCD. Em seguida os prolongadores dos botões foram encaixados sobre os botões e a parte superior foi encaixada sobre a parte inferior tomando o cuidado para deixar os prolongadores expostos.

Os protótipos montados foram fechados com cola mesmo o ideal sendo a utilização de parafusos. O primeiro protótipo fechado pode ser visto na Figura 13.

Figura 13: Primeiro protótipo fechado do medidor



Fonte: Autor

3.11. Testes de Campo do sistema

O protótipo construído foi instalado para testes próximo a uma máquina de lavar por ser um local conveniente por contar com uma tomada próximo ao ponto de medição. O medidor foi deixado enviando os dados de medição para um Broker MQTT instalado em um Raspberry Pi na mesma rede local.

A Figura 14 mostra o protótipo do medidor com um sensor de vazão de $\frac{3}{4}$ " instalado próximo a máquina de lavar.

Figura 14: Sistema instalado junto a uma máquina de lavar



Fonte: Autor

O sensor funcionou de forma esperada mesmo com as simulações de queda de energia e de conexão com o servidor.

A Tabela 1 exemplifica os dados enviados pelo medidor durante os testes e recebido pelo Broke MQTT instalado no Raspberry Pi.

Tabela 1: Exemplo de dados coletados pelo medidor

	data	hora	valor	volume
716	2018-12-01	18:13:29	0.155184	0.006190
717	2018-12-01	18:13:34	0.180782	0.006441
718	2018-12-01	18:13:39	0.179182	0.006690
719	2018-12-01	18:13:44	0.180782	0.006941
720	2018-12-01	18:13:49	0.180782	0.007192
721	2018-12-01	18:13:54	0.179182	0.007441
722	2018-12-01	18:13:59	0.179182	0.007690
723	2018-12-01	18:14:04	0.179182	0.007938

Fonte: Autor

4. Resultados e Considerações Finais

Este trabalho tinha por objetivo desenvolver um medidor inteligente de consumo de água a um custo de produção do protótipo de R\$ 100,00. O medidor deveria ser baseado no medidor inteligente desenvolvido anteriormente no Polo de inovação Campos dos Goytacazes que tinha um custo de aproximadamente R\$ 500,00.

O protótipo do dispositivo foi construído com todo o hardware desenvolvido e integrado além de se comunicar com o servidor MQTT. Quando comparado ao medidor original as únicas características que foram removidas ou alteradas foram: 1) O período em que o sensor consegue manter os dados internamente enquanto offline foi reduzido devido a quantidade de memória disponível no ESP-12E; 2) Consequentemente o sistema perdeu a capacidade de armazenar localmente os dados de medição de forma permanente; 3) O sistema também perdeu a interface que mostra os dados em tempo real, restando apenas a interface de configuração.

Quanto os custos de produção do protótipo do novo dispositivo, este é um valor que pode variar de acordo a escala de produção. No pior dos casos, com todos os componentes sendo comprados no varejo, e relação dos valores é explicitada abaixo e os custos podem variar de acordo com a cotação do dólar, visto que são todos componentes importados:

- Componentes passivos (resistores, capacitores, etc...): ~R\$ 5,00
 - Módulo ESP-12E: ~R\$ 20,00
 - Placa de circuito impresso: ~R\$ 0,75
 - Módulo RTC DS1307 c/ AT24C32: ~R\$ 11,00
 - Display LCD 16x2 c/ módulo I2C LCD: ~R\$ 30,00
 - Bateria CR2032: ~R\$ 2,00
 - Sensor vazão ½": ~R\$ 30,00
-
- Custo Total com eletrônica: ~R\$ 98,75

Esse detalhamento de custo não leva em consideração os custos com invólucro que por serem feitos em 3D possuem valor mais elevado comparados aos

produzidos em série e é possível encontrar um desses invólucros em catálogos de empresas especializadas na fabricação de peças em plástico moldado que seja compatível com o projeto da eletrônica do medidor desenvolvido neste projeto.

Por último, apesar dos gastos com invólucro estourarem o custo meta de R\$ 100,00 proposto para este dispositivo, é importante ressaltar que os valores mostrados acima são o pior caso, com todos os componentes comprados em lojas varejistas no Brasil e em uma situação de produção em escala este valor cairia consideravelmente.

Apenas dois protótipos foram finalizados, sendo um destinado a testes de bancada e um destinado a testes de campo, tendo estes atendido aos requisitos de operação. Os demais protótipos aguardam definição do projeto I2S para que sejam construídos e integrados.

Como trabalhos futuros recomenda-se, além da integração do dispositivo ao I2S, pequenas alterações na placa de circuito, como:

- Remoção do DS1307 integrado à placa e demais componentes agregados agora que os mesmos são utilizados em um módulo externo;
- Adição de um conector micro-USB para alimentação do sistema, uma vez que isto facilitaria o uso e a reposição da fonte de alimentação por fontes padrão de mercado.
- Mudar o jumper de seleção *Flash/Run mode* para um local da placa em que este não fique abaixo do display LCD, facilitando a manutenção do sistema;

Recomenda-se também como trabalhos futuros, estudos para viabilidade do uso do sistema em locais afastados de qualquer fonte de alimentação por meio do uso de baterias ou painéis fotovoltaicos.

Referências Bibliográficas

©WINSON. **WSH130**. [S.l.: s.n.]. Disponível em: <<http://www.winson.com.tw/Data%20Sheet/WSH130.pdf>>. Acesso em: 9 jan 2019. , 7 Abr 2017

AI-THINKER TECHNOLOGY CO. **ESP-12E 802.11 b/g/n Wi-Fi Module**. [S.l.: s.n.]. Disponível em: <http://wiki.ai-thinker.com/_media/esp8266/docs/a013ps01a0_esp-12e_product_specification_v1.0.pdf>. Acesso em: 12 jan 2018. , 2017

ANA - AGÊNCIA NACIONAL DE ÁGUAS. **Conjuntura dos Recursos Hídricos no Brasil 2017**. [S.l.: s.n.]. Disponível em: <http://conjuntura.ana.gov.br/static/media/conjuntura_completo.caf2236b.pdf>. Acesso em: 23 mar 2018. , 2017

ASHTON, Kevin. **That “Internet of Things” Thing**. 22 Jun 2009. Disponível em: <<https://www.rfidjournal.com/articles/view?4986>>. Acesso em: 4 jan 2019.

CLARK K A, SITNIKOV T, GIFFORD P. **Water sensing assembly for measuring properties of water in a water distribution system and managing measurement data in water utility companies**. [S.l.]: Derwent Innovations Index, [S.d.] Patente dos Estados Unidos nº US2017059543-A1.

EOZAWA. **Matriz energética**. Notícia. Disponível em: <<http://www.brasil.gov.br/meio-ambiente/2010/11/matriz-energetica>>. Acesso em: 23 mar 2018.

ESPRESSIF SYSTEMS. **ESP8266EX Datasheet**. [S.l.: s.n.]. Disponível em: <https://www.espressif.com/sites/default/files/documentation/0a-esp8266ex_datasheet_en.pdf>. Acesso em: 7 jan 2019. , 2018

GAMA, Hiago Santos e RANGEL, Lucas Cordeiro. **Desenvolvimento de medidor eletrônico de energia elétrica**. 2017. 108 f. Trabalho de conclusão de Curso – Instituto Federal Fluminense, Campos dos Goytacazes, 2017. Disponível em: <<http://bd.centro.iff.edu.br/jspui/handle/123456789/1671>>. Acesso em: 23 mar 2018.

GEORGE, Damien. **MicroPython - Python for microcontrollers**. Disponível em: <<http://micropython.org/>>. Acesso em: 9 jan 2019.

GONÇALVES JUNIOR, Marcos José Rangel. **Desenvolvimento de um medidor eletrônico de vazão utilizando um single board computer embarcado**. 2016. 69 f. Trabalho de conclusão de Curso – Instituto Federal Fluminense, Campos dos Goytacazes, 2016. Disponível em: <<http://bd.centro.iff.edu.br/jspui/handle/123456789/1410>>. Acesso em: 23 mar 2018.

IMES K R, HOLLISTER J. **Method of altering operating condition of device e.g. smart appliance located at e.g. residential site, involves detecting availability of control action data at remote server in conjunction with communication of device data** [S.l.]: Derwent Innovations Index, [S.d.] Patente dos Estados Unidos nº US2011307101-A1.

MAXIM INTEGRATED PRODUCTS. **DS130764 x 8, Serial, I2C Real-Time Clock.** . [S.l.: s.n.]. Disponível em: <<https://datasheets.maximintegrated.com/en/ds/DS1307.pdf>>. Acesso em: 9 jan 2019. , 2015

MQTT FAQ. Disponível em: <<https://docs.micropython.org/en/latest/esp8266/quickref.html>>. Acesso em: 04 Abr 2019.

Quick reference for the ESP8266 — MicroPython 1.9.4 documentation. Disponível em: <<http://mqtt.org/faq>>. Acesso em: 18 jan 2019.

RIBEIRO, V. B. e colab. Integration and intelligence for sustainability — I2S An overview of the I2S: Intelligent building's concept from the Campos dos Goytacazes innovation hub campus. In: 2017 IEEE URUCON, Out 2017, [S.l.: s.n.], Out 2017. p. 1–4.

RIBEIRO, Vitor Bastos. **Sistema supervisorio para a arquitetura I2S.** 2018. 2018. Disponível em: <<http://twix.reitoria.iff.edu.br:8080/jspui/handle/123456789/2106>>. Acesso em: 4 jan 2019.