

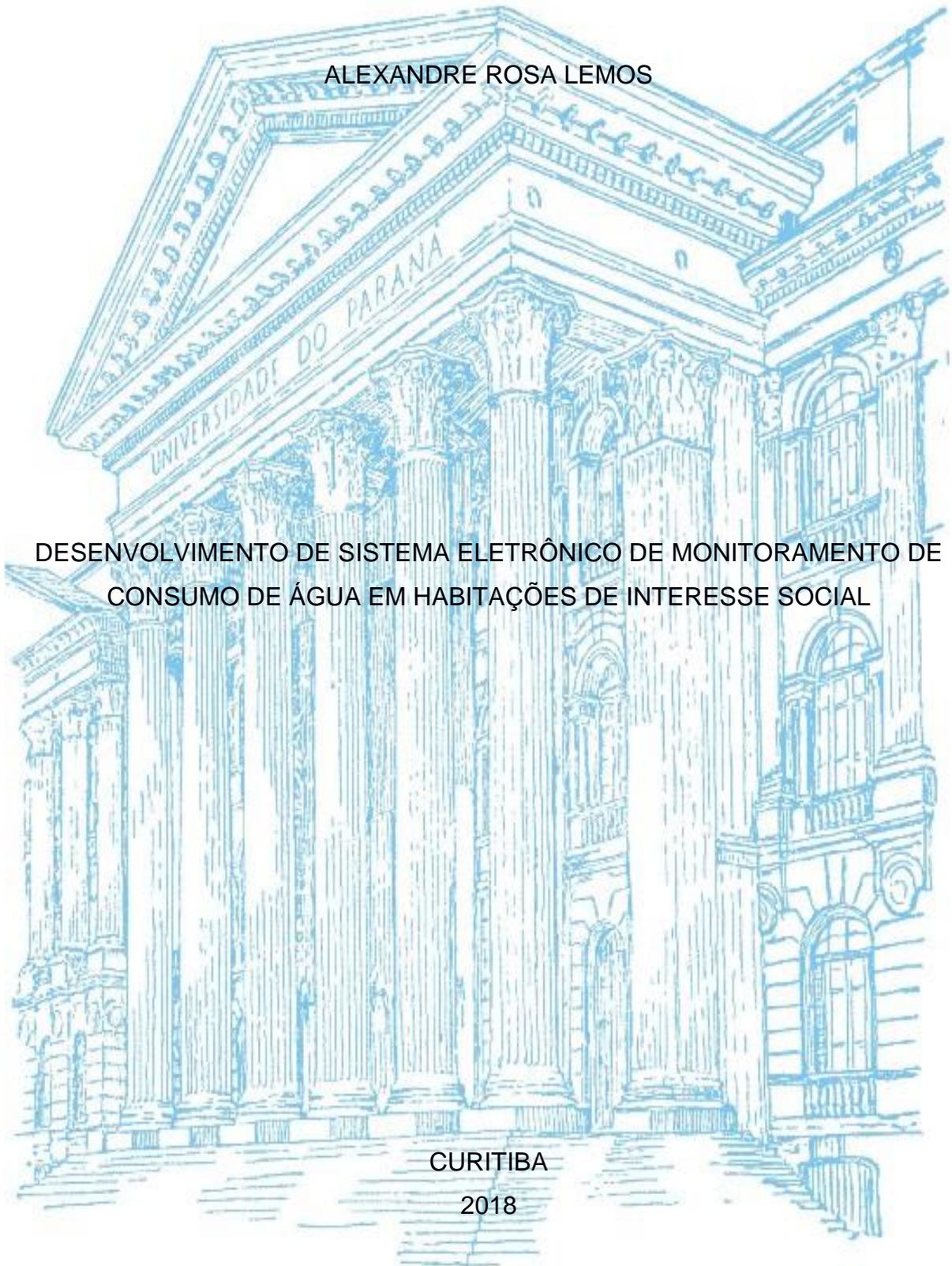
UNIVERSIDADE FEDERAL DO PARANÁ

ALEXANDRE ROSA LEMOS

DESENVOLVIMENTO DE SISTEMA ELETRÔNICO DE MONITORAMENTO DE
CONSUMO DE ÁGUA EM HABITAÇÕES DE INTERESSE SOCIAL

CURITIBA

2018



ALEXANDRE ROSA LEMOS

DESENVOLVIMENTO DE SISTEMA ELETRÔNICO DE MONITORAMENTO DE
CONSUMO DE ÁGUA EM HABITAÇÕES DE INTERESSE SOCIAL

TCC apresentado ao curso de Graduação em Engenharia Elétrica, Setor de Tecnologia, Universidade Federal do Paraná, como requisito parcial à obtenção do título de Bacharel em Engenharia Elétrica.

Orientador: Prof. Dr. James A. Baraniuk

CURITIBA

2018

TERMO DE APROVAÇÃO

ALEXANDRE ROSA LEMOS

DESENVOLVIMENTO DE SISTEMA ELETRÔNICO DE MONITORAMENTO DE CONSUMO DE ÁGUA EM HABITAÇÕES DE INTERESSE SOCIAL

TCC apresentado ao curso de Graduação em Engenharia Elétrica, Setor de Tecnologia, Universidade Federal do Paraná, como requisito parcial à obtenção do título de Bacharel em Engenharia Elétrica.

Prof. Dr. James Alexandre. Baraniuk.
Orientador - Departamento de Engenharia Elétrica - UFPR.

Prof. Dr. André Bellin Mariano.
Departamento de Engenharia Elétrica - UFPR

Prof. Dr. José Carlos da cunha.
Departamento de Engenharia Elétrica - UFPR

Curitiba, 26 de novembro de 2018.

RESUMO

Este trabalho trata do desenvolvimento de um sistema de monitoramento de consumo de água visando aplicação em habitações de interesse social. O consumo de água é registrado por um dispositivo chamado hidrômetro que é instalado na entrada de água das residências e pode ser acompanhado pelo consumidor diretamente pela leitura visual do seu valor no mostrador do hidrômetro. Com o advento do sistema proposto, o consumidor passa a receber essa informação diretamente em uma Unidade Central instalada em sua residência, com dados atualizados em tempo real, facilitando assim o controle por parte deste. A intenção do projeto é auxiliar o consumidor para que este não perca o benefício de tarifa social recebido e pode também contribuir para uma utilização mais racional dos recursos naturais.

Palavras-chave: monitoramento. hidrômetro. Consumo de água.

Abstract

This work deals with the development of a water consumption monitoring system for application in social housing. The water consumption is recorded by a device called a hydrometer that is installed in the entrance of water of the residences and can be accompanied by the consumer directly by the visual reading of its value in the hydrometer display. With the advent of the proposed system, consumers will receive this information directly from a Central Unit installed in their home, with updated data in real time, thus facilitating their control. The intention of the project is to assist the consumer so that he does not lose the benefit of social tariff received and can also contribute to a more rational use of natural resources.

Keywords: monitoring. hydrometer. water consumption.

LISTA DE FIGURAS

FIGURA 1 - TARIFA SOCIAL PARA CONSUMO MENSAL DE 10M ³	5
FIGURA 2 – SIMULADOR DE FATURA DE TARIFA NORMAL	6
FIGURA 3 - REPRESENTAÇÃO GRÁFICA DO SISTEMA	8
FIGURA 4 - DIAGRAMA DE BLOCOS DO SISTEMA.....	9
FIGURA 5 – HIDROMETRO INSTALADO NA ENTRADA DE ÁGUA DO CONSUMIDOR.....	10
FIGURA 6 – MICROCONTOLADOR ESP8266.....	11
FIGURA 7 – AMBIENTE DE DESENVOLVIMENTO ARDUINO.....	12
FIGURA 8 – PLATAFORMA DE HOSPEDAGEM E CONTROLE DE VERSÃO GITHUB.....	12
FIGURA 9 – FUNCIONAMENTO DO SENSOR DE VAZÃO YF-S201	14
FIGURA 10 – SENSOR DE VAZÃO DE ÁGUA YF-S201.....	15
FIGURA 11 – ADAPTADOR DE ROSCA ¾ PARA ½ POLEGADA	16
FIGURA 12 - MODULO ESP8266 MODELO ESP01	17
FIGURA 13 -MÓDULO ESP WEMOS	17
FIGURA 14 - BATERIA RECARREGÁVEL UTILIZADA NA UNIDADE REMOTA.....	18
FIGURA 15 - PLACA FTDI DESENVOLVIDA PELO ENGENHEIRO VILSON R. MOGNON	19
FIGURA 16 - DIAGRAMA ESQUEMATICO DA LACA ADAPTADORA ESP01 PARA FTDI	19
FIGURA 17 - LAYOUT DA PCI DO ADAPTADOR ESP01 PARA FTDI	20
FIGURA 18 - PLACA ADAPTADORA DE ESP01 PARA FTDI	20
FIGURA 19 - DIAGRAMA ESQUEMÁTICO DA UNIDADE REMOTA.....	20
FIGURA 20 - LAYOUT DA PCI DA UNIDADE REMOTA.....	21
FIGURA 21 - MÓDULO ESP8266 NODEMCU.....	22
FIGURA 22 - MODULO RTC DS3231	23
FIGURA 23 - MÓDULO E CARTÃO MICRO SD.....	24
FIGURA 24 - DISPAY DE LCD PARALELO 16 LINHAS X 2 COLUNAS	24
FIGURA 25 - ADAPTADOR DE DISPLAY PARALELO PARA I2C.....	25
FIGURA 26 - BATERIA RECARREGÁVEL UTILIZADA NA UNIDADE CENTRAL.....	26
FIGURA 27 - MÓDULO CARREGAR DE BATERIA TP4056.....	26
FIGURA 28 – PLACA DE TESTES COM MODULOS NODEMCU, RTC E SD CARD.....	27
FIGURA 29 - DIAGRAMA ESQUEMÁTICO DA UNIDADE CENTRAL	28
FIGURA 30 - LAYOUT DA PCI DA UNIDADE CENTRAL	29

FIGURA 31 – GERAÇÃO DE ARQUIVOS DE FABRICAÇÃO GERBER E NC DRILL.....	29
FIGURA 32 – ARQUIVO GERBER PARA FRESAGEM DA PCI DA UNIDADE CENTRAL	30
FIGURA 33 – ARQUIVO DE FURAÇÃO DA PCI DA UNIDADE CENTRAL.....	30
FIGURA 34 – UNIDADE CENTRAL PROCESSADA NO SOFTWARE CAM CIRCUITPRO	31
FIGURA 35 - FABRICAÇÃO DA PCI DA UNIDADE CENTRAL PELA LPKF PROTOMAT S63	31
FIGURA 36 – UNIDADE REMOTA ACOPLADA AO SENSOR DE FLUXO.....	33
FIGURA 37 – UNIDADE CENTRAL COM DISPLAY DE LCD 16X2	33
FIGURA 38 - CONFIGURAÇÃO DO AMBIENTE PARA OPERAR COM MÓDULOS ESP.....	34
FIGURA 39 - SELEÇÃO DA PLACA ESP01 NO AMBIENTE DE GRAVAÇÃO	35
FIGURA 40 - TESTE DE COMUNICAÇÃO ENTRE UNIDADES UNIDADE REMOTA E UNIDADE CENTRAL	36
FIGURA 41- ARQUIVO .CSV GRAVADO NO CARTÃO SD COM INFORMAÇÕES DE CONSUMO, DATA E HORA.....	36
FIGURA 42 - TESTE DE COMUNICAÇÃO COM O SERVIDOR GRATUITO THINGSPEAK	37
FIGURA 43 – TESTE DE LEITURA DOS PULSOS DE SAÍDA DO SENSOR YF-S201	38
FIGURA 44 – INFORMAÇÕES DE CONSUMO DOS MESES ANTERIOR E ATUAL	39

LISTA DE TABELAS

TABELA 1 – COMPARAÇÃO ENTRE TARIFA SOCIAL E CONVENCIONAL.....	7
---	---

LISTA DE SIGLAS

UFPR	- UNIVERSIDADE FEDERAL DO PARANÁ
ICON	- INTELIGENCIA NO CONSUMO EM HABITAÇÃO DE INTERESSE SOCIAL
DELT	- DEPARTAMENTO DE ENGENHARIA ELÉTRICA
ANA	- AGÊNCIA NACIONAL DAS ÁGUAS
SANEPAR	- COMPANHIA DE SANEAMENTO DO PARANÁ
PCI	- PLACA DE CIRCUITO IMPRESSO
WIFI – WIRELESS FIDELITY	
IDE	- INTEGRATED DEVELOPMENT ENVIRONMENT
IOT	- INTERNET OF THINGS
OTA	- OVER THE AIR
CAD	- COMPUTER-AIDED DESIGN
CAM	- COMPUTER-AIDED MANUFACTURING
RTC	- REAL TIME CLOCK
CSV	- COMMA-SEPARATED VALUE
LCD	- LIQUID CRYSTAL DISPLAY
FTDI	- FUTURE TECHNOLOGY DEVICES INTERNATIONAL
SD	- SECURE DIGITAL
USB	- UNIVERSAL SERIAL BUS
UDP	- USER DATAGRAM PROTOCOL
SPI	- SERIAL PERIPHERAL INTERFACE
I2C	- INTER-INTEGRATED CIRCUIT

SUMÁRIO

1. INTRODUÇÃO	3
1.1 PROBLEMAS E MOTIVAÇÃO	3
2 OBJETIVO GERAL	8
2.1.1 OBJETIVOS ESPECÍFICOS	8
3 METODOLOGIA	9
3.1 MEDIDORES DE CONSUMO DE ÁGUA.....	9
3.2 DISPOSITIVOS DE COMUNICAÇÃO SEM FIO	10
3.3 AMBIENTE DE PROGRAMAÇÃO E DESENVOLVIMENTO DE FIRMWARE.....	11
3.4 SISTEMAS DE ARMAZENAMENTO DE DADOS.....	13
3.5 INTERFACE COM USUÁRIO.....	13
3.6 SENSOR DE FLUXO DE ÁGUA.....	13
4 PROTÓTIPO FUNCIONAL	15
4.1 UNIDADE REMOTA	15
4.1.1 SENSOR DE VAZÃO DE ÁGUA.....	15
4.1.2 MICROCONTROLADOR	16
4.1.3 ALIMENTAÇÃO	18
4.1.4 INTERFACE DE GRAVAÇÃO DA UNIDADE REMOTA	18
4.2 UNIDADE CENTRAL.....	21
4.2.1 ESP8266 NODEMCU.....	22
4.2.2 REFERÊNCIA DE DATA E HORA.....	23
4.2.3 ARMAZENAMENTO EXTERNO DE DADOS	23
4.2.4 INTERFACE COM O CONSUMIDOR.....	24
4.2.5 ALIMENTAÇÃO	25
4.2.6 IMPLEMENTAÇÃO DA UNIDADE CENTRAL	27
4.3 SISTEMA DE MONITORAMENTO DO CONSUMO DE ÁGUA	33
5 DESENVOLVIMENTO DO FIRMWARE	34
5.1 COMUNICAÇÃO ENTRE DISPOSITIVOS DE AQUISIÇÃO E DISPONIBILIZAÇÃO DOS DADOS	

5.2	INSERÇÃO DA INFORMAÇÃO REAL DE CONSUMO NO SISTEMA.....	37
5.3	VIZUALIZAÇÃO DO CONSUMO.....	38
6	CONSIDERAÇÕES FINAIS E TRABALHOS FUTUROS.....	40
7	APENCICE 1 – FIRMWARE DA UNIDADE REMOTA	42
8	APENCICE 2 – FIRMWARE DA UNIDADE CENTRAL.....	44

1. INTRODUÇÃO

Um sistema de monitoramento de consumo de água está sendo desenvolvido pela pós-graduação em design da UFPR e conduzido pelo prof. Dr. Aguinaldo dos Santos, denominado como PROJETO ICON (Inteligência no Consumo na Habitação de Interesse Social). Coube ao Departamento de Engenharia Elétrica, na pessoa do prof. Dr. James Alexandre Baraniuk, colaborar com o desenvolvimento de uma solução eletrônica para o sistema. Este trabalho de conclusão de curso apresenta a implementação de um sistema de medição, transmissão e leitura de consumo de água, visando sua instalação em Habitações de Interesse Social.

Habitação de Interesse Social é um tipo de habitação destinada à população de baixa renda e é regulamentada pela lei federal 11.124/2005.

Em 2017, 186 mil famílias se beneficiaram da tarifa social, cujo valor até 5m³ era de R\$ 13,20 (R\$ 8,80 da tarifa de água e R\$ 4,40 da tarifa de esgotamento sanitário). Esse montante representou 5,3% do total de economias residenciais da companhia, equivalendo a um subsídio de quase R\$ 97 milhões (SANEPAR).

1.1 PROBLEMAS E MOTIVAÇÃO

Existem programas de concessão de benefícios oferecidos à população de baixa renda, como as tarifas sociais de água e energia elétrica. A Agência Nacional de Águas (ANA) regulamenta a aplicação da tarifa social para moradores de habitações multifamiliares regulares e irregulares de baixa renda, onde não for tecnicamente possível a instalação de medidores para cada uma das famílias.

A implementação de um sistema de monitoramento do consumo de água permitirá ao usuário realizar o acompanhamento em tempo real do consumo, podendo este tomar medidas para evitar ultrapassar o consumo máximo permitido de acordo com a regulamentação vigente em sua região. No estado do Paraná, para que um consumidor possa usufruir do benefício da tarifa social, é necessário realizar um cadastro em uma Unidade Central de Relacionamento da SANEPAR apresentando a documentação exigida no site oficial da companhia. De acordo com o site da companhia, este benefício é concedido por um período de 24 meses, podendo ser renovado por igual período mediante comprovação documental e atendimento dos

critérios. Ainda de acordo com o site da SANEPAR, os critérios utilizados para ter direito ao benefício da tarifa social são os seguintes:

- ✓ Imóvel: somente devem ser cadastrados os imóveis com área construída de até 70m² (setenta metros quadrados), para fins residenciais.
- ✓ Consumo: o consumo mensal de água deve ser de até 10m³/mês. Para famílias com mais de 4 (quatro) pessoas e consumo superior a 10m³/mês, deverá ser considerado o consumo de até 2,5m³/mês por residente no imóvel.
- ✓ Renda: a renda da família residente no imóvel será de até ½ salário mínimo por pessoa ou de até 2 salários mínimos (federal) para imóveis com até 4 ocupantes, vigente na data de solicitação do benefício.

Tomando como base o caso da Companhia de Saneamento do Paraná podemos encontrar em sua página oficial, um simulador de tarifas e custos. Esse simulador possui as seguintes opções configuráveis:

- ✓ TIPO DE LIGAÇÃO: ÁGUA OU ÁGUA E ESGOTO
- ✓ TARIFA SOCIAL: NÃO OU SIM
- ✓ MICRO E PEQUENO COMÉRCIO: NÃO OU SIM

Na FIGURA 1, temos uma imagem gerada a partir do simulador de tarifas disponível no site da SANEPAR. Nesta simulação vemos os valores praticados para um consumidor, beneficiário da tarifa social, para um consumo mensal de 10m³.

FIGURA 1 - TARIFA SOCIAL PARA CONSUMO MENSAL DE 10M³

SANEPAR
COMPANHIA DE SANEAMENTO DO PARANÁ

TIPO DE LIGAÇÃO
 Água + Esgoto
 Água

TARIFA SOCIAL
 Não Sim

MICRO E PEQUENO COMÉRCIO
 Não Sim

SIMULADOR DE TARIFAS

LOCAL
CURITIBA

RES: 1 COM: 000 IND: 000 UTP: 000 POP: 000

FAIXAS DE CONSUMO	VOLUME	VALOR m ³ /RS		TOTAIS	
		ÁGUA	ESGOTO	ÁGUA	ESGOTO
Mínimo	5	9,25	4,62	9,25	4,62
de 6 a 10	5	0,28	0,70	1,40	0,70

CONSUMO/m³
10

ÁGUA	ESGOTO	TOTAL
10,65	5,32	15,97

SANEPAR **SIMULAR**

FONTE: SITE DA SANEPAR (2018)

A FIGURA 2 mostra a simulação para um consumidor que não possui o benefício da tarifa social e que teve o mesmo consumo de água em igual período.

FIGURA 2 – SIMULADOR DE FATURA DE TARIFA NORMAL

SANEPAR
COMPANHIA DE SANEAMENTO DO PARANÁ

TIPO DE LIGAÇÃO
 Água + Esgoto
 Água

TARIFA SOCIAL
 Não Sim

MICRO E PEQUENO COMÉRCIO
 Não Sim

SIMULADOR DE TARIFAS

LOCAL
CURITIBA

RES: 1 COM: 000 IND: 000 UTP: 000 POP: 000

FAIXAS DE CONSUMO	VOLUME	VALOR m ³ /RS		TOTAIS	
		ÁGUA	ESGOTO	ÁGUA	ESGOTO
Mínimo	5	34,58		34,58	29,39
de 6 a 10	5	1,07		5,35	4,54

CONSUMO/m³
10

ÁGUA	ESGOTO	TOTAL
39,93	33,93	73,86

SANEPAR **SIMULAR**

FONTE: SITE DA SANEPAR (2018)

O consumidor beneficiário da tarifa social, paga R\$ 15,97 pelo consumo de 10m³ de água em um mês, enquanto que um consumidor que não usufrui do benefício, paga uma tarifa de r\$ 73,86. Essa diferença resulta em uma economia de R\$ 57,89 por mês, totalizando r\$ 694,68 no período de um ano. O projeto resultante pretende

contribuir para a redução do risco de perda do benefício e das despesas de famílias de baixa renda, beneficiários da tarifa social ou não, possibilitando também uma utilização mais racional dos recursos naturais. De acordo com GURUNG (2014), com o advento de medidores de consumo de água inteligentes e de alta resolução permite usar essa base de dados gerados para criar curvas de demandas de abastecimento adequadas às regiões analisadas. A partir do aperfeiçoamento do sistema, poderão ser implantados mais sensores nos principais pontos de interesse da residência, como no chuveiro, lavanderia e cozinha criando perfis de uso de cada ponto especificado. Um evento de banho, por exemplo, pode consumir cerca de 40 litros de água segundo um estudo feito por GURAGAI (208). A TABELA 1 apresenta uma comparação entre os valores praticados pelas tarifas normal e social, destacando o desconto mensal e a economia anual resultantes do benefício concedido para quatro faixas de consumo.

TABELA 1 – COMPARAÇÃO ENTRE TARIFA SOCIAL E CONVENCIONAL

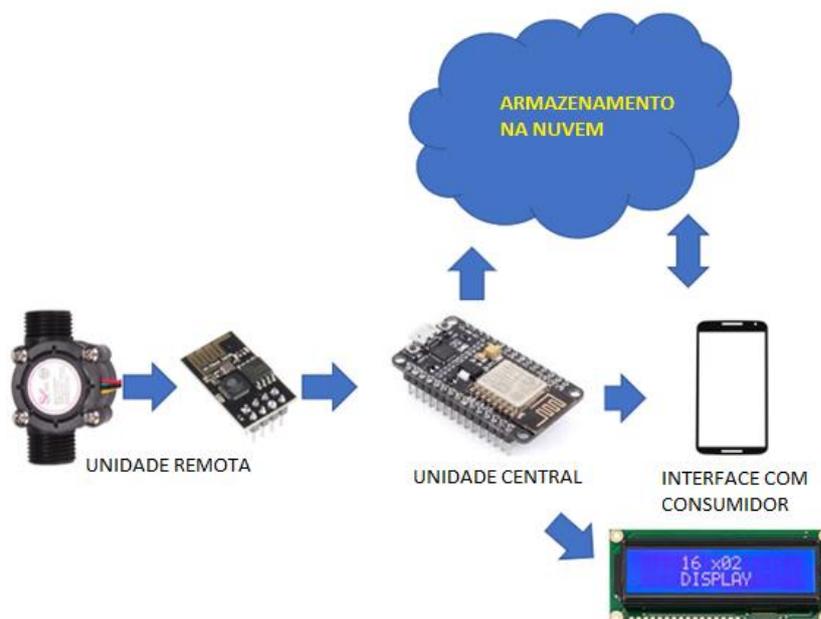
CONSUMO(m ³)	VOLUME(m ³)	TARIFA NORMAL	TARIFA SOCIAL	DESCONTO MENSAL	ECONOMIA ANUAL
Mínimo	5	R\$ 63,97	R\$ 13,87	R\$ 50,10	R\$ 601,20
De 6 a 10	10	R\$ 73,86	R\$ 15,97	R\$ 57,89	R\$ 694,68

FONTE: O AUTOR (DADOS DO SITE DA SANEPAR 2018)

2 OBJETIVO GERAL

Desenvolver um sistema de medição do consumo de água e disponibilizar essa informação em tempo real. Este sistema possui uma Unidade Remota, a qual realiza a aquisição e transmissão dos dados de consumo de água e uma Unidade Central que recebe e disponibiliza a informação de consumo para o usuário. A FIGURA 3 mostra uma representação gráfica do sistema.

FIGURA 3 - REPRESENTAÇÃO GRÁFICA DO SISTEMA



FONTE: O AUTOR (ADAPTADO DE BARANIUK 2018)

2.1.1 OBJETIVOS ESPECÍFICOS

Como requisitos parciais para a obtenção do objetivo geral, as seguintes atividades foram desempenhadas:

- Estudo das tecnologias disponíveis para o desenvolvimento do projeto;
- Construção de placas de circuito impresso que agreguem os componentes necessários a implementação física do sistema;
- Desenvolvimento de um firmware de controle dos dispositivos;
- Disponibilização ao usuário do sistema a informação do consumo de água;
- Realização de testes de validação em laboratório.

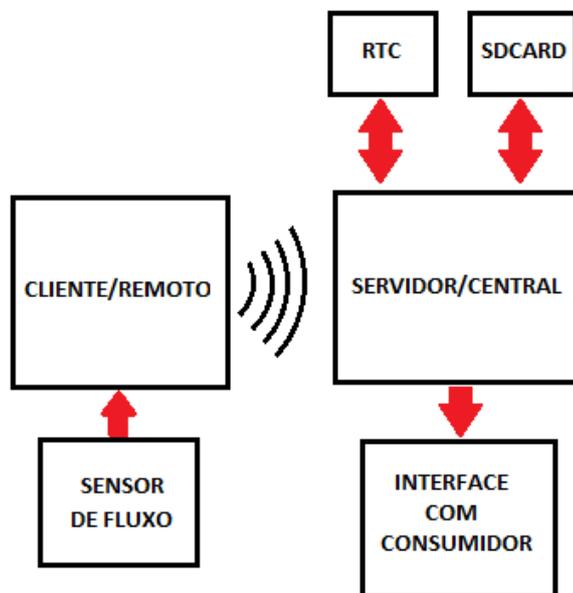
3 METODOLOGIA

A especificação do projeto foi feita a partir do estudo das tecnologias utilizadas e alternativas disponíveis. Na sequência foram desenvolvidos códigos e circuitos para cada uma das etapas de funcionamento para posterior integração. Abaixo são apresentadas de maneira geral as principais etapas de desenvolvimento:

- ✓ Comunicação WiFi entre os dispositivos;
- ✓ Obtenção da data e hora a partir de um modulo RTC externo;
- ✓ Gravação dos dados recebidos em um cartão SD externo;
- ✓ Disponibilização de uma interface visual para leitura dos dados.

Os testes em laboratório validaram o protótipo funcional que é apresentado neste relatório. Uma representação do sistema em forma de diagrama de blocos é mostrada na FIGURA 4.

FIGURA 4 - DIAGRAMA DE BLOCOS DO SISTEMA



FONTE: O AUTOR (2018)

3.1 MEDIDORES DE CONSUMO DE ÁGUA

Atualmente os medidores de consumo de água, chamados de hidrômetro (FIGURA 5), são instalados na entrada de água dos consumidores e registram a

quantidade de água consumida, a qual é lida por um funcionário da companhia de saneamento responsável por fornecer o serviço. Na fatura do consumidor constam os consumos acumulados e consumo do mês atual, sendo essa diferença consumo o taxado. Os consumidores podem eventualmente realizar um controle pessoal, anotando os valores lidos. No entanto isto não é prático, visto que caso um desses valores seja anotado incorretamente, ou seja esquecido de ser registrado isto comprometeria o controle. O sistema desenvolvido apresenta a leitura do consumo atual em tempo real enquanto mantém disponível a informação do consumo do mês anterior em uma interface amigável, como um display digital, tornando viável um acompanhamento e eventual controle do consumo de água.

FIGURA 5 – HIDROMETRO INSTALADO NA ENTRADA DE ÁGUA DO CONSUMIDOR



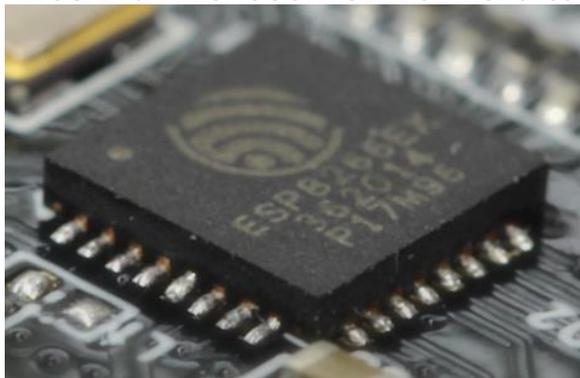
FONTE: SITE DA SANEPAR (2018)

3.2 DISPOSITIVOS DE COMUNICAÇÃO SEM FIO

Com os avanços no desenvolvimento de sensores e dispositivos microcontrolados e da comunicação sem fio, a implementação da IoT tem se tornado mais simples e com menor custo. O padrão WiFi A família de microcontroladores ESP da empresa ESPRESSIF, possuem comunicação *Wifi* integrada, produtos de baixo custo e alto poder de integração em sistemas de *hardware* e *firmware*, visto que a comunidade científica desenvolve bibliotecas *open source* a fim de popularizar tecnologias e compartilhar conhecimento. Dentre os modelos produzidos pela ESPRESSIF o microcontrolador ESP8266 é um dos mais populares, sendo utilizado em módulos como os NodeMCU e ESP01 especificados para este projeto. O ESP8266 é um microcontrolador de 32 bits que funciona na frequência de até

160MHz, sendo que apenas 20% da sua capacidade de processamento é utilizada pela comunicação WiFi. O microcontrolador ESP8266 consome 170mA de corrente quando está transmitindo em sua potência máxima, 50mA na recepção e 15mA no modo de economia de energia. (OLIVEIRA, 2017). A FIGURA 6 mostra o microcontrolador ESP8266.

FIGURA 6 – MICROCONTOLADOR ESP8266

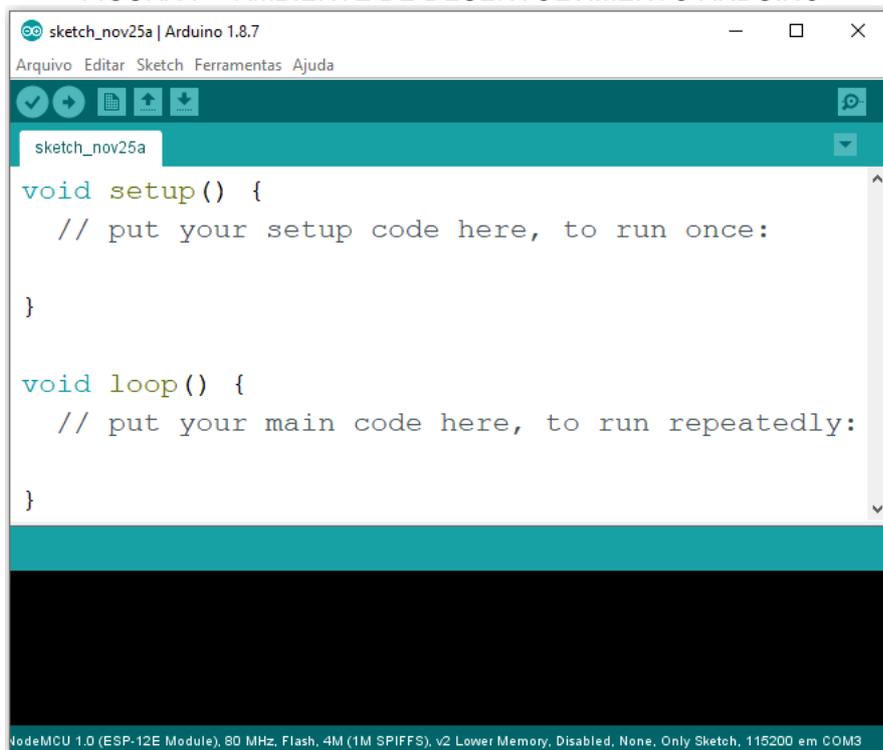


FONTE: SITE DA ESPRESSIF (2018)

3.3 AMBIENTE DE PROGRAMAÇÃO E DESENVOLVIMENTO DE FIRMWARE

Os microcontroladores da ESPRESSIF podem ser programados em diversos ambientes de desenvolvimento, como as plataformas Eclipse e Arduino. Devido à facilidade de implementação, optou-se por utilizar o ambiente de desenvolvimento Arduino na programação dos módulos ESP utilizados no projeto. Por ser uma plataforma livre, esta possui um grande acervo de bibliotecas gratuitas em sites da comunidade desenvolvedora. Dentro dessas bibliotecas existem funções que realizam o tratamento de sinais, viabilizam comunicação *wireless* e disponibilizam a visualização de dados para o usuário. A FIGURA 7 mostra a tela inicial do ambiente de desenvolvimento Arduino. A plataforma de hospedagem com controle de versão GitHub foi a fonte de consulta mais utilizada para o desenvolvimento do projeto, pois disponibiliza vários modelos de implementação de códigos similares aos necessários para o funcionamento do sistema. A FIGURA 8 mostra uma consulta de ocorrências envolvendo o microcontrolador ESP8266 e o ambiente de desenvolvimento Arduino efetuada no dia 25 de novembro de 2018. Pode-se observar que no momento da consulta existiam 2.904 arquivos no acervo, desde bibliotecas a códigos de projetos desenvolvidos pela comunidade.

FIGURA 7 – AMBIENTE DE DESENVOLVIMENTO ARDUINO



FONTE: ARDUINO (2018)

FIGURA 8 – PLATAFORMA DE HOSPEDAGEM E CONTROLE DE VERSÃO GITHUB

The screenshot shows the GitHub repository page for 'esp8266 / Arduino'. The repository name is 'esp8266 / Arduino'. It has 877 Watchers, 8,313 Stars, and 5,716 Forks. The repository is categorized under 'Code', 'Issues' (407), 'Pull requests' (110), 'Projects' (5), 'Wiki', and 'Insights'. The repository is titled 'ESP8266 core for Arduino'. It has 2,904 commits, 26 branches, 16 releases, 297 contributors, and is licensed under LGPL-2.1. The current branch is 'master'. There are buttons for 'New pull request', 'Create new file', 'Upload files', 'Find file', and 'Clone or download'. A list of recent commits is shown, including:

- devyte Fix rtc mem bounds check (#5372) - Latest commit 72ad935 19 hours ago
- bootloaders/eboot cleanup/unify flash sector size define value (#5327) - a day ago
- cores/esp8266 Fix rtc mem bounds check (#5372) - 19 hours ago
- doc document wificlient loop (#5355) - 5 days ago
- libraries Remove obsolete BSSL debug print routines (#5368) - 2 days ago
- package support new board (invent one) (#5286) - 28 days ago
- tests Sketch emulation on host (#5342) - 5 days ago
- tools cleanup/unify flash sector size define value (#5327) - a day ago
- variants support new board (invent one) (#5286) - 28 days ago
- .gitignore Add valgrind and string tests to host_tests (#4939) - 4 months ago
- .gitmodules Add a build directory for libbearssl.a (#4736) - 6 months ago

FONTE: SITE OFICIAL GITHUB.COM (2018)

3.4 SISTEMAS DE ARMAZENAMENTO DE DADOS

De acordo com o datasheet do microcontrolador ESP8266, este não possui memória ROM interna, por isso pode utilizar memórias flash SPI externas de até 16Mb. Um módulo de cartão micro SD se comunica com a Central através da interface SPI, possibilitando atingir uma memória da ordem de gigabytes para a criação de um histórico de consumo a longo prazo, criando um arquivo CSV em um cartão micro SD.

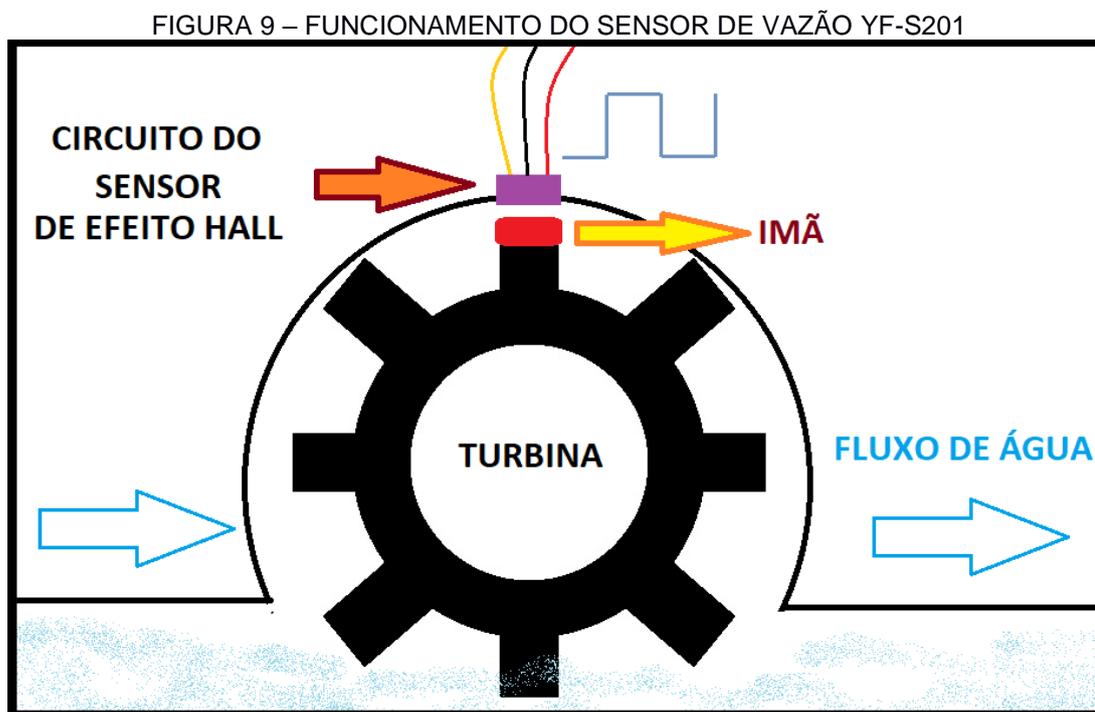
3.5 INTERFACE COM USUÁRIO

Como interface com o usuário, optou-se a utilização de um display LCD, visto que nem todos os consumidores possuem dispositivos eletrônicos como smartphones ou computadores e tampouco acesso à internet para consulta dos dados de consumo.

3.6 SENSOR DE FLUXO DE ÁGUA

O sensor YF-201 é um sensor de efeito Hall utilizado no projeto como sensor de vazão de água. Os sensores de efeito Hall consistem em dispositivos de sensoriamento ativados magneticamente, sendo assim imunes às condições de funcionamento adversas como presença de poeira, variações de temperatura, condições de iluminação e umidade. O sensor de efeito Hall básico é um dispositivo semicondutor. Uma tensão de alimentação faz circular uma corrente de polarização no sensor. Uma tensão é gerada pelo sensor na presença de um campo magnético. A “tensão Hall” é proporcional à intensidade do campo magnético. Se não houver campo magnético a tensão na saída do sensor será nula. Se o sensor de efeito Hall é inserido em uma região onde existe um campo magnético, a tensão de saída será diretamente proporcional à intensidade desse campo. O “efeito Hall” foi descoberto por E. F. Hall em 1879. Um ímã afixado em uma pá de uma turbina, é inserida em um encapsulamento plástico fechado e, no qual é colocado um sensor de efeito Hall. Quando a água flui através da tubulação, faz girar a turbina causando interferência magnética no sensor. A taxa de interferência depende da velocidade do fluxo de água,

então o sensor de efeito Hall produz um sinal de pulso em sua saída. De acordo com o fabricante, cada pulso corresponde a um fluxo de 2,25ml, e a contagem dos pulsos corresponde ao volume de água que flui através do sensor. Um modelo do funcionamento do sensor YF-S201 é mostrado na FIGURA 9.



FONTE: ADAPTADO DE theoryCIRCUIT.com (2018)

4 PROTÓTIPO FUNCIONAL

O protótipo funcional do sistema de monitoramento do consumo de água, é composto pelas Unidades Remota e Central de processamento. Esta seção trata do desenvolvimento de hardware e firmware deste protótipo.

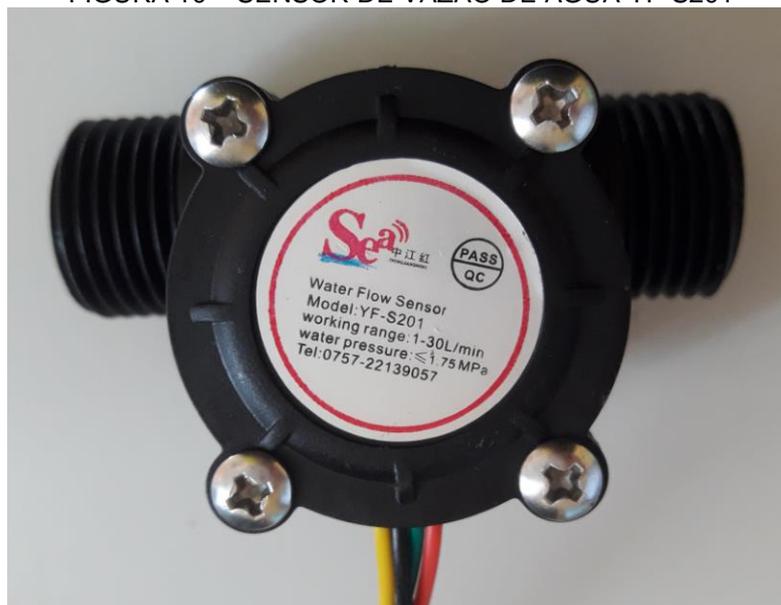
4.1 UNIDADE REMOTA

A Unidade Remota integra um sensor de vazão, um módulo de processamento dos dados, fornecidos por este sensor, e uma bateria recarregável de 3.6 volts para realizar sua alimentação.

4.1.1 SENSOR DE VAZÃO DE ÁGUA

O sensor YF-S201 utilizado, é mostrado na FIGURA 10. A água que flui através deste sensor movimenta uma turbina causando uma interferência no campo magnético, devido ao efeito Hall e, gera um pulso em sua saída. De acordo com a folha de dados do fabricante, cada um destes pulsos corresponde à uma vazão de 2,25ml.

FIGURA 10 – SENSOR DE VAZÃO DE ÁGUA YF-S201



FONTE: O AUTOR (2018)

As características do sensor YF-S201 são descritas abaixo:

- ✓ TIPO: EFEITO HALL;
- ✓ TENSÃO DE OPERAÇÃO: 5-18V;
- ✓ CORRENTE MÁXIMA: 15MA@5V;
- ✓ FAIXA DE FLUXO: 1-30L/MIN;
- ✓ PRESSÃO MÁXIMA: 2,0 MPA;
- ✓ PULSOS POR LITRO: 450;
- ✓ FREQUÊNCIA (HZ) = 7,5*FLUXO(L/MIN);
- ✓ TEMPERATURA DE TRABALHO: -25 A 80°C;
- ✓ EXATIDÃO: 10%.

O sensor yf-s201 possui rosca de $\frac{3}{4}$ de polegada e para testar o sistema foi necessária a inclusão de um adaptador (FIGURA 11) para $\frac{1}{2}$ polegada que é o padrão da torneira onde o sistema foi validado.

- ✓ FIGURA 11 – ADAPTADOR DE ROSCA $\frac{3}{4}$ PARA $\frac{1}{2}$ POLEGADA



✓

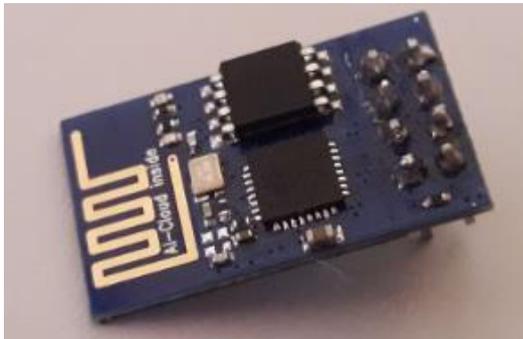
- ✓ FONTE: O AUTOR (2018)

4.1.2 MICROCONTROLADOR

O módulo ESP01, visto na FIGURA 12, foi escolhido por possuir o microcontrolador ESP8266, o qual possui comunicação WiFi integrada e conta com acesso aos pinos de interrupção utilizados como entrada de dados oriundos do sensor de vazão. Antes de se chegar a conclusão da viabilidade técnica para a gravação e funcionalidade do ESP01, a unidade remota foi testada com o modulo ESP WEMOS

que também utiliza o microcontrolador ESP8266 e é mostrado na FIGURA 13.

FIGURA 12 - MÓDULO ESP8266 MODELO ESP01

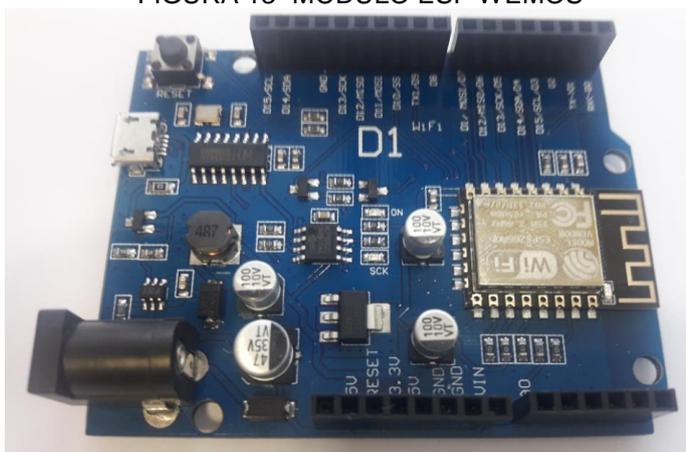


FONTE: O AUTOR (2018)

O módulo ESP01 apresenta as seguintes características:

- ✓ SUPORTE À REDES: 802.11 B/G/N;
- ✓ TENSÃO DE OPERAÇÃO: 3,3V;
- ✓ ALCANCE: 90m;
- ✓ COMUNICAÇÃO: SERIAL (TX/RX);
- ✓ SUPORTA COMUNICAÇÃO TCP E UDP;
- ✓ CONECTORES: GPIO, I2C, SPI, UART, ENTRADA ADC, SAÍDA PWM E SENSOR DE TEMPERATURA INTERNO;
- ✓ DIMENSÕES: 25x14x1mm;
- ✓ PESO: 7G.

FIGURA 13 -MÓDULO ESP WEMOS



FONTE: O AUTOR (2018)

4.1.3 ALIMENTAÇÃO

A alimentação da Unidade Remota é provida por uma bateria de Íons de Lítio com tensão de 3.6V e capacidade de corrente de 190mAh, mostrada na FIGURA 14.

FIGURA 14 - BATERIA RECARREGÁVEL UTILIZADA NA UNIDADE REMOTA



FONTE: O AUTOR (2018)

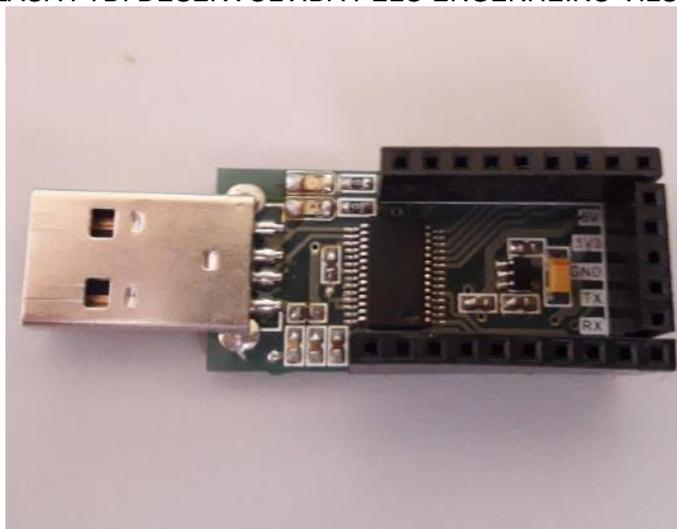
A seguir as características da bateria de acordo com dados do fabricante.

- ✓ TIPO: LI-ÍON;
- ✓ TENSÃO: 3,6V;
- ✓ CORRENTE: 190mAh;
- ✓ DIÂMETRO: 24.5mm;
- ✓ ESPESSURA: 7.7mm;
- ✓ PESO: 7,5g.

4.1.4 INTERFACE DE GRAVAÇÃO DA UNIDADE REMOTA

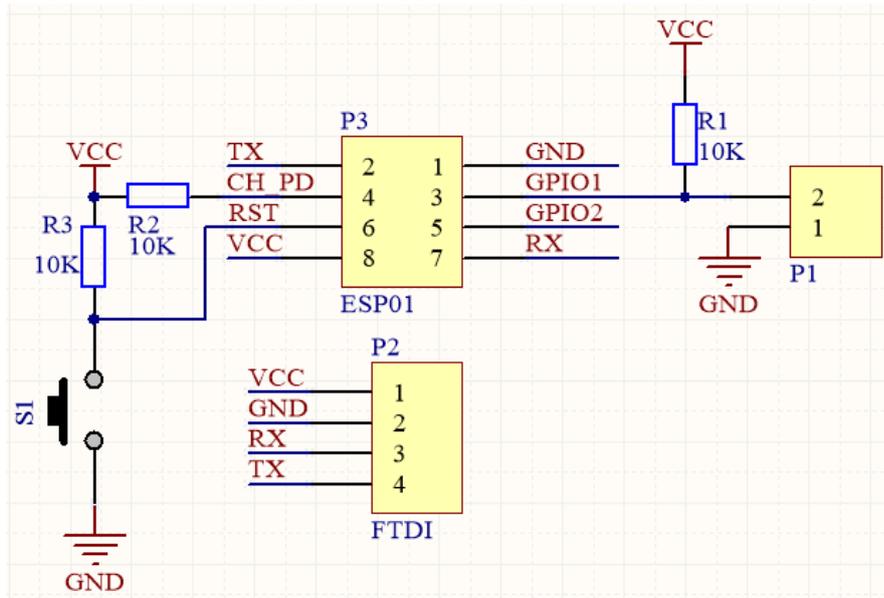
Por não possuir uma interface de gravação própria, o módulo ESP01 necessita de uma interface externa. Existem módulos padrão FTDI, como o desenvolvido pelo engenheiro Vilson Rodrigo Mognon e mostrado na FIGURA 15, que são utilizados para a gravação de microcontroladores. Ainda assim foi necessário o desenvolvimento de um adaptador para realizar a conexão do ESP01 à placa FTDI, sendo conectada à porta USB do microcomputador utilizado no desenvolvimento do firmware. A FIGURA 16 mostra o diagrama esquemático dessa placa adaptadora, e as FIGURAS 17 e 18 mostram o layout da placa de circuito impresso e a placa adaptadora montada respectivamente.

FIGURA 15 - PLACA FTDI DESENVOLVIDA PELO ENGENHEIRO VILSON R. MOGNON



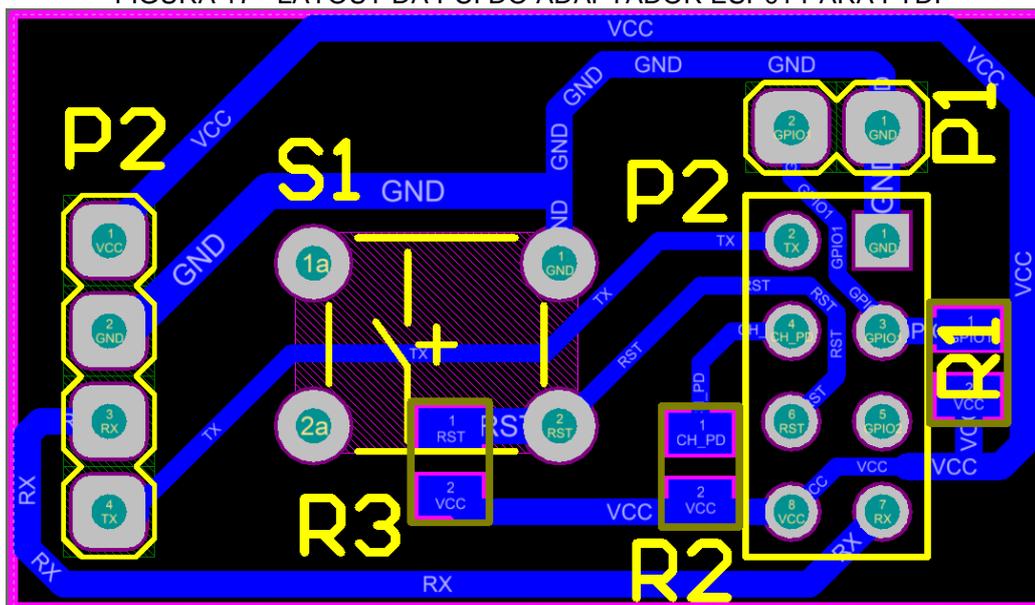
FONTE: O AUTOR (2018)

FIGURA 16 - DIAGRAMA ESQUEMATICO DA LACA ADAPTADORA ESP01 PARA FTDI



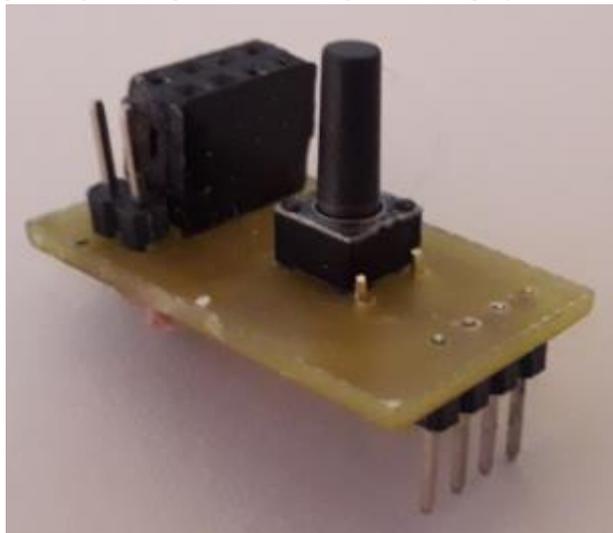
FONTE: O AUTOR (2018)

FIGURA 17 - LAYOUT DA PCI DO ADAPTADOR ESP01 PARA FTDI



FONTE: O AUTOR (2018)

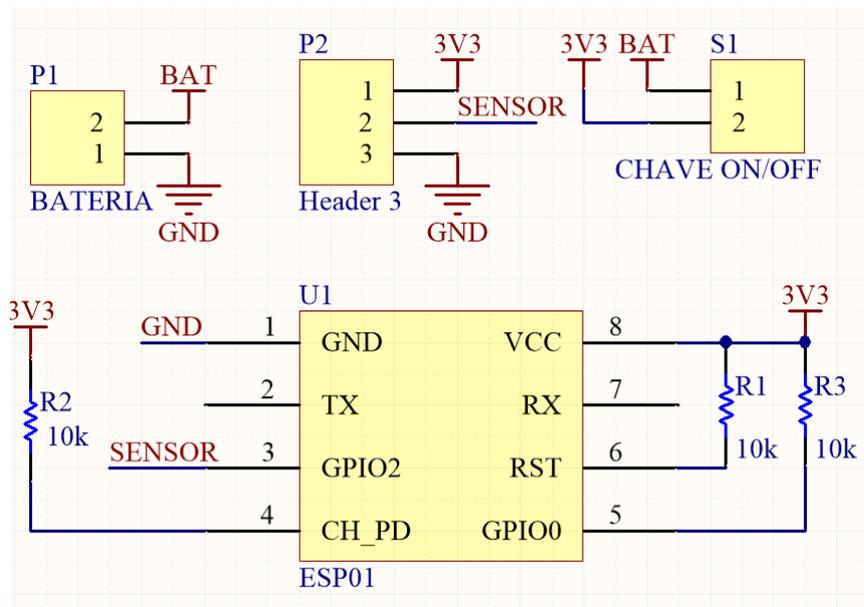
FIGURA 18 - PLACA ADAPTADORA DE ESP01 PARA FTDI



FONTE: O AUTOR (2018)

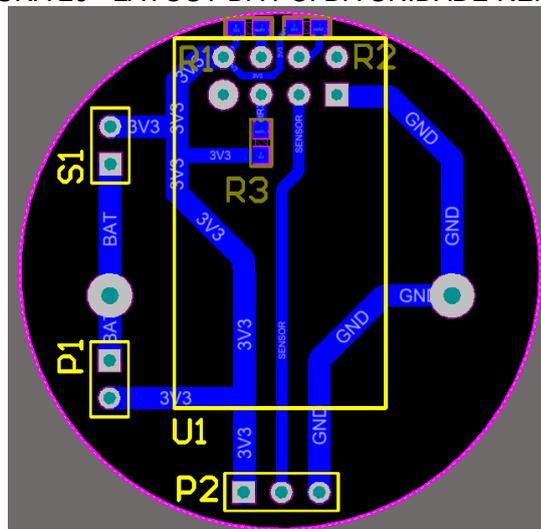
A fim de agregar os componentes da Unidade Remota foi desenvolvida uma placa de circuito impresso (PCI). A figura 19 mostra o diagrama esquemático dessa unidade remota. A partir do diagrama esquemático, o layout da PCI da Unidade Remota foi desenvolvido conforme mostrado na figura 20.

FIGURA 19 - DIAGRAMA ESQUEMÁTICO DA UNIDADE REMOTA



FONTE: O AUTOR (2018)

FIGURA 20 - LAYOUT DA PCI DA UNIDADE REMOTA



FONTE: O AUTOR (2018)

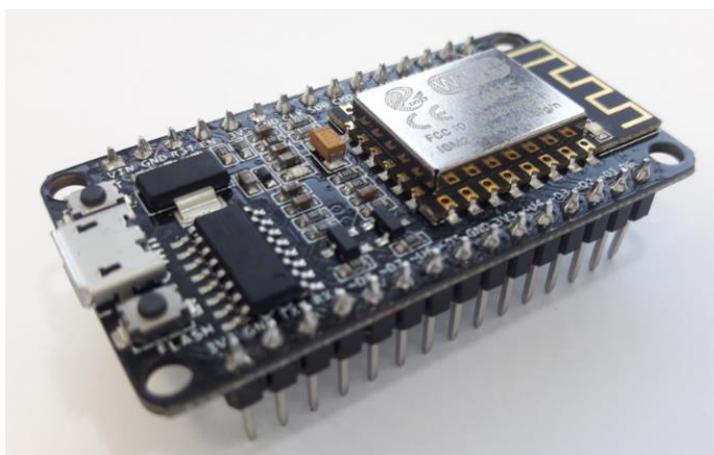
4.2 UNIDADE CENTRAL

A Unidade Central utiliza o microcontrolador ESP8266. O módulo NodeMCU foi escolhido por oferecer acesso a mais funções do microcontrolador, como entradas, saídas e as interfaces de comunicação I2C e SPI, as quais são utilizadas para efetuar a comunicação com os demais dispositivos da unidade central.

4.2.1 ESP8266 NODEMCU

O módulo Wifi ESP8266 NodeMCU é uma placa de desenvolvimento que combina o chip ESP8266, uma interface usb-serial e um regulador de tensão 3.3V. A programação pode ser feita através da IDE Arduino, utilizando comunicação serial com o computador via cabo micro-usb. A FIGURA 21 mostra o módulo NodeMCU.

FIGURA 21 - MÓDULO ESP8266 NODEMCU



FONTE: O AUTOR (2018)

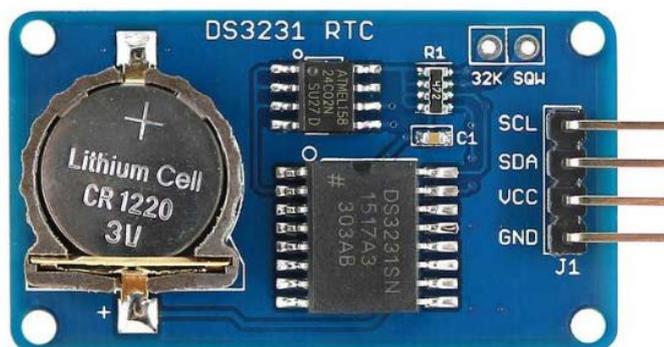
De acordo com a folha de dados do fabricante, o modulo ESP NodeMCU apresenta as especificações abaixo.

- ✓ WIRELESS PADRÃO 802.11 B/G/N;
- ✓ ANTENA EMBUTIDA;
- ✓ CONECTOR MICRO-USB;
- ✓ MODOS DE OPERAÇÃO: STA/AP/STA+AP;
- ✓ SUPORTA 5 CONEXÕES TCP/IP;
- ✓ PORTAS GPIO: 11;
- ✓ GPIO COM FUNÇÕES DE PWM, I2C, SPI;
- ✓ TENSÃO DE OPERAÇÃO: 4,5 ~ 9V;
- ✓ TAXA DE TRANSFERÊNCIA: 110-460.800BPS;
- ✓ SUPORTA *UPGRADE* REMOTO DE *FIRMWARE* (OTA);
- ✓ CONVERSOR ANALÓGICO DIGITAL (ADC).

4.2.2 REFERÊNCIA DE DATA E HORA

O módulo do NodeMCU não possui uma referência de tempo interna, por isso foi necessária a inserção de um módulo RTC DS3231, mostrado na FIGURA 22. Dessa forma é possível registrar a data e hora correta dos eventos medidos. O módulo RTC utiliza comunicação serial I2C.

FIGURA 22 - MODULO RTC DS3231



FONTE: O AUTOR (2018)

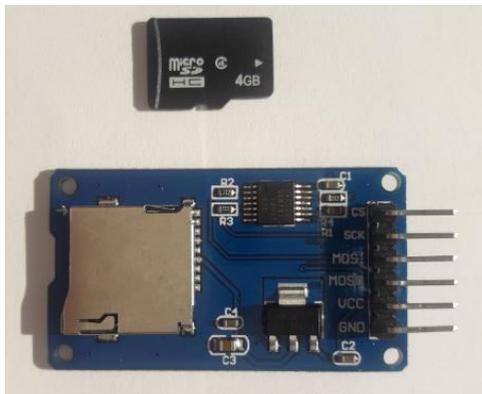
- ✓ TENSÃO DE FUNCIONAMENTO: 3.0 ~ 5.5VDC;
- ✓ CORRENTE DE TRABALHO: 3mA (MAX);
- ✓ INTERFACE: INTERFACE I2C PADRÃO (400KHZ);
- ✓ A EXATIDÃO DO *TIMING*:
- ✓ ± 2 PPM @ 0 ~ + 40°C; 3,5 PPM \pm @ -40 ~ + 85°C;

4.2.3 ARMAZENAMENTO EXTERNO DE DADOS

A fim de expandir a capacidade de memória da Unidade Central, foi inserido no sistema um módulo de cartão SD (FIGURA 23), que se comunica com o

microcontrolador pela interface SPI., gerando um arquivo .CSV.

FIGURA 23 - MÓDULO E CARTÃO MICRO SD



FONTE: O AUTOR (2018)

- ✓ TENSÃO DE ENTRADA: 5V OU 3.3V;
- ✓ INTERFACE SPI: MOSI, SCK, MISO E CS;
- ✓ FORMATAÇÃO: FAT16 OU FAT32.

4.2.4 INTERFACE COM O CONSUMIDOR

O acesso à internet possibilita a disponibilização de informações com a utilização de microcomputadores ou smartphones. Não se pode garantir o acesso a internet no local de instalação do sistema, por isso optou-se por uma interface local para visualização das informações de consumo. A FIGURA 24 mostra o display utilizado para realizar a apresentação dos dados ao consumidor.

FIGURA 24 - DISPLAY DE LCD PARALELO 16 LINHAS X 2 COLUNAS



FONTE: O AUTOR (2018)

- ✓ DISPLAY LCD 16x2
- ✓ BACKLIGHT: VERDE
- ✓ COR DOS CARACTERES: PRETO
- ✓ TENSÃO DE OPERAÇÃO: 5V
- ✓ LINHAS: 2
- ✓ COLUNAS: 16
- ✓ DIMENSÕES: 80 X 36 X 12MM
- ✓ ÁREA VISÍVEL: 64,5 X 16MM

O display disponível, no entanto, não possui interface de comunicação serial, e por isso foi integrado a ele uma placa adaptadora de comunicação paralela para serial i2c, que é mostrada na figura 25.

FIGURA 25 - ADAPTADOR DE DISPLAY PARALELO PARA I2C



FONTE: O AUTOR (2018)

- ✓ ENDEREÇO I2C: 0X20-0X27
- ✓ COMPATÍVEL COM DISPLAY LCD 16x2 E LCD 20x4
- ✓ TENSÃO DE OPERAÇÃO: 5V
- ✓ DIMENSÕES: 55 X 23 X 14MM
- ✓ PESO: 5G

4.2.5 ALIMENTAÇÃO

A Unidade Central é alimentada por uma bateria de Íon Lítio de 3.7 volts, mostrada na FIGURA 26. A carga da bateria é realizada através de um módulo carregador TP4056 (FIGURA 27) com entrada para conector micro USB. Desse modo é possível efetuar o carregamento da bateria da Unidade Central sem interromper seu funcionamento, com a utilização de uma fonte de alimentação de 5 volts, como as

utilizadas para o carregamento de smartphones padrão Android.

FIGURA 26 - BATERIA RECARREGÁVEL UTILIZADA NA UNIDADE CENTRAL



FONTE: O AUTOR (2018)

- ✓ TIPO: LI-ÍON;
- ✓ TENSÃO: 3.7V;
- ✓ CORRENTE: 3800mAh;
- ✓ DIMENSÕES: 18X65mm;
- ✓ PESO: 60G.

FIGURA 27 - MÓDULO CARREGAR DE BATERIA TP4056



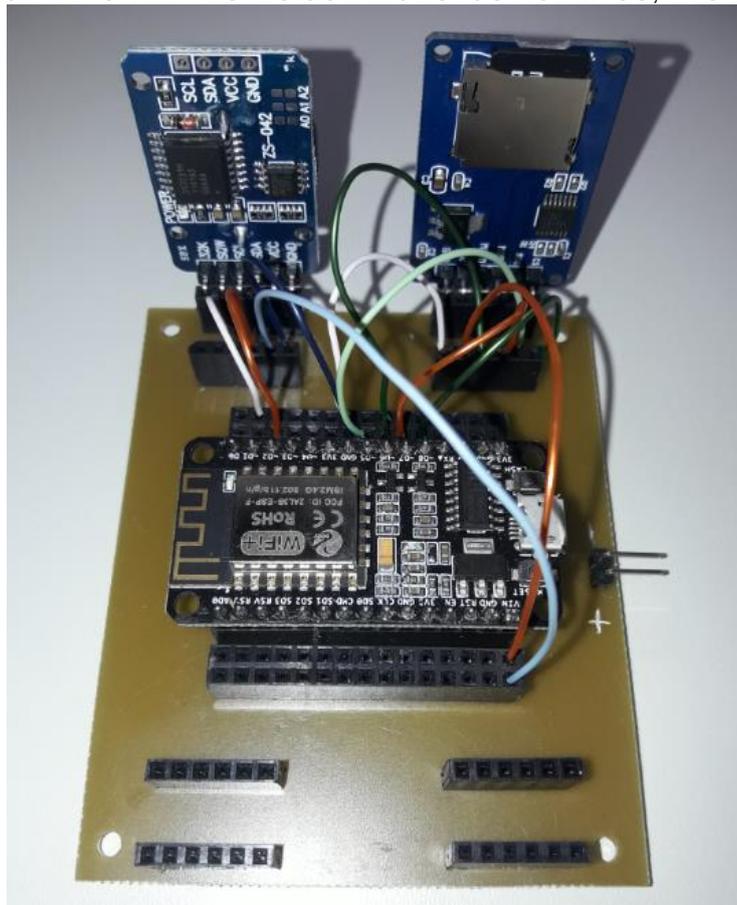
FONTE: O AUTOR (2018)

- ✓ TENSÃO DE OPERAÇÃO: 4.5V A 5V;
- ✓ CAPACIDADE MÁXIMA DE CARGA: 1A;
- ✓ TENSÃO DE CORTE NA SAÍDA: 4.2V +/- 1%;
- ✓ PROTEÇÃO CONTRA SOBRECARGA;
- ✓ CONEXÃO MICRO USB;
- ✓ LEDS INDICADORES;
- ✓ TEMPERATURA DE OPERAÇÃO: -10°C À 85°C

4.2.6 implementação da Unidade Central

Antes de se chegar a versão final do hardware da unidade central os módulos e funções foram testadas por partes. A figura 28 mostra uma das placas auxiliares montadas para testar os módulos RTC e SD CARD.

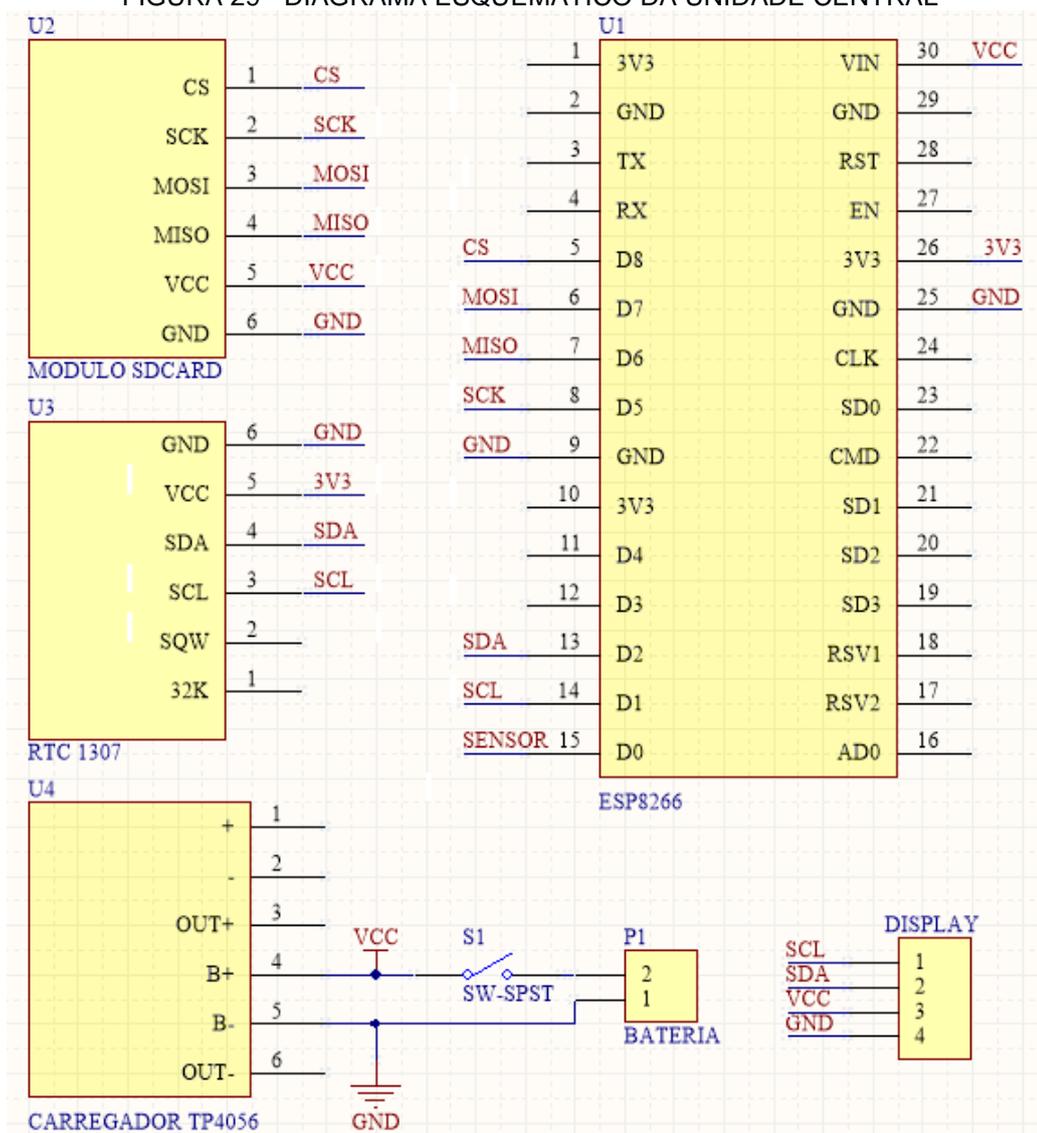
FIGURA 28 – PLACA DE TESTES COM MODULOS NODEMCU, RTC E SD CARD



FONTE: O AUTOR (2018)

Uma vez selecionados os componentes realizados os testes de compatibilidade entre eles, seguiram-se o projeto, fabricação e montagem do protótipo funcional através da construção de placas de circuito impresso para integrar o sistema. Na FIGURA 29 vemos o diagrama esquemático da última versão da Unidade Central.

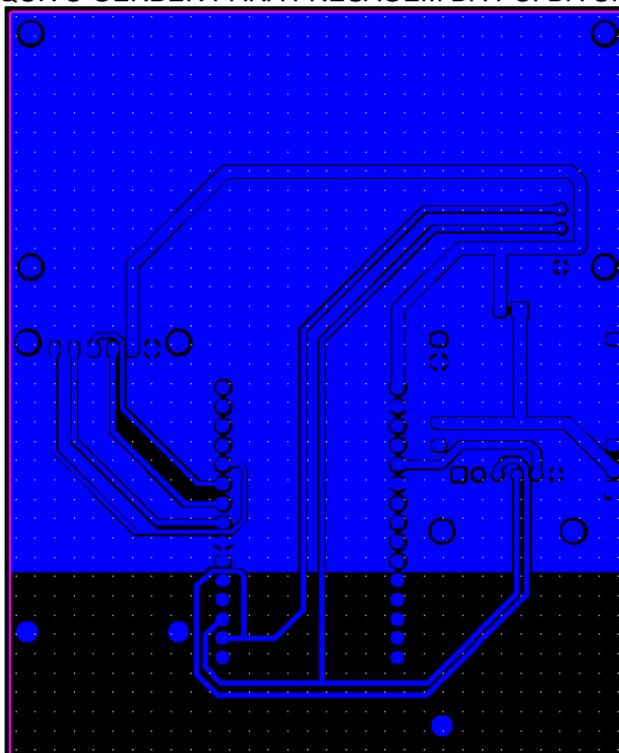
FIGURA 29 - DIAGRAMA ESQUEMÁTICO DA UNIDADE CENTRAL



FONTE: O AUTOR (2018)

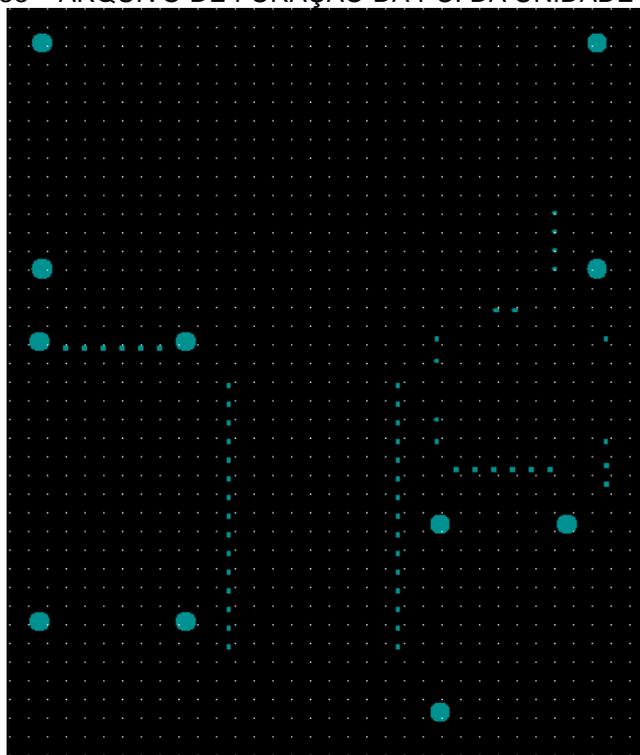
Com base neste esquemático foi gerado o layout da PCI visto na FIGURA 30. A partir do arquivo da PCI foram gerados os arquivos de fabricação CAM de fresagem e furação no CAD Altium Designer (FIGURA 31), conforme mostrado nas FIGURAS 32 e 33.

FIGURA 32 – ARQUIVO GERBER PARA FRESAGEM DA PCI DA UNIDADE CENTRAL



FONTE O AUTOR (2018)

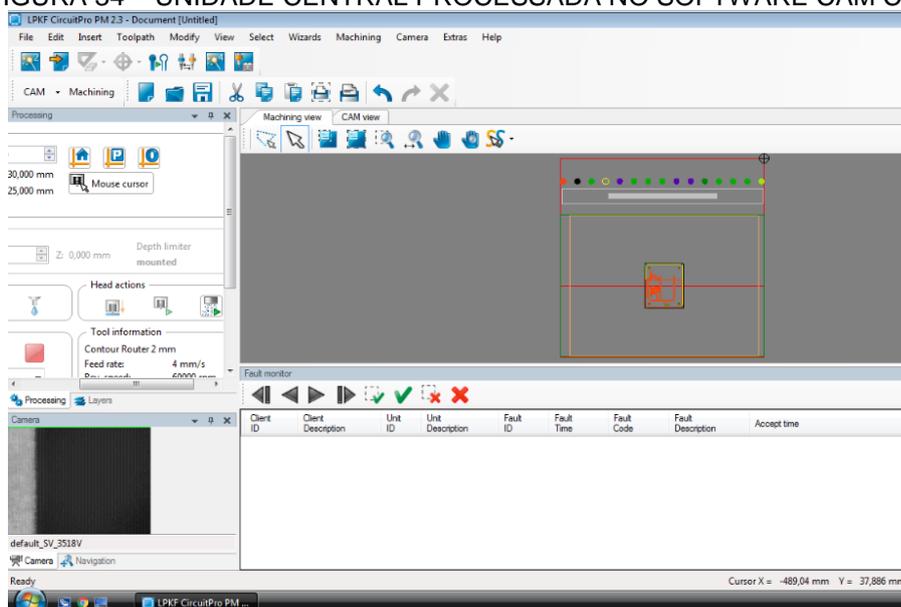
FIGURA 33 – ARQUIVO DE FURAÇÃO DA PCI DA UNIDADE CENTRAL



FONTE: O AUTOR (2018)

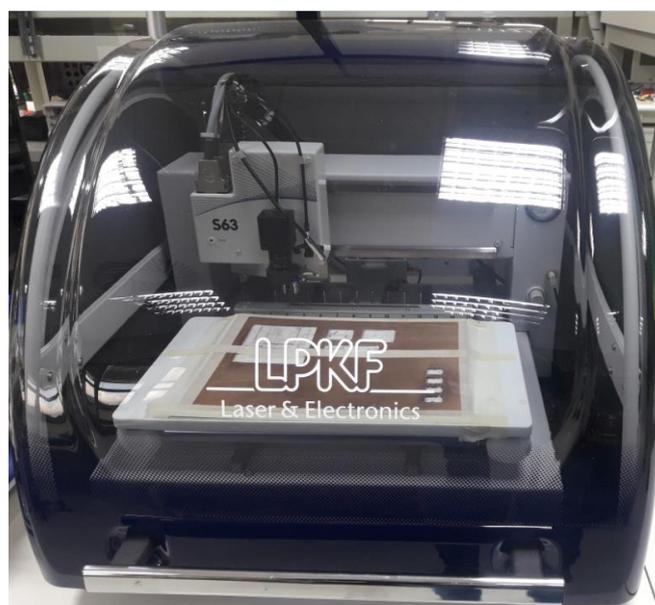
Após a geração dos arquivos de produção estes foram inseridos no software CircuitPRO (FIGURA 34) de gerenciamento de fabricação de placas de circuito impresso da fresadora LPKF S63(FIGURA 35). A FIGURA 36 mostra a PCI da Unidade Central em produção. Os procedimentos para a fabricação das placas auxiliares ao desenvolvimento e da Unidade Remota foram os mesmos descritos nesta seção.

FIGURA 34 – UNIDADE CENTRAL PROCESSADA NO SOFTWARE CAM CIRCUITPRO



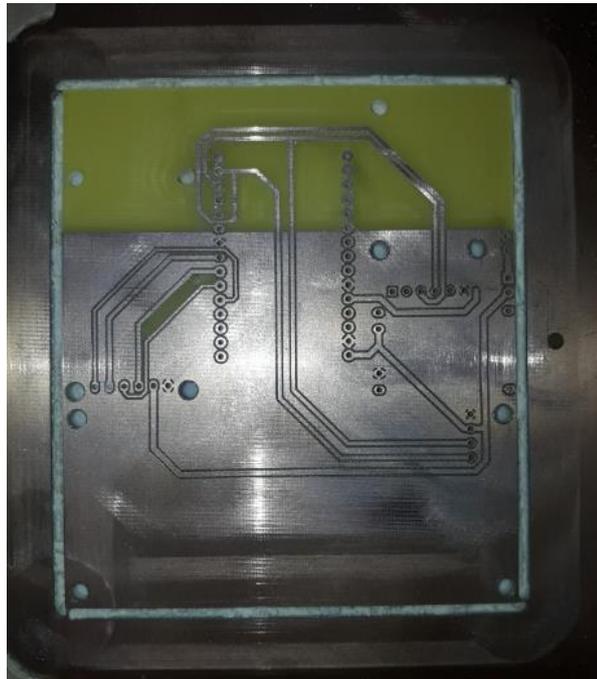
FONTE: O AUTOR (2018)

FIGURA 35 – PROTIPADORA DE PCI PROTOMAT S63 DA LPKF



FONTE: O AUTOR (2018)

FIGURA 36 - FABRICAÇÃO DA PCI DA UNIDADE CENTRAL PELA LPKF PROTOMAT S63



FONTE: O AUTOR (2018)

4.3 SISTEMA DE MONITORAMENTO DO CONSUMO DE ÁGUA

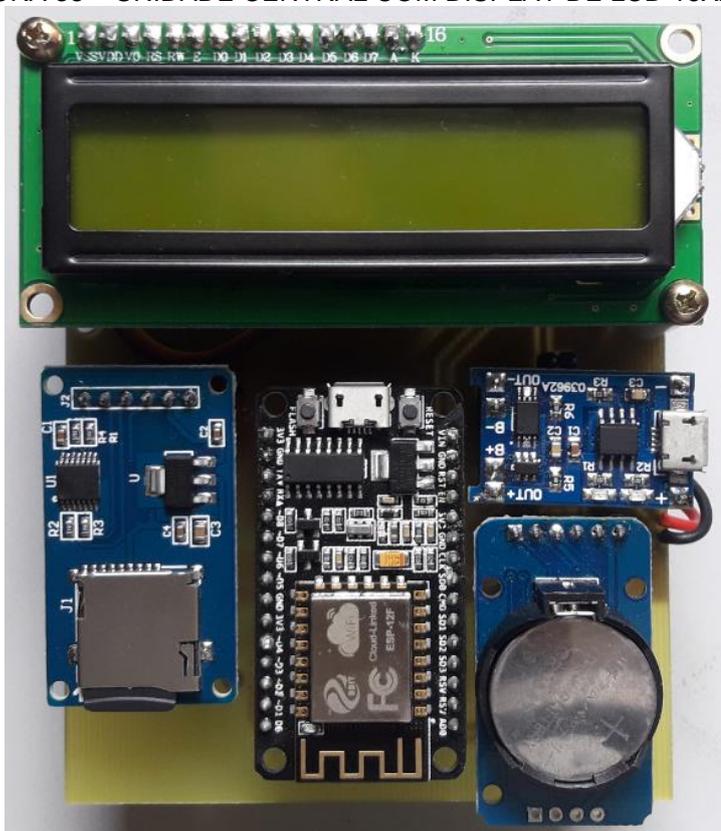
As versões finais das Unidades Central e Unidade Remota podem ser visualizadas nas FIGURAS 37 e 38 respectivamente.

FIGURA 37 – UNIDADE REMOTA ACOPLADA AO SENSOR DE FLUXO



FONTE: O AUTOR (2018)

FIGURA 38 – UNIDADE CENTRAL COM DISPLAY DE LCD 16X2



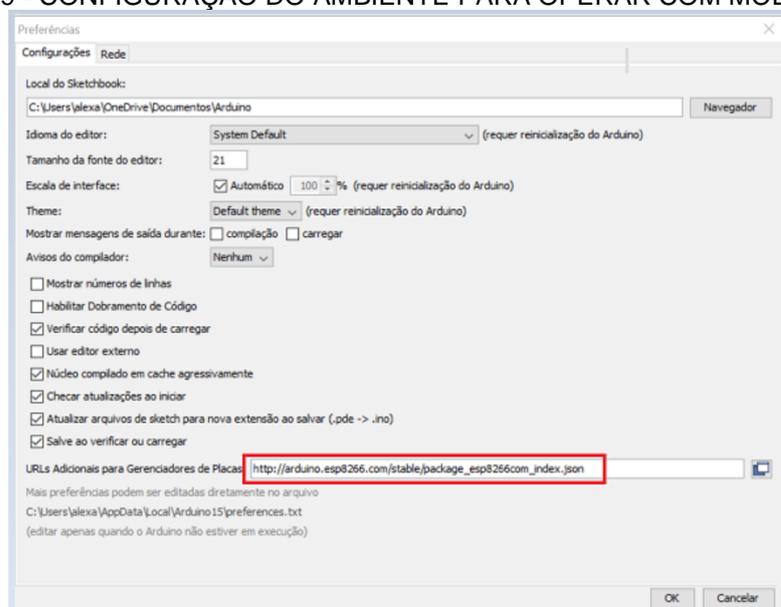
FONTE: O AUTOR (2018)

5 DESENVOLVIMENTO DO FIRMWARE

Apesar de serem coisas distintas os desenvolvimentos de hardware e de firmware caminham juntos e a escolha de cada um dos componentes do hardware foi feita levando em consideração os desenvolvimentos de firmware relacionados a cada um deles. A integração do sensor de fluxo, dos módulos da Espressif, módulos RTC, cartão SD, carregador de bateria e display foi realizada com base em códigos desenvolvidos para o ambiente de desenvolvimento Arduino e disponibilizados na plataforma de hospedagem com controle de versão GitHub. As funções de comunicação Wifi com protocolo UDP foram adaptadas de MORAIS (2018) que disponibilizou um código para a comunidade no portal Vida de Silício.

Para realizar a programação e a gravação do firmware nos dispositivos ESP, é necessário instalar um *patch* no ambiente de desenvolvimento Arduino, conforme mostrado na FIGURA 39. Esse *patch* permite incluir bibliotecas desenvolvidas para os módulos ESP, necessárias para a implementação dos códigos.

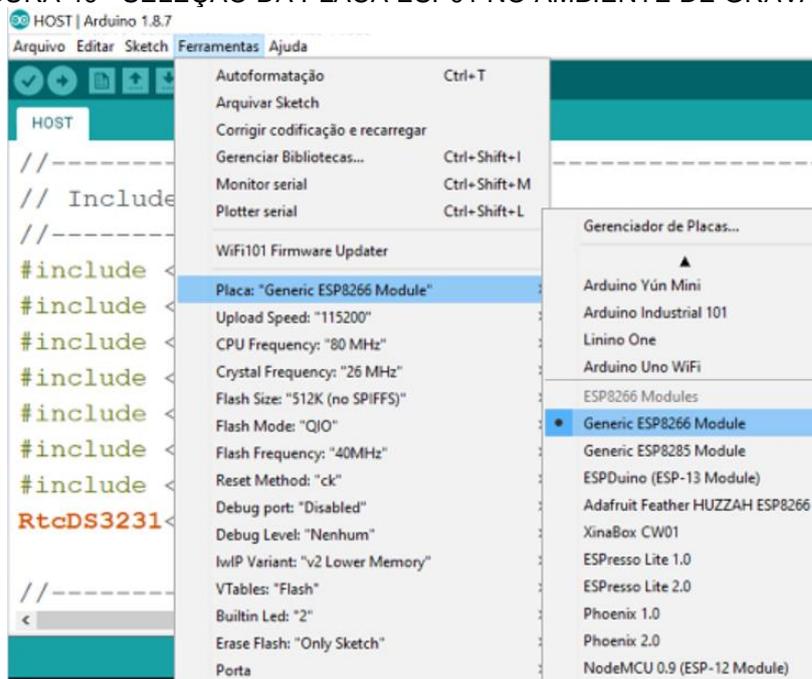
FIGURA 39 - CONFIGURAÇÃO DO AMBIENTE PARA OPERAR COM MÓDULOS ESP



FONTE: O AUTOR (2019)

A FIGURA 40 mostra a tela de seleção do dispositivo a ser programado, sendo o ESP01 configurado como Generic ESP8266 module e o ESP NodeMCU como NodeMCU 1.0 (esp-12 module).

FIGURA 40 - SELEÇÃO DA PLACA ESP01 NO AMBIENTE DE GRAVAÇÃO



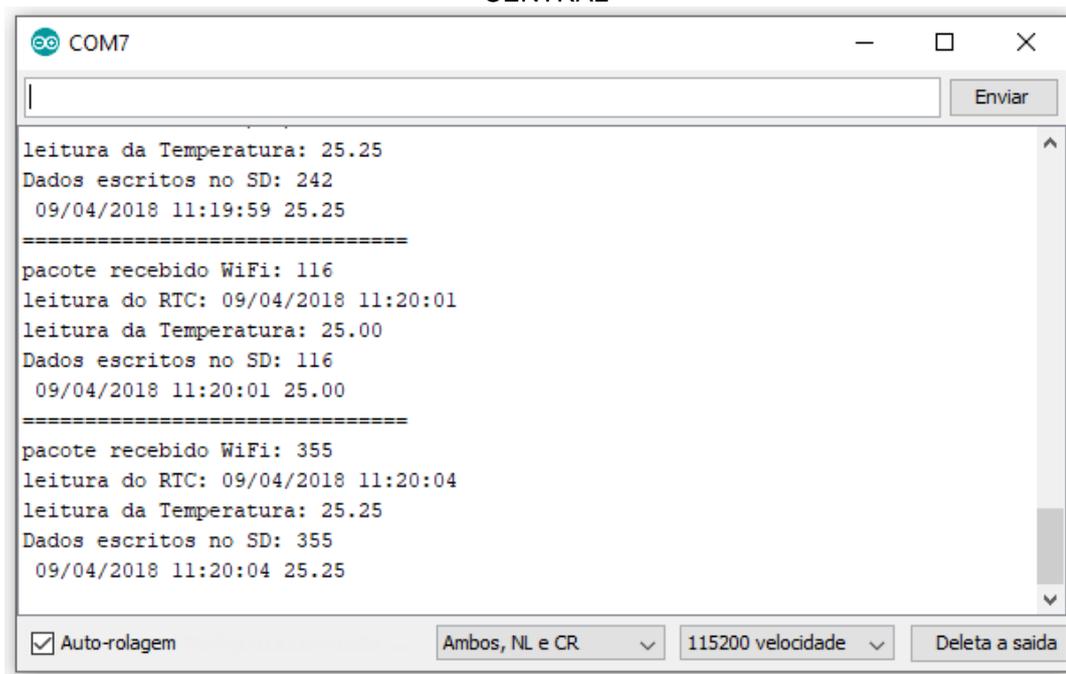
FONTE: O AUTOR (2018)

5.1 COMUNICAÇÃO ENTRE DISPOSITIVOS DE AQUISIÇÃO E DISPONIBILIZAÇÃO DOS DADOS

Inicialmente foram pesquisadas funções simples de teste para cada um dos dispositivos de hardware. O primeiro teste realizado foi o de comunicação Wifi entre dois dispositivos Espressif, sendo um deles um módulo ESP WEMOS, utilizado como simulador da Unidade Remota (Cliente) e um módulo NodeMCU utilizado como Unidade Central (Servidor). O firmware gravado na Unidade Central cria uma rede Wifi com senha e abre um canal de comunicação utilizando o protocolo UDP, o qual espera por uma conexão e imprime na interface serial do ambiente de desenvolvimento os dados recebidos. O protocolo UDP é extremamente simples, ele não realiza conexões, para transmitir uma informação basta enviar um datagrama, ou seja, um pacote de dados, para um endereço IP e uma porta pré-definidos (COSTA, 2008). No firmware da Unidade Central foi criada uma rede com endereço IP: 192.168.4.1 e porta 555. O firmware da Unidade Remota conecta-se à essa rede criada, desde que seja especificado em seu código o nome e a senha da rede criadas pela Unidade Central.

Em um primeiro momento, para efeitos de teste, foi desenvolvido um firmware que gera e transmite um número aleatório como informação de entrada. Seguindo o mesmo modelo de testes, firmwares simples de exemplos disponíveis no GitHub, foram implementados na Unidade Central a fim de integrar a ela os módulos RTC e SD CARD. Nesta etapa a Unidade Central recebe os números aleatórios enviados pela Unidade Remota e registra-os em um cartão SD criando um arquivo CSV com a hora fornecida pelo relógio de tempo real . A FIGURA 41 mostra os números aleatórios recebidos pela comunicação WiFi impressos na tela do ambiente de desenvolvimento e gravados no cartão SD com a referência de tempo provida pelo modulo RTC e a FIGURA 42 mostra os dados gravados no arquivo CSV.

FIGURA 41 - TESTE DE COMUICAÇÃO ENTRE UNIDADES UNIDADE REMOTA E UNIDADE CENTRAL



FONTE: O AUTOR (2018)

FIGURA 42- ARQUIVO .CSV GRAVADO NO CARTÃO SD COM INFORMAÇÕES DE CONSUMO, DATA E HORA

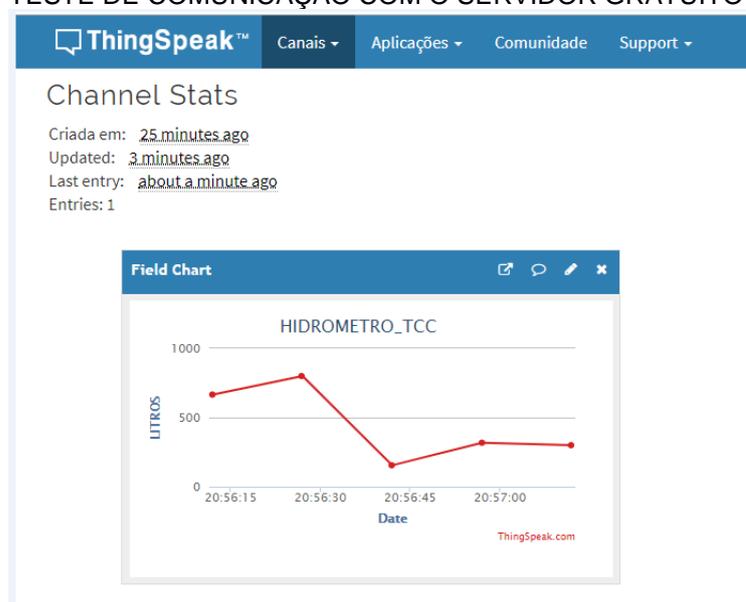
```

21:31:31 24.750.00 10/30/2018
21:31:35 24.750.00 10/30/2018
21:31:39 25.000.00 10/30/2018
21:31:43 25.000.00 10/30/2018
  
```

FONTE: O AUTOR (2018)

Paralelamente a este sistema de comunicação com a Unidade Central foi testada uma opção de comunicação da Unidade Remota diretamente com um servidor externo com disponibilização de um datalogger gratuito chamado ThingSpeak. A FIGURA 43 mostra a tela do site com as informações atualizadas em tempo real. Este cenário prevê que o consumidor terá sempre acesso à internet, por isso ele foi descartado para esta aplicação. Em locais com acesso à internet, além do armazenamento e consulta na Nuvem, é possível registrar os eventos com a data e hora fornecida pelo servidor externo.

FIGURA 43 - TESTE DE COMUNICAÇÃO COM O SERVIDOR GRATUITO THINGSPEAK



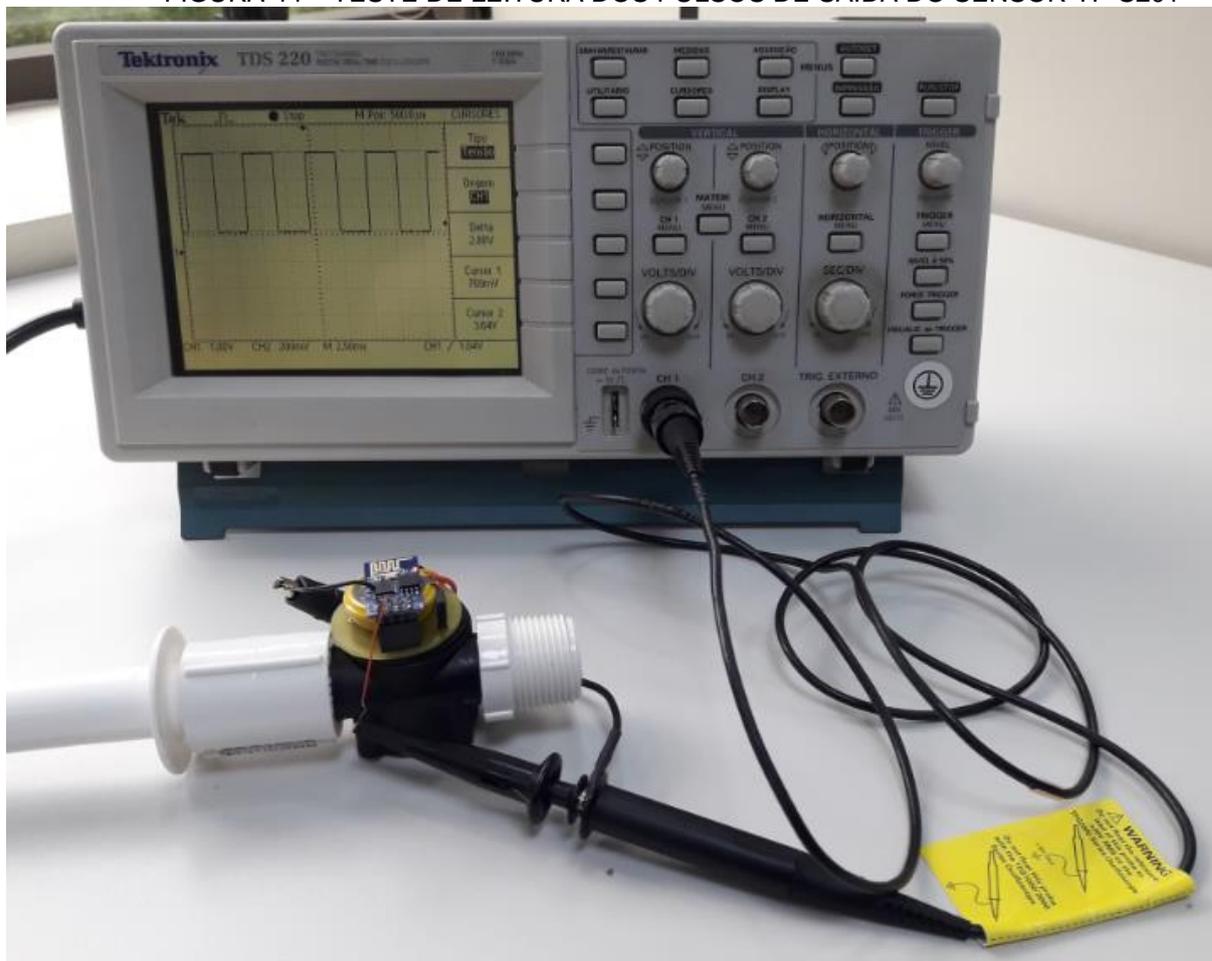
FONTE: O AUTOR (2018)

5.2 INSERÇÃO DA INFORMAÇÃO REAL DE CONSUMO NO SISTEMA.

Após a verificação das funções de comunicação e armazenamento, partiu-se para a leitura real dos dados de consumo de água fornecidos pelo sensor de vazão. O firmware gravado na unidade remota funciona com a habilitação de uma interrupção do processador a cada borda de subida gerada pelo sensor de efeito hall. Dentro desta interrupção é chamada uma função que registra a contagem de pulsos de forma acumulativa transmitindo esse número total de pulsos para a Unidade Central em tempo real. Na Central é feita a conversão do número de pulsos em volume de água consumido, já que cada pulso equivale a uma vazão de 2.25ml. A FIGURA 44 mostra

a ligação do osciloscópio à saída do sensor e a forma de onda medida com a passagem do fluxo através deste. Cada giro da turbina corresponde a um pulso e nesse teste o fluxo de água foi simulado com ar, visto que a finalidade deste teste era testar a transmissão e recepção dos dados.

FIGURA 44 – TESTE DE LEITURA DOS PULSOS DE SAÍDA DO SENSOR YF-S201



FONTE: O AUTOR (2018)

5.3 VIZUALIZAÇÃO DO CONSUMO.

Como o sistema visa auxiliar o consumidor a ter a noção do consumo em tempo real, optou-se por disponibilizar a informação da vazão de forma acumulada e também manter a informação de consumo do mês anterior. Dessa forma, na primeira linha do display a informação fornecida no campo “T” representa o consumo total acumulado desde a instalação do sistema até a virada do mês, enquanto que na linha 2 do display é mostrado no campo “A” o consumo do mês corrente. Para realizar esta função foi preciso extrair da String de data e hora fornecidos pelo relógio do tempo

real a informação do mês atual dentro de uma função e compará-la com essa mesma informação do mês gravada em uma variável global. No momento em que essas duas variáveis, local e global se tornam diferentes ocorre a mudança do mês e a informação da linha 2 do display passa para a linha 1, enquanto que a informação mostrada na linha dois é zerada e começa a mostrar o consumo desde o início do mês. Como a quantidade de água consumida não pode ser maior do que 10 m^3 optou-se por mostrar a quantidade consumida em litros. Como cada 1000 litros corresponde a 1 m^3 o consumidor próximo de atingir 10.000 litros deve tomar medidas para evitar este gasto excessivo, o qual poderia implicar na perda do benefício da tarifa social. A FIGURA 45 mostra as informações de consumo anterior e atual no display da unidade central.

FIGURA 45 – INFORMAÇÕES DE CONSUMO DOS MESES ANTERIOR E ATUAL



FONTE: O AUTOR (2018)

6 CONSIDERAÇÕES FINAIS E TRABALHOS FUTUROS

O resultado final obtido foi satisfatório, tendo em vista o tempo de desenvolvimento, os materiais e equipamentos disponíveis para a implementação do projeto. O sistema hoje é capaz de medir o fluxo de água através do sensor e apresentar em um display de forma acumulativa, em tempo real, o volume consumido em litros. Se implementado em uma residência, o consumidor será capaz de tomar medidas de redução no consumo a fim de preservar o benefício da tarifa social adquirido. Com o avanço e a popularização das tecnologias é possível criar sistemas modulares de acordo com as necessidades do cliente. A integração de módulos que realizam diferentes funções é possível graças ao esforço da comunidade que desenvolve e disponibiliza soluções em eletrônica.

O projeto apresenta os elementos chave de grande parte dos projetos de engenharia elétrica, desenvolvendo assim a competência necessária para a implementação de circuitos sensores, transmissores e de armazenamento de dados.

O caráter social do projeto é uma importante forma de retribuição para a sociedade por parte da universidade e a integração com o departamento de pós-graduação de design da UFPR estimula a percepção da necessidade de interação entre diversos setores acadêmicos na busca por soluções e alternativas aos sistemas ineficazes e, ou obsoletos. Um sistema eletrônico pode ser sempre aprimorado, conforme novas tecnologias sejam desenvolvidas, porém, uma vez que a solução atinja seu objetivo tem-se que trabalhar de modo a manter o sistema funcionando de maneira mais robusta possível. Existem meios de projetar sistemas que possibilitem atualizações de firmware e hardware mantendo a compatibilidade. Uma dessas tecnologias já disponíveis e compatíveis com os módulos *WiFi* da Espressif é chamada de OTA (*over the air*), permitindo realizar atualizações de firmware de maneira Unidade Remota. A questão da alimentação do sistema também pode ser aprimorada como o estudo de sensores que possam gerar a energia necessária para o carregamento das baterias do próprio sistema. A comunicação em redes *mesh* também é uma tendência para o projeto, visto que cada Unidade Central individual pode vir a se conectar com as demais até que uma delas possa fornecer as informações de consumo global para uma Central que possua acesso à internet, disponibilizando o acesso e garantindo o armazenamento dos dados na nuvem, de onde podem ser analisados a fim de traçar perfis de consumo individuais e coletivos.

REFERÊNCIAS

OLIVEIRA, SERGIO. **INTERNET DAS COISAS COM ESP8266, ARDUINO E RASPBERRY PI**. EDIÇÃO 1. LOCAL: SÃO PAULO: NOVATEC, 2017. 16 P.

COSTA, DANIEL GOUVEIA. **JAVA EM REDE: RECURSOS AVANÇADOS DE PROGRAMAÇÃO**. EDIÇÃO 1. LOCAL: RIO DE JANEIRO: BRASPORT, 2008.60 P

E. CURRY, W. DERGUECH, S. HASAN, C. KOUROUPETROGLOU, U. UL HASSAN, **A REAL-TIME LINKED DATASPACE FOR THE INTERNET OF THINGS: ENABLING “PAY-AS-YOU-GO” DATA MANAGEMENT IN SMART ENVIRONMENTS**, FUTURE GENERATION COMPUTER SYSTEMS (2018)

THULO RAM GURUNG, RODNEY A. STEWART, ASHOK K. SHARMA, CARA D. BEAL, **SMART METERS FOR ENHANCED WATER SUPPLY NETWORK MODELLING AND INFRASTRUCTURE PLANNING** RESOURCES, CONSERVATION AND RECYCLING, SCIENCE DIRECT (2014)

B. GURAGAI A, T. HASHIMOTO A, K. OGUMA B, S. TAKIZAWA, **DATA LOGGER-BASED MEASUREMENT OF HOUSEHOLD WATER CONSUMPTION AND MICRO-COMPONENT ANALYSIS OF AN INTERMITTENT WATER SUPPLY SYSTEM**, JOURNAL OF CLEANER PRODUCTION, SCIENCE DIRECT (2018)

SITE OFICIAL DA SANEPAR – **TARIFA SOCIAL** - DISPONÍVEL EM: < <http://site.sanepar.com.br/clientes/tarifa-social-da-sanepar> >. ACESSO EM 23 DE NOVEMBRO DE 2018.

SITE OFICIAL DA SANEPAR – **SIMULADOR DE TARIFAS** - DISPONÍVEL EM: < <http://site.sanepar.com.br/informacoes/simulador-de-tarifas>>. ACESSO EM 23 DE NOVEMBRO DE 2018.

MORAIS, JOSÉ – PORTAL VIDA DE SILICIO – **COMUNICAÇÃO WIRELESS UDP COM ESP** - DISPONÍVEL EM: < <https://portal.vidadesilicio.com.br/comunicacao-wireless-esp-protocolo-udp/>>. ACESSO EM 23 DE AGOSTO DE 2018.

MINISTÉRIO DO PLANEJAMENTO, DESENVOLVIMENTO E GESTÃO – **HABITAÇÃO DE INTERESSE SOCIAL** - DISPONÍVEL EM: < <http://www.planejamento.gov.br/assuntos/gestao/patrimonio-da-uniao/destinacao-de-imoveis/habitacao-de-interesse-social>>. ACESSO EM 23 DE NOVEMBRO DE 2018.

7 APENCICE 1 – FIRMWARE DA UNIDADE REMOTA

```

1  #INCLUDE <ESP8266WIFI.H>//BIBLIOTECA DO WIFI.
2  #INCLUDE <WIFIUDP.H>//BIBLIOTECA DO UDP.
3  #INCLUDE <SPI.H>
4
5  WIFIUDP UDP;//CRIA UM OBJETO DA CLASSE UDP.
6  LONG X;//VARIÁVEL PARA SER ENVIADA.
7
8  VOLATILE INT PULSOS; //VARIÁVEL PARA A QUANTIDADE DE PULSOS
9  INT I=0; //VARIÁVEL PARA CONTAGEM
10
11 CONST INT INTER = 2;// CONFIGURANDO GPIO4 COMO INTERRUPTÃO
12
13 VOID SETUP()
14 {
15     //PINMODE(LED, OUTPUT);//HABILITA O LED ONBOARD COMO SAIDA.
16     //DIGITALWRITE(LED, 1);//DESLIGA O LED.
17
18     WIFI.MODE(WIFI_STA);//DEFINE O ESP8266 COMO STATION.
19     SERIAL.BEGIN(115200);
20     SERIAL.PRINTLN("SERIAL OK");
21     PINMODE(INTER, INPUT);
22     ATTACHINTERRUPT(INTER, FLUXO, RISING); //CONFIGURA O PINO
2 (INTERRUPÇÃO 0) PARA TRABALHAR COMO INTERRUPTÃO
23     SERIAL.PRINTLN("\n\nINICIO\n\n"); //IMPRIME INICIO NA SERIAL
24     CONNECT();//SUB-ROTINA PARA CONECTAR-SE AO HOST.
25     SEI(); //HABILITA INTERRUPTÃO
26
27 }
28
29 VOID LOOP()
30 {
31
32     SEND();//SUB-ROTINA PARA ENVIAR OS DADOS AO HOST.
33
34     DELAY(2000);//AGUARDA MEIO SEGUNDO.
35 }
36
37 VOID CONNECT()//SUB-ROTINA PARA VERIFICAR A CONEXAO COM O HOST.
38 {
39     IF (WIFI.STATUS() != WL_CONNECTED)//CASO NAO ESTEJA CONECTADO
AO HOST, IRA SE CONECTAR.
40     {
41         WIFI.BEGIN("NODEMCU", "1234");//CONECTA À REDE DO HOST.
42
43         WHILE (WIFI.STATUS() != WL_CONNECTED) {
44             DELAY(500);
45             SERIAL.PRINT(".");
46         }
47
48         IF (WIFI.STATUS() == WL_CONNECTED)
49         {
50             SERIAL.PRINTLN("CONECTADO...");
51             SERIAL.PRINTLN("IP ADDRESS: ");
52             SERIAL.PRINT(WIFI.LOCALIP());
53         }
54         ELSE
55         {
56             SERIAL.PRINTLN("NAO DEU...");

```

```

57     }
58
59   }
60 }
61
62 VOID SEND()//SUB-ROTINA PARA ENVIAR DADOS AO HOST.
63 {
64   IF (WIFI.STATUS() == WL_CONNECTED)//SÃ³ IRA ENVIAR DADOS SE
ESTIVER CONECTADO.
65   {
66     X = PULSOS; //CONTAGEM DE PULSOS DO SENSOR
67
68     UDP.BEGINPACKET("192.168.4.1", 555);//INICIALIZA O PACOTE
DE TRANSMISSAO AO IP E PORTA.
69     UDP.PRINTLN(X);//ADICIONA-SE O VALOR AO PACOTE.
70     UDP.ENDPACKET();//FINALIZA O PACOTE E ENVIA.
71
72     // DIGITALWRITE(LED, 0);//-
73     DELAY(5);//-
74     //DIGITALWRITE(LED, 1);//PISCA O LED RAPIDAMENTE APOS ENVIAR.
75   }
76   ELSE//CASO NAO ESTEJA COM UMA CONEXAO ESTABELICIDA AO HOST,
PISCARA LENTAMENTE.
77   {
78     //DIGITALWRITE(LED, 0);
79     DELAY(250);
80     //DIGITALWRITE(LED, 1);
81   }
82 }
83
84 VOID FLUXO()
85 {
86   PULSOS++; //INCREMENTA A VARIÃVEL DE CONTAGEM DOS PULSOS
87
88 }

```

8 APENCICE 2 – FIRMWARE DA UNIDADE CENTRAL

```

1 //-----
2 // INCLUDES
3 //-----
4 #INCLUDE <ESP8266WIFI.H>//BIBLIOTECA DO WIFI.
5 #INCLUDE <WIFIUDP.H>//BIBLIOTECA DO UDP.
6 #INCLUDE <SPI.H>           // 1 BIBLIOTECA DE COMUNICAÇÃO SPI
NATIVA
7 #INCLUDE <SD.H>           // 1 BIBLIOTECA DE COMUNICAÇÃO COM
CARTÃO SD NATIVA
8 #INCLUDE <WIRE.H> // MUST BE INCLUDED HERE SO THAT ARDUINO LIBRARY
OBJECT FILE REFERENCES WORK
9 #INCLUDE <RTCD3231.H>
10 #INCLUDE <LIQUIDCRYSTAL_I2C.H>
11 RTCD3231<TWOWIRE> RTC(WIRE);
12
13 //-----
14 // DECLARAÇÃO DE VARIÁVEIS GLOBAIS
15 //-----
16 CONST INT CHIPSELECT = D8;// 2 CONSTANTE QUE INDICA EM QUAL PINO
ESTÁ; CONECTADO O CHIP SELECT DO MÓDULO DE COMUNICAÇÃO
17 FILE DATAFILE;           // 2 OBJETO RESPONSÁVEL POR ESCREVER/LER
DO CARTÃO SD
18 BOOL CARTAOK = TRUE;     // 2 VARIÁVEL DO TIPO BOOL, RESPONSÁVEL
POR INFORMAR SE A COMUNICAÇÃO COM O CARTÃO SD ESTÁ; FUNCIONANDO OU NÃO.
19 WIFIUDP UDP;             //CRIA UM OBJETO DA CLASSE UDP.
20 STRING DADOSWIFI;
21 STRING DADOSRTC;
22 STRING DADOSTEMP;
23 STRING DADOSSD;
24 DOUBLE VOLUME;
25 UINT8 T MES;
26 DOUBLE CONSUMO_TOTAL = 0;
27 DOUBLE CONSUMO_MESANT = 0;
28 DOUBLE CONSUMO_MESAT = 0;
29 LIQUIDCRYSTAL_I2C LCD(0X27, 2, 1, 0, 4, 5, 6, 7, 3, POSITIVE);
30 //-----
31 // FUNÇÃO DE SETUP
32 //-----
33 VOID SETUP ()
34 {
35     PINMODE(D4, OUTPUT);//HABILITA O LED ONBOARD COMO SAIDA.
36     DIGITALWRITE(D4, 1);//DESLIGA O LED.
37
38     SERIAL.BEGIN(115200);//HABILITA A COMUNICAÇÃO SERIAL PARA A
STRING RECEBIDA SER LIDA NO SERIAL MONITOR.
39     LCD.BEGIN (16, 2);
40     LCD.SETBACKLIGHT(HIGH);
41     LCD.SETCURSOR(0, 0);
42     LCD.PRINT("T: ");
43     LCD.SETCURSOR(0, 1);
44     LCD.PRINT("A: ");
45     WIFI.MODE(WIFI_AP);//DEFINE O ESP8266 COMO ACESS POINT.
46     WIFI.SOFTAP("NODEMCU", "1234");//CRIA UM WIFI DE NOME "NODEMCU"
E SEM SENHA.
47     DELAY(2000);//AGUARDA 2 SEGUNDOS PARA COMPLETAR A CRIAÇÃO DO
WIFI.
48     SERIAL.PRINT("IP ADDRESS: ");
49     SERIAL.PRINTLN(WIFI.SOFTAPIP());

```

```

50     UDP.BEGIN(555); //INICIALIZA A RECEPÇÃO DE DADOS UDP NA PORTA
555
51
52     //CONFIGURA RTC
53     RTC.BEGIN();
54     RTCDATETIME COMPILED = RTCDATETIME(__DATE__, __TIME__);
55     //RTC.SETDATETIME(__DATE__, __TIME__);
56     RTC.SETDATETIME(COMPILED);
57     // NEVER ASSUME THE RTC WAS LAST CONFIGURED BY YOU, SO
58     // JUST CLEAR THEM TO YOUR NEEDED STATE
59     RTC.ENABLE32KHZPIN(FALSE);
60     RTC.SETSQUAREWAVEPIN(DS3231SQUAREWAVEPIN_MODENONE);
61
62     // 2 SE NÃO FOR POSSÁVEL SE COMUNICAR COM O CARTÃO SD O
CÓDIGO NÃO DEVE PROSSEGUIR
63     IF (!SD.BEGIN(CHIPSELECT)) {
64         SERIAL.PRINTLN("ERRO NA LEITURA DO ARQUIVO NÃO EXISTE UM
CARTÃO SD OU O MÓDULO ESTÁ CONECTADO CORRETAMENTE ?");
65         CARTAOK = FALSE;
66         RETURN;
67     }
68 }
69
70 //-----
71 // FUNÇÃO PRINCIPAL
72 //-----
73 VOID LOOP()
74 {
75     DADOSWIFI = LISTEN(); //SUB-ROTINA PARA VERIFICAR A EXISTENCIA
DE PACOTES UDP.
76     VOLUME = (DOUBLE) DADOSWIFI.TOINT();
77     VOLUME = VOLUME*2.25;
78     CONSUMO_MESAT = VOLUME - CONSUMO_MESANT;
79     //LCD.SETCURSOR(0,1);
80     //LCD.PRINT(STRING(VOLUME));
81     LCD.SETCURSOR(3, 0);
82     LCD.PRINT(STRING(VOLUME) + "L          ");
83     LCD.SETCURSOR(3, 1);
84     LCD.PRINT(STRING(CONSUMO_MESAT) + "L          ");
85     DADOSRTC = LEITURARTC(); // LEITURA DO RTC
86     DADOSTEMP = LEITURATEMP(); // LEITURA DA TEMPERATURA
87     DADOSSD = STRING(VOLUME) + " " + DADOSRTC + " " + DADOSTEMP;
88     GRAVARSD(DADOSSD);
89
90     SERIAL.PRINTLN("=====");
91     SERIAL.PRINT("PACOTE RECEBIDO WIFI: ");
92     SERIAL.PRINT(VOLUME);
93     SERIAL.PRINT("LEITURA DO RTC: ");
94     SERIAL.PRINTLN(DADOSRTC);
95     SERIAL.PRINT("LEITURA DA TEMPERATURA: ");
96     SERIAL.PRINTLN(DADOSTEMP);
97     SERIAL.PRINT("DADOS ESCRITOS NO SD: ");
98     SERIAL.PRINTLN(DADOSSD);
99
100     SERIAL.PRINTLN("TOTAL: " + STRING(VOLUME) + "L");
101     SERIAL.PRINTLN("ATUAL: " + STRING(CONSUMO_MESAT) + "L");
102
103     CHECAMES();
104     DELAY(2000); // AGUARDAMOS 2 SEGUNDOS PARA EXECUTAR O LOOP
NOVAMENTE
105 }

```

```

106
107 //-----
108 // FUNÇÕES DO CARTÃO SD
109 //-----
110 VOID GRAVARSD (STRING DADOS) {
111     // CASO O CARTÃO ESTEJA OK É CRIADO UM ARQUIVO NO CARTÃO SD
COM O NOME DATALOG.CSV EM MODO DE ESCRITA
112     IF (CARTAOOK) {
113         DATAFILE = SD.OPEN("DATALOG.CSV", FILE_WRITE);
114     }
115     IF (DATAFILE) {
116         DATAFILE.PRINT(DADOS); // ESCRIVEMOS NO ARQUIVOS
117         DATAFILE.CLOSE(); // FECHAMOS O ARQUIVO
118     }
119 }
120
121 //-----
122 // FUNÇÃO DE LEITURA DE DADOS WIFI
123 //-----
124 STRING LISTEN()//SUB-ROTINA QUE VERIFICA SE HÁ; PACOTES UDP'S
PARA SEREM LIDOS.
125 {
126     STRING REQ;//STRING QUE ARMAZENA OS DADOS RECEBIDOS PELA REDE.
127     IF (UDP.PARSEPACKET() > 0)//SE HOUVER PACOTES PARA SEREM LIDOS
128     {
129         REQ = "";//RESETA A STRING PARA RECEBER UMA NOVA INFORMAÇÃO
130         WHILE (UDP.AVAILABLE() > 0)//ENQUANTO HOUVER DADOS PARA SEREM
LIDOS
131         {
132             CHAR Z = UDP.READ();//ADICIONA O BYTE LIDO EM UMA CHAR
133             REQ += Z;//ADICIONA O CHAR À STRING
134         }
135
136         RETURN REQ;
137     }
138 }
139
140 //-----
141 // FUNÇÕES DO RTC
142 //-----
143 STRING LEITURARTC() {
144     STRING STR;
145     IF (!RTC.ISDATETIMEVALID())
146     {
147         // CAUSAS COMUNS: BATERIA BAIXA OU DESCONECTADA
148         SERIAL.PRINTLN("RTC PERDEU CONFIANÇA NA DATA E HORA!");
149     }
150     RTCDATETIME NOW = RTC.GETDATETIME();
151     STR = FORMATDATETIME(NOW);
152     RETURN STR;
153 }
154
155 #DEFINE COUNTOF(A) (sizeof(A) / sizeof(A[0]))
156 STRING FORMATDATETIME(CONST RTCDATETIME& DT)
157 {
158     CHAR DATESTRING[20];
159     SNPRINTF_P(DATESTRING,
160               COUNTOF(DATESTRING),
161               PSTR("%02U/%02U/%04U %02U:%02U:%02U"),
162               DT.MONTH(),
163               DT.DAY(),

```

```

164         DT.YEAR() ,
165         DT.HOUR() ,
166         DT.MINUTE() ,
167         DT.SECOND() );
168     STRING STR(DATESTRING);
169     RETURN STR;
170 }
171
172 //-----
173 // FUNÇÃO DE LEITURA DE TEMPERATURA
174 //-----
175 STRING LEITURATEMP() {
176     RTCTEMPERATURE TEMP = RTC.GETTEMPERATURE();
177     FLOAT DEG = TEMP.ASFLOATDEGC();
178     STRING STR(DEG);
179     RETURN STR;
180 }
181 //-----
182 // FUNÇÃO DE CHECA MES
183 //-----
184 VOID CHECAMES() {
185     UINT8_T MESTEMP;
186     RTCDATETIME NOW = RTC.GETDATETIME();
187     MESTEMP = NOW.MONTH();
188
189     IF(MESTEMP != MES)
190     {
191         //MUDOU O MÃAS
192         //SALVAR CONSUMO ANT
193         //MES AT = DADOSWIFI - CONSUMO ANT
194         CONSUMO_MESANT = VOLUME;
195     }
196     MES = MESTEMP;
197
198     //SERIAL.PRINTLN(MES);
199     //SERIAL.PRINTLN(LEITURARTC());
200     //SERIAL.PRINTLN(CONSUMO_MESAT);
201     //SERIAL.PRINTLN(VOLUME);
202     //LCD.PRINT(CONSUMO_MESAT);
203     //LCD.PRINT(VOLUME);
204
205 }

```