



**UNIVERSIDADE FEDERAL DE CAMPINA GRANDE
CENTRO DE ENGENHARIA ELÉTRICA E INFORMÁTICA
DEPARTAMENTO DE ENGENHARIA ELÉTRICA**

Desenvolvimento de um Wattímetro Trifásico Microprocessado

VICTOR HUGO DOS SANTOS GERMANO

6 de agosto de 2019

Campina Grande - PB

VICTOR HUGO DOS SANTOS GERMANO

Desenvolvimento de um Wattímetro Trifásico Microprocessado

Trabalho de Conclusão de Curso apresentado à Coordenação do Curso de Graduação em Engenharia Elétrica da Universidade Federal de Campina Grande como parte dos requisitos necessários para a obtenção do grau de Bacharel em Ciências no Domínio da Engenharia Elétrica.

Área de Concentração: Eletrotécnica

Edson Guedes da Costa

Orientador

6 de agosto de 2019

Campina Grande - PB

Desenvolvimento de um Wattímetro Trifásico Microprocessado

Trabalho de Conclusão de Curso apresentado à Coordenação do Curso de Graduação em Engenharia Elétrica da Universidade Federal de Campina Grande como parte dos requisitos necessários para a obtenção do grau de Bacharel em Ciências no Domínio da Engenharia Elétrica.

Área específica de formação: Eletrotécnica

Aprovado em: / /

Edson Guedes da Costa

Universidade Federal de Campina Grande

Orientador

Pablo Bezerra Vilar

Universidade Federal de Campina Grande

Avaliador

Dedico este trabalho a minha avó, Maria da Paz Brandão, e minha bisavó, Audália Maria Marques, que muito fizeram para que eu pudesse estar aqui. Saudades eternas.

AGRADECIMENTOS

Primeiramente queria agradecer àquelas pessoas que considero a base de tudo pra mim: minha bisavó Audália, meus pais Augusto e Neuma e meus avós que muito lutaram para que pudesse estar aqui, tendo mais essa conquista que lhes dá muito orgulho e felicidade. Também grato aos demais familiares, entre tios, tias e primos que me incentivaram a ver a carreira que escolhi como uma decisão certa.

Aos amigos que estão na minha terra natal, Juazeiro do Norte - CE, cuja a distância e o tempo não foram empecilhos para nossa relação consolidada e honesta em várias formas. E aos amigos que fiz em Campina Grande - PB, cidade que me abraçou com muito carinho nesses 5 anos, por terem me dado conforto e apoio nas horas mais difíceis, seja com momentos de lazer, uma sugestão ou um abraço, pretendo guardar tudo isso comigo.

Aos irmãos de luta que fiz: João Pedro, Kaio Nikelisson, Adeilmo Júnior, Ravi, Robson Donato, João Victor Jales, e Ulisses Gomes. Pessoas com quem me diverti muito nas horas mais boêmias.

Aos colegas e professores da UFCG, funcionários do LAT e do Departamento de Engenharia Elétrica que contribuíram para que eu alcançasse os objetivos para o desenvolvimento do projeto. Destaco a contribuição de Adeilmo Júnior, João Vitor Jales, Lenilson Barbosa e Edson Guedes.

“Milhares de velas podem ser acesas de uma única vela, e a vida da vela não será encurtada. Felicidade nunca diminui ao ser compartilhada.”.

Siddhartha Gautama.

RESUMO

A Eletrotécnica prossegue com o desenvolvimento tecnológico voltando-se fortemente em projetos com dispositivos de instrumentação para medição e diagnósticos dos equipamentos que intervenham no sistema elétrico de potência. Este documento descreve as etapas da construção de um Wattímetro Trifásico Microprocessado, citando, quando necessário, os componentes aplicados e suas funcionalidades na elaboração do equipamento previsto para uso na disciplina Laboratório de Equipamentos Elétricos, oferecidos pelo Departamento de Engenharia Elétrica da UFCG.

Palavras-chave: Wattímetro, M90E36A, ESP8266, Engenharia Elétrica.

ABSTRACT

The Electrical Engineering continues with the technological development directing itself strongly in projects with devices of instrumentation for the measurement and diagnostics of the equipment that intervene in the electric power system. This document describes the steps of the construction of a Microprocessed Three-Phase Wattmeter, mentioning when it's necessary the components applied and their functionalities in the elaboration of the equipment provided for use in the discipline Electrical Equipment Laboratory, offered by the Electrical Engineering Department of UFCG.

Keywords: Wattmeter, M90E36A, ESP8266, Electrical Engineering.

Lista de Abreviaturas e Siglas

ADC *Analog-to-Digital Converter* - Conversor Analógico-Digital. 22, 23, 28, 30

CI Circuito Integrado. 22, 31

DMA *Direct Memory Access* - Acesso Direto à Memória. 23

DSP *Digital Signal Processor* - Processador de Sinais Digitais. 22

f.p fator de potência. 17, 18

GPIO *General Purpose Input/Output* - Portas Programáveis de Entrada e Saída. 21, 26

I Corrente Elétrica. 18

I²C *Inter-Integrated Circuit*. 21

IDE *Integrated Development Environment* - Ambiente de Desenvolvimento Integrado. 22

IEEE Instituto de Engenheiros Eletricistas e Eletrônicos. 18

IoT *Internet of Things*. 16

LAT Laboratório de Alta Tensão. 16, 17

LCD *Liquid Crystal Display* - *Display* de Cristal Líquido. 25, 26

LSB *Less Significant Bit* - Bit Menos Significativo. 31

MSB *Most Significant Bit* - Bit Mais Significativo. 24, 31

OLED *Organic Light-Emitting Diode* - Diodo Orgânico Emissor de Luz. 27

P potência ativa. 17, 18, 33

PWM *Pulse Width Modulation* - Modulação por Largura de Pulso. 21

Q potência reativa. 17, 33

rms *Root Mean Square* - Valor Eficaz. 18

S potência aparente. 18

SD *Secure Digital Card* - Cartão de Memória SD. 25, 26

SPI *Serial Peripheral Interface.* 21, 23, 24, 26

SPIFFS *SPI Flash File System.* 26

TC Transformador de Corrente. 20, 21

UFCG Universidade Federal de Campina Grande. 1, 15, 16, 25

V Tensão. 18

Lista de Figuras

1	Esquema elétrico para o ensaio de curto-circuito no transformador.	19
2	Esquema elétrico para o ensaio em circuito aberto no transformador.	20
3	Sensor de corrente não-invasivo.	21
4	NodeMCU como módulo para o ESP8266.	22
5	Chip Atmel M90E36A	22
6	Fotografia do módulo para o M90E36A.	24
7	Foto da primeira montagem para o wattímetro projetado.	26
8	Foto da segunda montagem para o wattímetro com as adequações do <i>hardware</i>	27
9	Página <i>web</i> principal desenvolvida para o wattímetro.	29
10	Páginas secundárias desenvolvidas para o wattímetro.	29
11	Projeto para o circuito de atenuação para suporte dos ADCs.	31
12	Montagem da bancada para os primeiros testes de calibragem do wattímetro.	34
13	Esquemático simplificado das conexões para a bateria de testes do wattímetro.	35
14	Primeiros resultados para o Wattímetro com ajustes nos ganhos de tensão e corrente. . .	36

Lista de Tabelas

1	Adaptações das funções do M90E36A.	25
2	Funções do ESP8266 para registro e uso dos dados na página <i>web</i>	27
3	Especificações elétricas do M90E36A em seu <i>datasheet</i>	30
4	Resultados com ajuste no ganho para tensão eficaz da fase C.	36
5	Resultados com ajuste no ganho para corrente eficaz da fase C.	37
6	Resultados com ajuste no ganho para corrente eficaz da fase C.	38

Sumário

Lista de Abreviaturas e Siglas	9
1 INTRODUÇÃO	15
1.1 OBJETIVOS	15
1.2 MOTIVAÇÃO	16
1.3 ESTRUTURA DO TRABALHO	16
2 FUNDAMENTAÇÃO TEÓRICA	17
2.1 WATTÍMETRO	17
2.2 CÁLCULO DA POTÊNCIA ELÉTRICA	18
2.3 ENSAIO EM TRANSFORMADORES	19
2.3.1 Ensaio de Curto-Circuito	19
2.3.2 Ensaio em Circuito Aberto	20
2.4 TRANSFORMADORES DE CORRENTE E O SCT-13	20
2.5 ESP8266 NodeMCU ESP-12E	21
2.6 ATMEL M90E36A	22
3 DESENVOLVIMENTO DO WATTÍMETRO	24
3.1 MÓDULO PARA O M90E36A	24
3.2 COMUNICAÇÃO E COMPATIBILIDADE DOS DISPOSITIVOS PERIFÉRICOS	24
3.3 PÁGINA <i>WEB</i>	28
3.4 ETAPAS DE CALIBRAGEM DO WATTÍMETRO	30
3.4.1 Circuitos Atenuadores	30
3.4.2 Registradores de Configuração	31
3.4.3 Calibragem dos Ganhos de Corrente e Tensão Eficazes	32
3.4.4 Registradores de Ajuste do Ângulo de Fase	32
4 CALIBRAGEM DO WATTÍMETRO	34
5 CONCLUSÃO	39
6 ATIVIDADES FUTURAS	40
6.1 PLACA DE CIRCUITO IMPRESSO	40
6.2 OUTRAS ETAPAS	40
6.3 PESQUISAS FUTURAS: CONTINUAÇÃO DA CALIBRAGEM DO WATTÍMETRO	40
7 REFERÊNCIAS	41

A APÊNDICE	42
A.1 ESQUEMA DO CIRCUITO DO WATTÍMETRO	42
A.2 CÓDIGOS IMPLEMENTADOS E ADAPTADOS	43

1 INTRODUÇÃO

No âmbito da eletrotécnica, é debatido a necessidade de o corpo docente ser levado a práticas que além de reforçar o aprendizado sobre as teorias que fundamentam as características e eventos do sistema elétrico de potência, que também contextualize os ensaios estabelecidos pelas normas técnicas nacionais para equipamentos elétricos[1].

Tomando a prática como algo imprescindível, a disciplina Laboratório de Equipamentos Elétricos elabora uma diversidade de experimentos. A título de exemplo, pode-se destacar os ensaios de tipo em transformadores de potência.

Este experimento em específico qualifica um perfil magnetelétrico ao transformador. Submetido a condições de circuito aberto e de curto-circuito, os resultados do levantamento fornecem dados que influem para o cálculo de perdas em tais equipamentos; portanto, torna-se necessário um dispositivo para medição de potência elétrica consumida ou fornecida por um equipamento elétrico ou circuito: o wattímetro. A instrumentação ganha destaque no desenvolvimento do wattímetro, termo chave para a proposta.

A concepção avoca consigo boa parte do conhecimento adquirido em projetos anteriormente elaborados por alunos do curso de Engenharia Elétrica na Universidade Federal de Campina Grande (UFCG) que estejam relacionados ao tema; mas traz novas características que justificam prosseguir com o desenvolvimento deste equipamento. É digno de destaque algumas novidades como a capacidade do wattímetro operar nas condições de instrumento trifásico e monofásico; e graças às funcionalidades do *hardware* escolhido, obtenção de outras grandezas como fator de potência e magnitude das componentes harmônicas de tensão e corrente; além de sua comunicação extrapolada para outros meios de visualização e registro dos dados.

1.1 OBJETIVOS

O objetivo do trabalho é o desenvolvimento de um wattímetro digital trifásico para medição da potência elétrica consumida por um elemento ou circuito elétrico que particularmente resulte em níveis baixos de potência, categorizado para operar com até 380 volts entre fases e detectabilidade de corrente na faixa de 0,01 a 5 ampère, desenvolvida inicialmente para uma classe B de exatidão.

A sequência dos objetivos específicos que se tinha inicialmente segue uma ordem cronológica em alcançá-los:

- Projetar o wattímetro digital trifásico (funcionalidades básicas) e montar o protótipo;
- Analisar e desenvolver a extensão de funcionalidade do wattímetro e sua interação com o usuário;
- Montar definitivamente o wattímetro (funcionalidades estendidas);

- Testar e validar o protótipo por testes comparativos de operação com os resultados obtidos por um osciloscópio e/ou qualímetro;
- Produzir um manual simplificado didático para uso do wattímetro.

1.2 MOTIVAÇÃO

O projeto encontrou como motivação, o desenvolvimento de competências nas áreas da eletrônica analógica, aquisição e processamento de dados, microcontroladores e afins, além do desenvolvimento de placas de circuito impresso.

Neste sentido, a avaliação crítica, a problematização da realidade, e a capacidade de engenhar soluções criativas contribuirão para o desenvolvimento do projeto, além de ampliar e sedimentar os conhecimentos alcançados no decorrer da graduação.

Dada outras execuções objetivando a construção de um wattímetro, observou-se alguns aspectos que impulsionaram traçar novos rumos: redução ao máximo do número de componentes do equipamento de medição, trazendo dispositivos que venham a garantir todas as funcionalidades propostas para o wattímetro; a capacidade do equipamento em fornecer os parâmetros para as três fases; e por fim, expor tecnologias que trazem concepções mais modernas de instrumentação, especificamente à interconexão digital (IoT).

Este projeto também visa o suprimento do Laboratório de Alta Tensão (LAT) da UFCG, uma vez que o protótipo desenvolvido poderá ser utilizado no local para pesquisas ou ensaios com equipamentos de baixo nível de potência, inclusive servindo ao laboratório da disciplina Equipamentos Elétricos, do Departamento de Engenharia Elétrica, onde são estudados os transformadores de potência[1].

1.3 ESTRUTURA DO TRABALHO

Este trabalho de conclusão de curso está dividido em cinco capítulos. O primeiro capítulo destina-se à parte introdutória. No segundo, tem-se uma fundamentação teórica a respeito do funcionamento básico de um wattímetro, do cálculo de potência, dos ensaios de curto circuito e circuito aberto em transformadores e dos componentes principais que integram o equipamento de instrumentação proposto. O capítulo 3 destina-se aos procedimentos empregados no desenvolvimento do protótipo até o estágio em que o projeto se encontra. No capítulo 4 é descrito a calibragem realizada para o wattímetro. No capítulo 5 é descrito as conclusões para cada objetivo traçado em projeto. Para os capítulos 6 e 7 são listadas as atividades e pesquisas futuras.

2 FUNDAMENTAÇÃO TEÓRICA

Nesta fundamentação teórica relata-se de forma generalista concepções enquadradas em áreas de concentração bem distintas do curso de Engenharia Elétrica, que trazem a fundamentação sobre o que é o wattímetro, os tipos de ensaios empregados no LAT; as informações sobre a eletrônica empregada; e uma abordagem mais atualizada para as ferramentas que um equipamento de instrumentação pode conter.

2.1 WATTÍMETRO

A instrumentação refere-se a qualquer dispositivo ou conjunto destes utilizado para medir, indicar registrar ou controlar variáveis de um processo. Para o objeto a ser desenvolvido no projeto, classifica-se o wattímetro como um instrumento de medição da potência elétrica que esteja sendo dissipada ou fornecida por um elemento de circuito (carga)[2].

Como um princípio para dispositivos de medição, o wattímetro deve realizar uma leitura que condiz com a tolerância projetada para representação de uma grandeza, e sua inserção em qualquer circuito elétrico tenha interferência mínima sobre o sistema a ser mensurado.

Construtivamente, o wattímetro pode ser classificado como eletrodinâmico ou eletrônico. O primeiro é composto por um enrolamento de corrente e um enrolamento de tensão. A bobina (ou enrolamento) de corrente deve apresentar baixa impedância (idealmente nula) estando conectado em série à carga, enquanto a bobina de tensão, com impedância extremamente alta (idealmente infinita) estará conectado em paralelo à carga. A inércia mecânica do sistema móvel produz um ângulo de deflexão proporcional ao valor médio do produto da tensão vezes a corrente elétrica[1][3].

Já o wattímetro eletrônico é constituído por transdutores, circuitos de condicionamento. No caso do wattímetro digital, adiciona-se ainda um aparato de microcomputadores e/ou microprocessadores. A aquisição dos valores temporais de tensão e corrente elétrica em certa frequência de amostragem, mais a capacidade de cálculos com tais parâmetros, seja daqueles inerentes ao *hardware*, ou desenvolvido em *software*, permite obter resultados além do simples cálculo da potência ativa (P), como por exemplo, cálculo da potência reativa (Q) e o fator de potência (f.p). Outra característica wattímetro eletrônico é que o equipamento não apresenta desgaste das partes móveis, além de estar mais adequado a uma leitura direta do que, por exemplo, uma indicação por um ponteiro em uma escala graduada. Estes problemas permitem erros em maior grau em relação à calibragem e à leitura equivocada do observador.

A disparidade do preço de um wattímetro eletrônico em relação a um eletrodinâmico vem diminuindo ano após ano, justificando o desenvolvimento de equipamentos eletrônicos com maior complexidade de operação para uso mais triviais, incluindo o uso disciplinar.

2.2 CÁLCULO DA POTÊNCIA ELÉTRICA

O conceito mais fundamental trata sobre a medição da potência aparente (S) para elementos de circuito bipolares (dois terminais), que consiste em multiplicar as grandezas tensão elétrica eficaz (sob os terminais) V_{rms} e a intensidade corrente elétrica eficaz I_{rms} que flui pelo elemento, conforme pode ser visto na Equação 1:

$$S = V_{rms} \cdot I_{rms}. \quad (1)$$

No domínio discreto, é realizado a medição pontual para o valor da potência instantânea com os valores pontuais de tensão e corrente, como está descrito na Equação 2:

$$p[n] = v[n] \cdot i[n]. \quad (2)$$

A potência ativa (P) é conceitualmente a potência média de um sinal. Ao ser considerado um sinal discreto e periódico com período N , seu valor pode ser calculado utilizando a Equação 3:

$$P = \frac{1}{N} \sum_{n=0}^{N-1} v[n] \cdot i[n]. \quad (3)$$

A metodologia para encontra a potência aparente (S) se baseia nos valores pontuais das amostras de tensão e corrente. Uma vez obtidos S e P , é possível relacioná-los ao fator de potência ($f.p$) como é mostrado na Equação 4:

$$f.p = \frac{P}{S}. \quad (4)$$

Para sinais sinusoidais em regime permanente, o fator de potência está relacionado à função cosseno, cujo argumento φ_{vi} fornece o desfasamento elétrico do sinal de corrente com relação ao sinal da tensão elétrica considerados na mensuração, conforme visto na Equação 5:

$$\cos(\varphi_{vi}) = \frac{P}{S}. \quad (5)$$

Condicionado a sistemas trifásicos, a potência ativa trifásica $P_{3\phi}$ é simplesmente o resultado da soma aritmética das potências ativas em cada uma das fases. O cálculo para a potência aparente trifásica $S_{3\phi}$ faz uso do método elaborado pelo IEEE para cálculos dos padrões de medições relacionados à potência elétrica[4].

2.3 ENSAIO EM TRANSFORMADORES

Os ensaios básicos de rotina que fazem uso de wattímetros para a caracterização de transformadores são descritos conforme estabelece a norma NBR5380, a qual deve ser consultada para maiores detalhes. É esperado que o conteúdo esteja em acordo ao guia de experimentos do Laboratório de Equipamentos Elétricos[1][5].

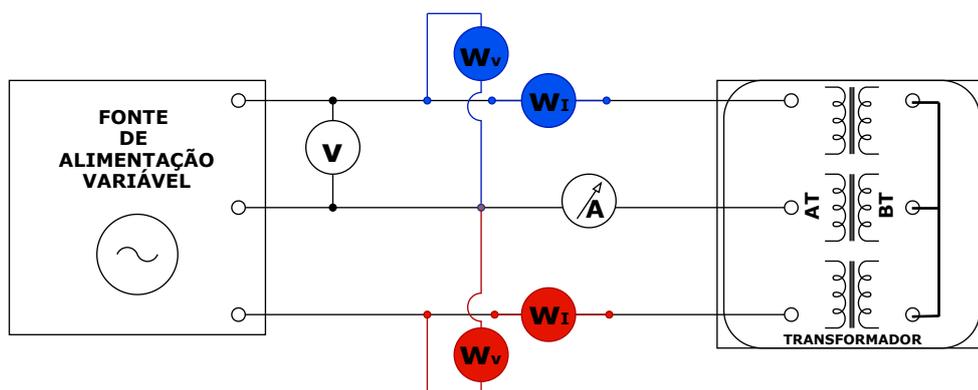
2.3.1 Ensaio de Curto-Circuito

O ensaio de curto-circuito possibilita a determinação das perdas no cobre (P_J), da queda de tensão interna e dos valores de impedância, resistência e reatância dos enrolamentos do equipamento magnetelétrico. Perdas no cobre provém do efeito Joule nos enrolamentos do transformador que apresenta uma resistência interna ao fluxo de correntes.

As correntes nominais do transformador são estabelecidas, exigindo cuidado na prática deste ensaio devido aos níveis consideráveis de corrente elétrica que flui no enrolamento secundário. Condiicionado ao lado de baixa tensão curto-circuitado, o primário com a tensão entre fases reduzida a 10% do seu valor nominal é suficiente para atingir os níveis nominais de corrente elétrica. A tensão reduzida e o contato elétrico acarreta em uma menor indução do núcleo, tornando desprezível o efeito de histerese e de correntes de Foucault na avaliação das perdas neste ensaio.[1]

No arranjo montado no Laboratório de Equipamentos Elétricos são utilizados dois wattímetros dispostos em uma configuração característica para medição da potência em circuitos trifásicos (método dos dois wattímetros). Também são utilizados um amperímetro, para medição da corrente de excitação na da fase de referência, e um voltímetro, conforme o diagrama de montagem apresentado na Figura 1.

Figura 1: Esquema elétrico para o ensaio de curto-circuito no transformador.



Fonte: Guia de experimentos do Laboratório de Equipamentos Elétricos.

2.3.2 Ensaio em Circuito Aberto

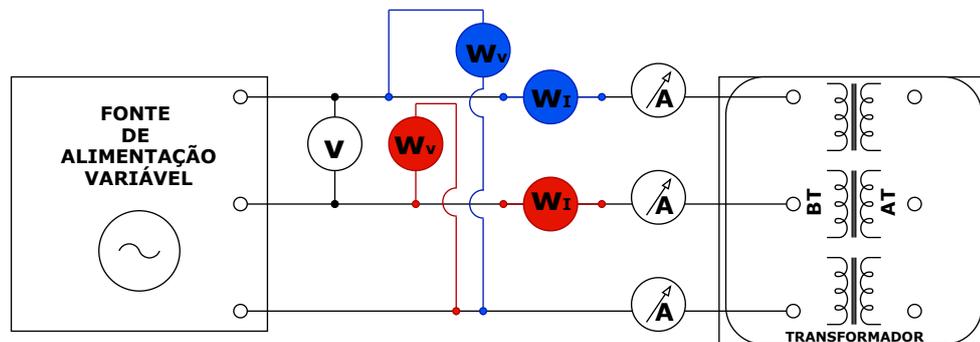
O ensaio em circuito aberto permite a determinação das perdas do transformador ao operar em vazio. Estima-se, portanto, a energia necessária para excitação do circuito magnético, considerando as perdas relacionadas à histerese e às correntes de Foucault[5].

Além disso o método permite a estimação da corrente em vazio (atrelado ao nível de saturação do núcleo magnético) e a relação de transformação do equipamento.

O ensaio de circuito aberto é realizado, portanto, aplicando tensão nominal senoidal nos enrolamentos de baixa tensão enquanto os terminais do lado de alta tensão ficam em aberto, o que exige cuidado de operação, se tratando de níveis de tensão com ordem de grandeza em kV .

O esquemático do Laboratório de Equipamento Elétricos mostra que são utilizados, além de dois wattímetros para medição das potências, três amperímetros para medição das correntes de excitação em cada uma das fases, e um voltímetro, conforme diagrama de montagem apresentado na Figura 2.

Figura 2: Esquema elétrico para o ensaio em circuito aberto no transformador.



Fonte: Guia de experimentos do Laboratório de Equipamentos Elétricos.

2.4 TRANSFORMADORES DE CORRENTE E O SCT-13

Transformadores de corrente (TCs) são equipamentos projetados e construídos especificamente para alimentar instrumentos de medição, proteção ou controle em sistemas elétricos. Sua finalidade constitui-se em reduzir a corrente do sistema de potência a um valor adequado aos instrumentos, de modo a haver uma correspondência constante entre os valores instantâneos referentes aos sinais de corrente de saída e de entrada, assim como uma defasagem mínima entre eles. A aplicação do dispositivo promove o isolamento elétrico entre os instrumentos e o sistema de potência[6].

O Sensor de corrente STC-13 mostrado na Figura 3 corresponde a uma família de Transformador de Corrente (TC)s que serão utilizados no projeto. Um benefício ao utilizá-lo é que seu método de transformação não é invasivo, conferindo-lhe a condição de isolamento elétrico do sistema de potência medido, embora apresente limitações para a condição máxima de operação (valores próximo de 5A). Muito usado em projetos com embarcados na automação residencial, como medidores de corrente elétrica, proteção de motores de corrente alternada, iluminação e entre outras funcionalidades[7].

Figura 3: Sensor de corrente não-invasivo.



Fonte: *Datasheet* do STC-13.

Para o transformador de corrente, o secundário é associado a uma carga de impedância conhecida, já que a detecção do sinal para operação digital se dá por um nível de tensão. Mas além disso, é preciso avaliar a influência da carga em seu secundário.

Ao projetar a introdução da impedância (geralmente um resistor), para ser obtido uma relação entre a intensidade da corrente de entrada e o nível tensão sob os terminais do secundário do TC, é necessário se atentar que quanto maior for valor de impedância no secundário, mais próximo o ponto de operação se aproxima da região de saturação do TC, já que o sensor se trata de um equipamento magnético. Em geral, o projeto além de considerar a condição de saturação, também avalia o nível de tensão às condições suportáveis para o circuito que será associado ao TC.

Para o caso do TC em uso (STC-13 005), sua relação de transformação está associada a um nível de tensão ($5A/1V$), o que sugere uma impedância já definida e inserida no secundário do sensor. A confirmação ocorre ao se consultar a descrição do fabricante.

2.5 ESP8266 NodeMCU ESP-12E

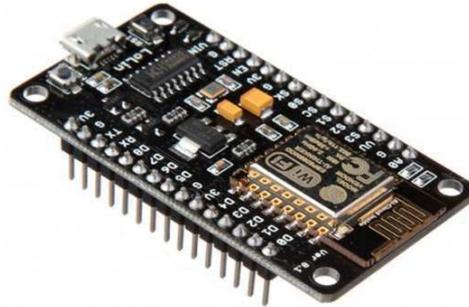
O ESP8266 é um chip de fabricação chinesa que se disseminou rapidamente ao entrar no mercado com baixo custo e funcionalidades bastante úteis para aplicação em engenharia, como a sua capacidade de comunicação por *Wi-Fi*[8].

A boa relação custo-benefício do chip impulsionou o projeto e a produção de módulos e placas de desenvolvimento, como o ESP8266 NodeMCU ESP-12E, que pode ser visto na figura 4. Além do chip, são adicionados à plataforma dispositivos que permitem maior praticidade de operação. Ele é um microcontrolador com arquitetura de 32 bits desenvolvido pela Espressif que possui especificações como:

- Wireless padrão 802.11 b/g/n;
- Antena embutida;
- Conector micro-usb;
- 11 Portas GPIO com funções de PWM e suporte às plataformas de comunicação I²C e SPI;

- Tensão de operação: 4, 5 - 9V;
- Taxa de transferência: 110 – 460800bps;
- Conversor analógico-digital (ADC);

Figura 4: NodeMCU como módulo para o ESP8266.



Fonte: ESP8266EX Datasheet - Espressif Systems.

A velocidade de operação do NodeMCU ESP-12E atinge até 80MHz. Sua memória para cache melhora seu desempenho ao ser aplicado em sistemas ou circuitos, e reduz a necessidade do módulo do ESP8266 de apresentar uma grande quantidade de memória. Já a *interface* usb-serial, citada nas especificações do módulo, permite sua programação usando LUA ou a própria IDE do Arduino.

2.6 ATMEL M90E36A

O M90E36A é um circuito integrado desenvolvido pela Atmel para medição com ampla faixa de operação, alto desempenho, e atuando sobre múltiplas fases. Fisicamente, apresenta encapsulamento de baixo custo do tipo TQFP48, como pode ser visto na Figura 5[9]:

Figura 5: Chip Atmel M90E36A



Fonte: Microchip Technology.

O CI incorpora 7 conversores analógicos-digitais de 24 bits, independentes e de 2ª ordem, que podem ser utilizados em três canais de tensão (fase A, B e C) e quatro canais de corrente (fase A, B, C e neutro) em um típico sistema trifásico de quatro fios.

Embutido no CI, o processador digital de sinais (DSP) executa cálculos de potência ativa (fundamental e harmônica), reativa e aparente sobre o sinal convertido no ADC e sua tensão de referência no chip. O DSP também permite o cálculo de parâmetros de medição, como o valor eficaz da tensão e da corrente, frequência, análise Fourier para tensão e corrente, e distorção harmônica total.

Uma interface SPI de quatro fios é fornecida entre o M90E36A e o microcontrolador externo. Além disso, é possível pelo mesmo canal ter acesso direto à memória (DMA) aos dados brutos dos 7 canais do ADC, o que oferece mais flexibilidade na aplicação do sistema. As dezenas de registradores com tamanho fixo de 16 bits presentes em sua memória, compõem a capacidade física de configuração de sua operação, leitura e escrita de dados durante a comunicação com o microcontrolador. Mais detalhes sobre o M90E36A são encontrados em seu *datasheet*.

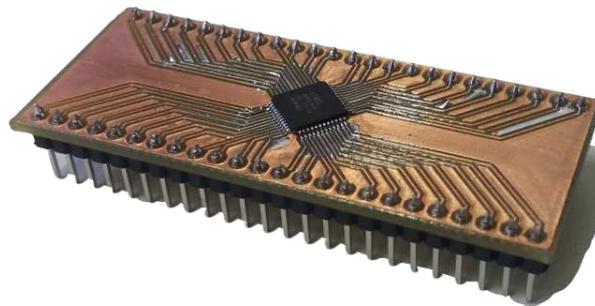
3 DESENVOLVIMENTO DO WATTÍMETRO

3.1 MÓDULO PARA O M90E36A

O chip originalmente adquirido não estava apto a ser conectado aos demais dispositivos, dado seu formato de encapsulamento. Necessitava, portanto, construir um módulo de acesso. A etapa consistiu em montar o *layout* no ambiente da ferramenta, e concebê-lo numa impressora específica à manufatura de placas para circuitos elétricos. O resultado é mostrado na Figura 6:

O desenvolvimento em separado do módulo para o M90E36A, em relação à placa que seria desenvolvida para o wattímetro, foi decidido pensando no chip como recurso substituível, pois diante de sua sensibilidade, tanto na sua introdução durante a confecção da placa, quanto para uso no laboratório em seu estágio pronto, tornaria dispendioso o descarte de toda uma placa em caso de falha do circuito integrado.

Figura 6: Fotografia do módulo para o M90E36A.



Fonte: Próprio autor.

3.2 COMUNICAÇÃO E COMPATIBILIDADE DOS DISPOSITIVOS PERIFÉRICOS

Esta etapa descreve as escolhas dos dispositivos eletrônicos que foram associados entre si e operam conjuntamente para o funcionamento do wattímetro.

A conexão entre dispositivos é influenciado diretamente pelo tipo de comunicação optado. Para a comunicação entre os principais dispositivos: o módulo do ESP8266 e o chip M90E36A, foi utilizado o protocolo SPI, que faz 4 conexões, nomeados SDI, SDO, SCLK e CS. A razão por usar o protocolo SPI é que o chip M90E36A apresenta esse único tipo de comunicação, impondo o ESP8266 a também utilizá-la. Quanto à conexão para alimentação do M90E36A, foi utilizado as saídas do ESP8266 que apresentam 3,3V entre si, que é o nível adequado de tensão para o funcionamento do chip.

Enquanto o protótipo do wattímetro continha apenas o ESP8266 e o M90E36A, o desenvolvimento da comunicação prosseguiu sem problemas. As funcionalidades de leitura e escrita de dados, feitas para um conjunto de 16 bits e iniciando sempre pelo bit mais significativo (MSB), atuavam corretamente, devido às garantidas impostas pelo processo de validação da comunicação.

O processo de validação consiste na gravação de parte dos dados que serão escritos em um

registrador do M90E36A. Este registrador é então comparado a um resultado que contém parte dos dados presentes nos registradores que foram alterados no processo de escrita.

É exibido na Tabela 1 algumas funcionalidades para a comunicação entre o M90E36A e o módulo ESP8266. O desenvolvimento das funções está presente no apêndice A.2 deste documento. Algumas delas já tinham sido desenvolvidas pelo próprio corpo docente da UFCG[10], sendo feitas as adequações necessárias para compatibilidade aos dispositivos que compõem o wattímetro.

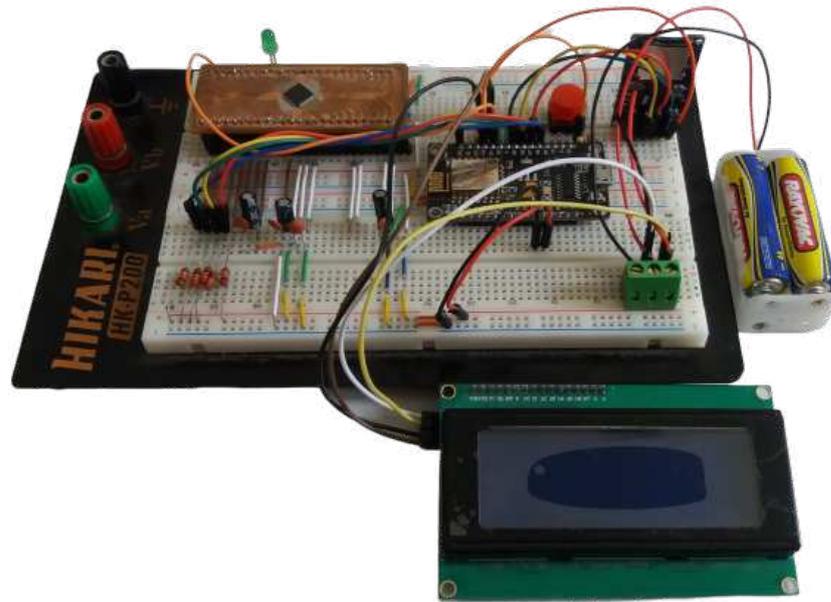
Tabela 1: Adaptações das funções do M90E36A.

Função	Descrição
pmicSetup()	Inicializa o M90E36A.
write16()	Escreve 16 bits em um registrador.
read16()	Lê 16 bits de um registrador.
csCalculator()	Calcula o CS.
realRms()	Retorna os valores RMS.
activePower()	Retorna a potência ativa.
reactivePower()	Retorna a potência reativa.
apparentPower()	Retorna a potência aparente.
powerFactor()	Retorna fator de potência.
checksumStatus()	Retorna status do Checksum.
measurementCalibration()	Inicia ou finaliza a calibragem de aquisição.
energyCalibration()	Inicia ou finaliza a calibragem de cálculos.
measurementGainCalibration()	Calibra os ganhos das medições.

Fonte: Plateny, 2013

Seguiu o desenvolvimento adicionando outros dispositivos ao protótipo como mostrado na Figura 7. Foram adicionados o módulo para leitura de cartões SD, servindo de memória para arquivos que seriam úteis na criação da página web; e o LCD 20x4, para mostrar a leitura direta dos parâmetros calculados pelo M90E36A.

Figura 7: Foto da primeira montagem para o wattímetro projetado.



Fonte: Próprio autor.

Na etapa em *hardware* foi incrementado, o protótipo do wattímetro não respondia corretamente às funções de comunicação já desenvolvidas antes, tão pouco se conseguia obter resultados satisfatórios daquelas funcionalidades elaboradas para os dispositivos recém-adicionados.

Procurando corrigir os erros que surgiram, foram diagnosticado dois fatores críticos à continuação do projeto. Primeiramente, os pinos genéricos de entrada e saída (GPIOs) do ESP8266 apresentavam limitações em seu uso; em segundo, havia conflito na comunicação do microcontrolador ESP aos seus periféricos: o módulo para suporte SD se comunicava por protocolo SPI, e teria que compartilhar 3 conexões com o M90E36A, evidenciado na foto da montagem na Figura 7.

Não obstante, os *datasheets* do módulo SD e do LCD descreviam que era necessário para o funcionamento dos dois dispositivos uma alimentação ao nível de tensão contínua de 5V. Para este tipo de alimentação, o ESP8266 não apresentava suporte, recorrendo-se então ao uso de uma alimentação externa, como também pode ser visto na Figura 7. A adição desta fonte, porém, afetou o processo construtivo do wattímetro.

Na busca por solucionar os problemas e tentar dar continuidade ao projeto, foram encontradas alternativas que sugeriam substituir ou retirar os dispositivos periféricos problemáticos e ao mesmo tempo evitar o uso dos GPIOs do ESP8266 que não respondiam bem ao que foram inicialmente designados.

O primeiro ato foi remover o módulo SD, cuja funcionalidade de gravar os dados foi assumida pelo uso da memória *flash* do módulo do ESP8266 e seu sistema para armazenamento de dados e arquivos internamente (SPIFFS). Como se tratava de um recurso que ocupava um espaço mínimo na memória do ESP8266; e ainda, pela familiaridade do projetista ao uso da ferramenta SPIFFS, a mudança tornou-se muito adequada. As funções desenvolvidas exclusivamente para uso do recurso SPIFFS são mostradas na Tabela 2:

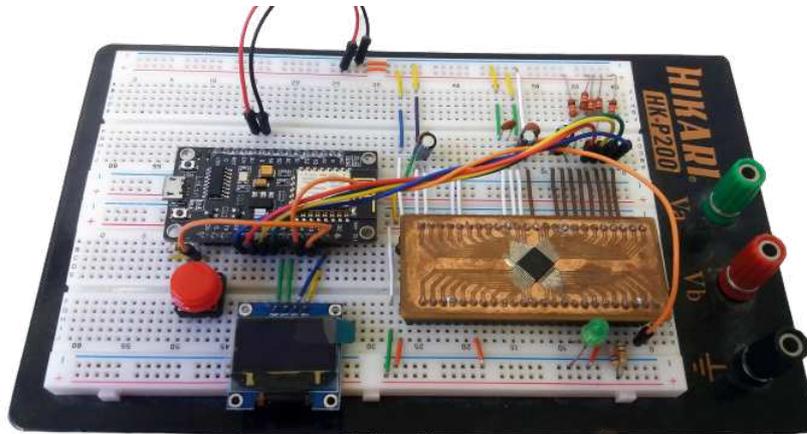
Tabela 2: Funções do ESP8266 para registro e uso dos dados na página *web*.

Função	Descrição
outputDMA()	Acesso direto à memória e registro dos dados.
fileWrite()	Escrita dos dados de potência para as 3 fases.

Fonte: Próprio autor.

Para solucionar os problemas com o LCD, optou-se por sua substituição pelo *display* OLED, como mostrado na Figura 8. Apesar do OLED conter um *display* gráfico de apenas 0,96 polegadas, a tela exibe informações com muita nitidez, devido ao seu alto contraste. A alimentação para o OLED na faixa de tensão em 3,3V, solicitando pouco consumo energético, se adequou ao suporte que o microcontrolador ESP8266 pode fornecer.

Figura 8: Foto da segunda montagem para o wattímetro com as adequações do *hardware*.



Fonte: Próprio autor.

Com a substituição do LCD pelo OLED e a retirada do módulo SD, resultando na montagem mostrada na Figura 8 e no apêndice A.1.

O protótipo voltou a responder corretamente, dando prosseguimento ao projeto.

3.3 PÁGINA WEB

Uma atenção particular foi dada ao desenvolvimento da página *web*. Primeiramente, ao dever de inovação para equipamentos de instrumentação, seguindo uma tendência mercadológica de produtos do tipo. E em segundo, fora pensado na criação de uma plataforma que permitisse uma análise das próprias medições, que podem ser utilizados inclusive na etapa futura de calibração do equipamento.

Esta etapa tinha como precursora para base de seu processo o projeto de um qualímetro desenvolvido pelo Setor de Qualidade do Laboratório de Alta Tensão.

A página foi desenvolvida em linguagem HTML, com seu *layout* configurada pelo mecanismo CSS. Ambas foram escolhidas por sua versatilidade e familiaridade de seu uso na engenharia.

Originalmente, além do uso da linguagem citada, que fazia uso de processamento do cliente (*Browser*) para visualização da página, também se desenvolveu em linguagem PHP o método de leitura e elaboração dos gráficos pelo qual o servidor ESP8266 seria responsável. Infelizmente o uso desta linguagem se tornou bem problemática, ao fato de que o servidor ESP8266 não reconhecia os trechos do código correspondentes ao PHP.

A alternativa para prosseguir com o desenvolvimento da página, deu-se pela leitura dos arquivos e geração dos gráficos por linguagem JavaScript, que atende de forma simplificada ao processo dinâmico contido na página *web*.

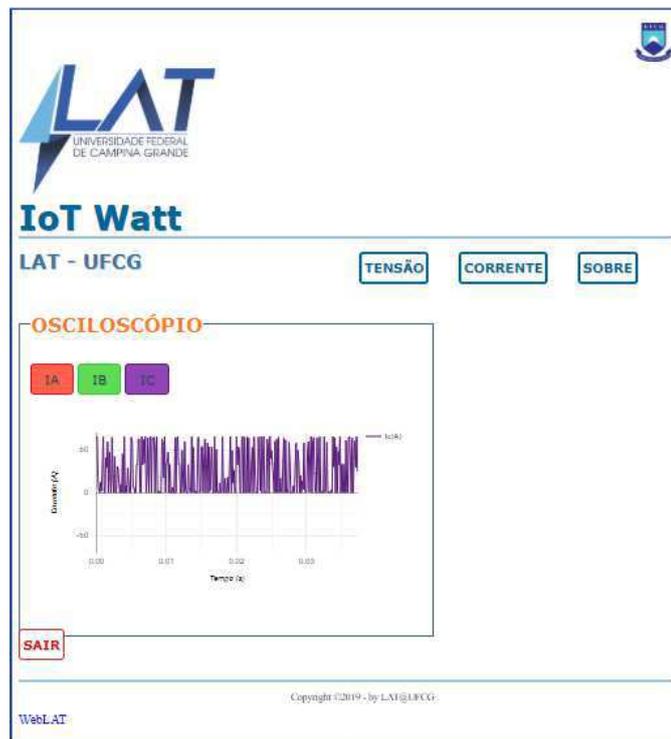
Na figura 9 é demonstrada a página *web* em funcionamento com a descrição da interface principal, que fornecerá os valores eficazes de tensão e corrente em cada fase. A página seguinte descrevem a representação de tensão e corrente com acesso direto ao resultado dos ADCs.

Figura 9: Página web principal desenvolvida para o wattímetro.



Fonte: Próprio autor.

Figura 10: Páginas secundárias desenvolvidas para o wattímetro.



Fonte: Próprio autor.

Das linguagens em uso citadas, pode se dizer que foram aplicadas as principais tecnologias da *World Wide Web*. O método optado para operar os dados faz exigências de processamento ao cliente (o computador que mostra os dados remotamente será responsável por garantir a dinâmica da página web, e não o servidor ESP8266).

A leitura para as páginas secundárias referentes aos sinais instantâneos de tensão e corrente ainda precisam ser ajustados. Não pelo processo da página *web* que os leem, mas sim por revisão na metodologia em que são lidas as medições por acesso direto à memória dos ADCs do M90E36A.

3.4 ETAPAS DE CALIBRAGEM DO WATTÍMETRO

Uma vez que o wattímetro foi montado em um *proto-board*, foi possível iniciar as etapas referentes à calibragem com os seus testes elaborados pela sequência de procedimentos descritos no *datasheet* do M90E36A e por documentos que citam aplicações com o circuito integrado[9][10].

A finalidade desta etapa corresponde em testar tanto a leitura do wattímetro quanto a resposta do servidor *web* para transmitir os dados de corrente e tensão eficazes em tempo real.

3.4.1 Circuitos Atenuadores

As especificações elétricas dos canais analógicos diferenciais são exibidas na Tabela 3, descritas no *datasheet* do M90E36A[9]. Sua leitura é necessária para o estabelecimento dos valores-limite que o circuito de atenuação deverá atender.

Tabela 3: Especificações elétricas do M90E36A em seu *datasheet*.

Parâmetro	Mínimo	Típico	Máximo	Unidade
Tensão Diferencial de Entrada	0, 12		720	mVrms
Tensão da Entrada Analógica	GND-300		VDD-1200	mV
Impedância de Entrada		120		kΩ

Fonte: *Datasheet* do M90E36A, 2013.

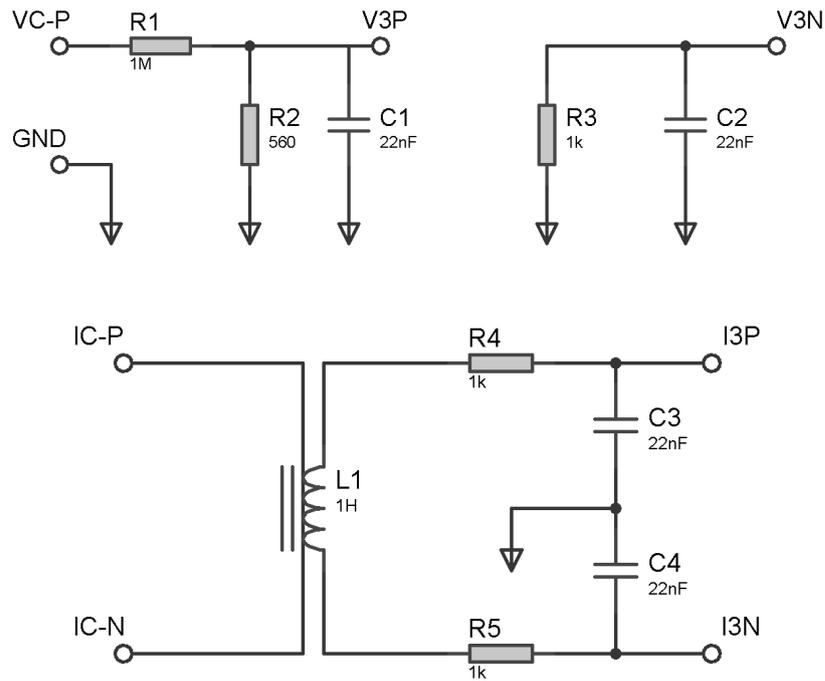
Três fatores importantes precisam ser avaliados para projetar os circuitos atenuadores: a divisão de tensão para o sinal de entrada, com o uso de resistores de 560 e 1MΩ; a relação de transformação para o sensor de corrente STC-13-005 (5A/1V)[7]; e a faixa de frequência projetada ao filtro anti-*aliasing*. Considerando que o circuito de atenuação da tensão suportará tensões de até 380 Vrms na entrada e considerando uma corrente máxima na entrada de 1 Arms, calcula-se os valores de pico que os ADCs estarão submetidos:

$$V_{VADC_{max}} = \frac{560}{1M + 560} \cdot 380\sqrt{2} = 300,77mV \quad (6)$$

$$V_{IADC_{max}} = \frac{1}{5} \cdot 1\sqrt{2} = 282,84mV \quad (7)$$

Os circuitos atenuadores foram montados em *proto-board*, cujo esquema de montagem é mostrado na figura abaixo. Vale ressaltar que o M90E36A apresenta ADCs diferenciais, com necessidade do circuito de atenuação apresentar terminais para a entrada positiva e um ponto comum referencial (adicionado um filtro) para cada canal negativo.

Figura 11: Projeto para o circuito de atenuação para suporte dos ADCs.



Fonte: Próprio autor, projetado no Proteus.

3.4.2 Registradores de Configuração

As etapas para configurar os registradores do M90E36A são mostradas logo abaixo. Em cada item será especificado o endereço do registrador, o conteúdo a ser escrito, e uma breve descrição do que isto afeta no funcionamento do chip. No caso de alterações à configuração aqui feita, consultar o *datasheet* do M90E36A[9].

- **0x00** → **0x789A**: Executa um *reset* para o M90E36A;
- **0x30** → **0x5678**: Habilita a calibragem dos registradores de configuração: o Circuito Integrado (CI) automaticamente estabelece os valores-padrão destes registradores;
- **0x33** → **0x1087**: Define a calibragem da associação ADC-fase, da frequência de operação, habilita os ADCs com filtros passa-alta, etc;
- **0x3B** → **f(0x31, 0x3A)**: Uma composição do MSB e LSB dos 10 registradores entre 0x31 a 0x3A é salva no registrador 0x3B;
- **0x30** → **0x8765**: Habilita comparar o valor registrado em 0x3B, ao resultado da operação anterior, colocando em nível alto o pino indicado no *datasheet* como *WarmOut* do M90E36A caso sejam diferentes ou encerra-se a operação.

3.4.3 Calibragem dos Ganhos de Corrente e Tensão Eficazes

Esta etapa compõe a escrita dos registradores que avaliam o ganho dos dados adquiridos. É necessário um equipamento de medição que sirva de referência aos possíveis ajustes no valor do ganho. As etapas a serem executadas são as seguintes:

- **0x60** → **0x5678**: Permite calibragem dos registradores de ganho;
- Feito a leitura do M90E36A e do equipamento de instrumentação de referência, relaciona-se os resultados aos ganhos para tensão e corrente pelas equações 8 e 9 correspondentes aos valores inicialmente especificados por experimentos de uso com o M90E36A (Plateny)[10]:

$$G_v = \frac{V_{ref}}{V_{M90E36A}} \cdot 52.800, \quad (8)$$

$$G_i = \frac{I_{ref}}{I_{M90E36A}} \cdot 30.000. \quad (9)$$

- As compensações G_v e G_i são usadas como fatores multiplicadores dos resultados e então verifica-se a nova saída. São escritos os ganhos diretamente sobre os registradores de ganho para tensão e corrente eficazes de cada fase (0x61, 0x62, 0x65, 0x66, 0x69 e 0x6A);
- **0x6F** → **f(0x61, 0x6E)**: Semelhante ao processo de avaliação da configuração dos registradores com o registrador 0x3B;
- **0x60** → **0x8765**: Habilita comparar o valor registrado em 0x6F, ao resultado da operação anterior, colocando um bit específico em nível alto caso sejam diferentes ou encerra-se a operação.

O processo para ajuste dos valores instantâneos apresentam etapas semelhantes ao procedimento para os dados de tensão e corrente eficazes, necessário à etapa que fará a compensação no acesso direto à memória. Consultar o *datasheet* do M90E36A sobre quais registradores se deve atuar[9].

3.4.4 Registradores de Ajuste do Ângulo de Fase

O processo se assemelha à calibragem dos ganhos de tensão e corrente. Necessário para ajustar o erro devido às distorções provocadas pelos circuito atenuadores, e em menor grau, pelo atraso da resposta dos sensores de corrente (que são ínfimos, pois os sinais operados estão situados na zona linear de operação do STC-13). Pode-se listar os registradores atuantes neste caso:

- **0x40** Habilita configuração e validação do processo;
- **0x48, 0x4A, 0x4C** Registradores a serem escritos as defasagens;
- **0x4D** Registrador que grava dados úteis à checagem da operação.

A técnica para especificar qual valor de compensação consiste em analisar cada fase que alimenta uma carga com fator de potência unitário. Ao se detectar um valor de potência reativa Q erroneamente, relaciona-se a defasagem ao valor mostrado da potência ativa P pela Equação 10:

$$\varphi_{Atraso} = \tan^{-1} \left(\frac{Q}{P} \right) \quad (10)$$

Como os sinais típicos que serão operados pelo wattímetro são de corrente alternada à frequência de $60Hz$, um ciclo angular de 360° corresponde a um ciclo temporal $t = 1/60s$. Na Equação 11 é feita a conversão do atraso angular para o atraso temporal:

$$t_{Atraso} = \frac{\varphi_{Atraso}}{360 \times 60} \quad (11)$$

Os registradores do M90E36A geram um *delay* em ciclos de $2,048MHz$. Sabendo dessa informação, calcula-se o fator de ajuste para o ângulo de fase de acordo com o está descrito na Equação 12:

$$\Delta_\varphi = t_{Atraso} \cdot 2,048 \times 10^6 \quad (12)$$

O protótipo do wattímetro ainda está no estágio de montagem do *proto-board*. Pretende-se continuar com a confecção do wattímetro ainda após o encerramento da descrição do projeto.

4 CALIBRAGEM DO WATTÍMETRO

As atividades encontram-se atualmente no processo de calibragem, juntamente com a construção do *layout* para o wattímetro. Como demonstração, será descrito o ajuste para os valores de corrente eficaz, tensão eficaz e potência ativa para a fase C (com à descrição de referência do M90E36A)[9].

Primeiramente, a bancada de testes foi montada como mostrado na Figura 12. A montagem está localizada no setor de Qualidade do Laboratório de Alta Tensão da UFCG. O teste fez uso dos seguintes equipamentos:

- Wattímetro;
- Osciloscópio;
- *Laptop*;
- Fonte de tensão programável;
- Alicates-amperímetro *Hioki 3283*;
- Conjunto de lâmpadas incandescentes.

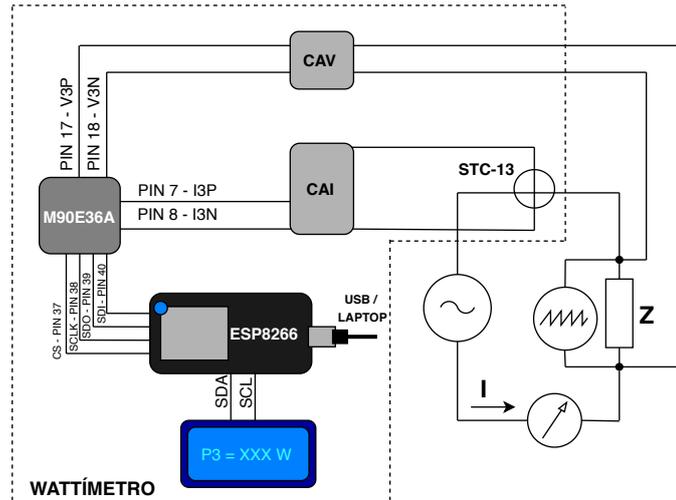
Figura 12: Montagem da bancada para os primeiros testes de calibragem do wattímetro.



Fonte: Próprio autor.

Um esquemático é mostrado na Figura 13 e destaca os elementos componentes do wattímetro (Figura 13) Além do CI M90E36A e do módulo ESP8266, está indicado o sensor STC-13 e os circuitos atenuadores de tensão e corrente (CAV , CVI) que foram descritos no capítulo anterior.

Figura 13: Esquemático simplificado das conexões para a bateria de testes do wattímetro.



Fonte: Próprio autor.

Após definir os registradores de configuração ao modo de operação do sistema, foram obtidas as primeiras medições mostradas pela página web para V_{rms} e I_{rms} da fase C, de acordo com o valores padrão de ganho $G_{v_o} = 52.800$ e $G_{i_o} = 30.000$.

As medições do alicate-amperímetro *Hioki 3283* serviram de referência para a intensidade da corrente elétrica eficaz. Enquanto para a referência de tensão utilizava-se a própria indicação da fonte de tensão programável, confirmados pelas medições do osciloscópio.

Os valores de referência para o cálculo são: $V_{ref} = 100V_{rms}$ e uma carga resistiva que solicita uma corrente de $I_{ref} = 100A_{rms}$.

Com os resultados do amperímetro e do wattímetro desenvolvido, foi possível obter a primeira compensação a ser introduzida no registrador referente aos parâmetros da fase C.

$$G_{v_c} = \frac{100}{92,65} \cdot 52.800 \approx 56989 \quad (13)$$

$$G_{i_c} = \frac{1,07}{20,28} \cdot 30.000 \approx 1.583 \quad (14)$$

Para um ganho G_{v_c} , uma faixa de valores para a tensão foi mensurada e comparada com o resultado demonstrado pela fonte programável. Os resultados podem ser visualizando na tela do wattímetro ou mesmo remotamente com a página web (Figura 14). Averiguando se o ganho foi suficiente ou se faz necessário algum ajuste quanto ao *offset* dos resultados.

Figura 14: Primeiros resultados para o Wattímetro com ajustes nos ganhos de tensão e corrente.



Fonte: Próprio Autor.

Tabela 4: Resultados com ajuste no ganho para tensão eficaz da fase C.

Medição	V_{rms} Oscl.(V)	V_{rms} Wattímetro.(V)	Erro (%)
1	10	10,16	1,60
2	20	20,11	0,55
3	30	30,07	0,23
4	40	40,04	0,10
5	50	49,98	0,04
6	60	59,95	0,08
7	70	70,01	0,01
8	80	80,06	0,08
9	90	89,96	0,04
10	100	100,05	0,05
11	110	110,02	0,02
12	120	120,12	0,10
13	130	130,06	0,05
14	140	140,03	0,02
15	150	150,08	0,05

Fonte: Próprio autor.

A faixa de erro estava menor que 5%. Foi decidido prosseguir com o ajuste da corrente. Para ganho o G_{i_c} , uma faixa de valores para corrente foi mensurada e comparada com o mesmo equipamento auxiliar. Averiguando se o ganho foi suficiente ou se faz necessário algum ajuste quanto ao *offset* dos resultados.

Tabela 5: Resultados com ajuste no ganho para corrente eficaz da fase C.

Medição	<i>I_{rms}</i> Hioki 3283 (A)	<i>I_{rms}</i> Wattímetro (A)	Erro (%)
1	0,38	0,38	0,53
2	0,51	0,50	1,38
3	0,60	0,60	0,17
4	0,68	0,68	0,29
5	0,76	0,75	0,92
6	0,83	0,82	0,97
7	0,89	0,89	0,45
8	0,96	0,95	0,73
9	1,01	1,01	0,00
10	1,07	1,07	0,00
11	1,13	1,13	0,00
12	1,18	1,18	0,00
13	1,24	1,23	0,81
14	1,29	1,28	0,78
15	1,34	1,33	0,75

Fonte: Próprio autor.

Com os dois parâmetros de tensão e corrente ajustados. É avaliado então a medição para a potência ativa e reativa em para C. O Valor de referência corresponde ao produto das medições de tensão e corrente eficazes, que se tratando de uma carga resistiva, corresponde a valor de potência ativa consumida pela carga.

Para os valores aplicados ao ganho de tensão e corrente, o wattímetro indicou $P_c = 103,15W$ e $Q_c = 26,2W$. O que para medição de uma carga resistiva, claramente indicava a necessidade de ajustar algum parâmetro. Primeiramente optou-se por calibrar o ângulo da fase C.

$$\varphi_{Atraso} = \tan^{-1} \left(\frac{26,2}{103,15} \right) = 14,25^\circ \quad (15)$$

$$t_{Atraso} = \frac{14,25^\circ}{360 \times 60} = 6,59^{-4} \quad (16)$$

$$\Delta_\varphi = 6,59^{-4} \cdot 2,048 \times 10^6 \approx 1351 \quad (17)$$

Aplicado ao registrador correspondente ao ângulo de fase C, obteve-se os primeiros resultados para a potência elétrica.

Tabela 6: Resultados com ajuste no ganho para corrente eficaz da fase C.

Medição	P_{ref} (W)	P Wattímetro (W)	Erro (%)
1	3,78	3,77	0,26
2	10,14	10,01	1,28
3	18,03	17,82	1,16
4	27,28	26,94	1,25
5	37,85	36,5	3,57
6	49,68	49,21	0,95
7	62,58	61,53	1,68
8	76,56	75,92	0,84
9	90,9	90,85	0,06
10	107	106,94	0,06
11	124,3	123,72	0,47
12	141,6	141,42	0,13
13	161,2	160,1	0,68
14	180,6	179,9	0,39
15	201	199,91	0,54

Fonte: Próprio autor.

A próxima etapa avaliaria para uma carga com $f.p = 0,5$ indutivo, servindo de ajuste direto dos registradores de ganhos para potência ativa e reativa, além de possíveis ajustes para *offset*.

Essa análise inicial permite prosseguir com outras metodologias de calibragem do wattímetro. Os resultados, apesar de satisfatórios, não definem uma categoria de classificação para o wattímetro, é preciso um estudo detalhado das normas vigentes sobre as etapas de calibragem para o equipamento de instrumentação proposto.

O progresso para calibrar outros parâmetros será feita após o wattímetro estar montado em uma placa de circuito impresso, pois é necessário avaliar sobre cada fase, submetido a distorções em cada conjunto de sensor + circuitos atenuadores de tensão e corrente.

5 CONCLUSÃO

A construção do wattímetro trifásico microprocessado foi parcialmente concluída. Ainda montado em *proto-board*, foi possível medir a potência consumida ou fornecida por uma carga, e isto inclui os primeiros procedimentos para calibragem do equipamento em uma fase nas faixas de tensão até 150 V e corrente em até 1,33 A. Ainda é necessário fazer a calibragem para as outras fases, além do ajuste nos circuito atenuadores para que o equipamento possa operar seguramente nas faixas de tensão e corrente propostas em projeto.

De acordo com o que estava descrito para os objetivos específicos, o projeto de funcionalidades básicas para o Wattímetro foi alcançado com a montagem e os primeiros testes de medição.

O desenvolvimento de extensão das funcionalidade do wattímetro e sua interação com o usuário foi parcialmente alcançado, com a criação da página *web* que faz uso dos dados gerados no equipamento para leitura remota dos parâmetros mensurados, demonstrados, inclusive aplicados para as primeiras avaliações de desempenho do wattímetro. Porém precisa ser ajustado o algoritmo de acesso direto à memória dos ADCs para os sinais de corrente e tensão que deverão ser exibidos na página *web*.

As dificuldades encontradas se refletem pela necessidade de dedicação quase exclusiva para o projeto, e isto afetou seriamente a não conclusão da construção definitiva do wattímetro.

No processo de calibragem para uma fase, foi possível testar e validar o a continuação do protótipo por testes comparativos de operação com os resultados obtidos por um osciloscópio, que para as primeiras medições, os erros máximos atingiram 1,60% para a magnitude de tensão eficaz, 1,53% para os registros de intensidade da corrente eficaz e 3,57% para potência ativa. Esta etapa foi concluída. O que motiva a prosseguir o desenvolvimento do equipamento de instrumentação.

O prazo não foi suficiente para produzir um manual simplificado didático para uso do wattímetro, que só poderá ser iniciado apenas na proximidade de montagem definitiva do equipamento.

No decorrer das atividades, foi possível adquirir conhecimentos sobre as etapas que envolviam projeto com embarcados, análise da arquitetura de computadores, projetos com eletrônica analógica e programação *web*, tornando o projeto um meio para remediar certas carências da formação acadêmica do projetista. Aliado a compreensão que já se tinha no ramo da eletrotécnica sobre equipamentos de instrumentação, o projeto forneceu uma visão mais generalista do tema, correspondendo à formação acadêmica de Engenharia Elétrica da UFCG.

6 ATIVIDADES FUTURAS

6.1 PLACA DE CIRCUITO IMPRESSO

Um módulo será produzido para acomodar os circuitos e garantir a conexão dos sinais de interesse de forma efetiva, evitando maiores gastos com a aquisição de módulos comerciais com funcionalidades parciais ao que é estabelecido na construção do wattímetro.

O estágio do wattímetro até o desenvolvimento deste documento se encontra na fase projeto para a construção da placa de circuito impresso (PCI), que está sendo elaborada no ambiente ARES, da ferramenta Proteus.

Quanto a produção desta placa, por praticidade, está pré-definido o emprego de componentes discretos para a montagem da placa, montado com resistores de alta precisão (1% de tolerância), minimizando o erro, especialmente ao montar os circuitos de condicionamento dos sinais de tensão.

Alguma solução deverá ser adicionada ao equipamento para garantir a segurança do mesmo, visto que muitos Wattímetros desenvolvidos na universidade foram danificados, em boa parte, por mau uso do equipamento pelos estudantes da disciplina Laboratório de Equipamentos Elétricos.

6.2 OUTRAS ETAPAS

- Construção de uma caixa protetora para o wattímetro;
- Realizar testes empregados ao experimento para ensaio de transformadores, seguindo o roteiro do guia de experimentos do LAT;
- Produzir um guia de uso do equipamento;

6.3 PESQUISAS FUTURAS: CONTINUAÇÃO DA CALIBRAGEM DO WATTÍMETRO

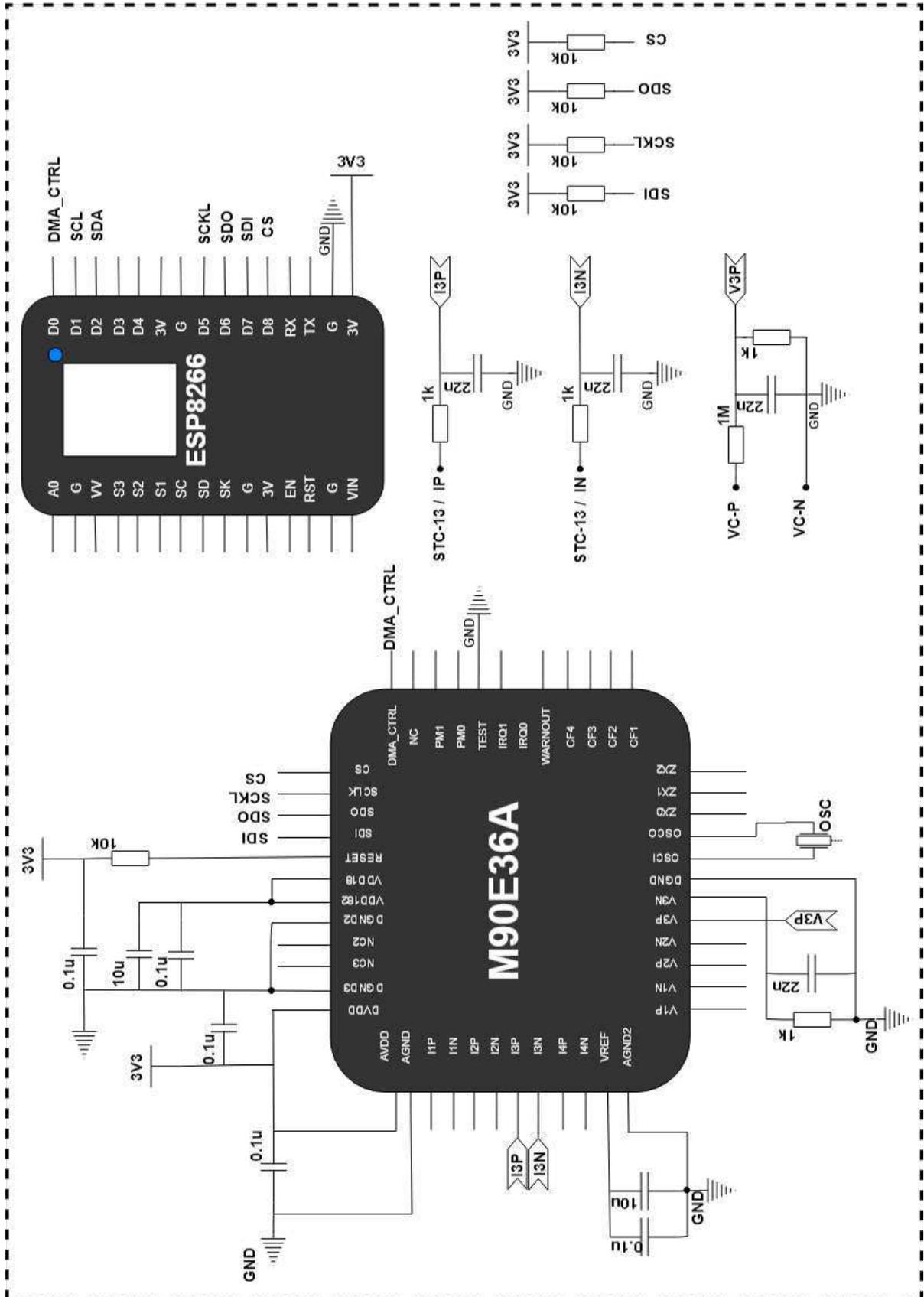
A demonstração para avaliar o funcionamento do wattímetro não é suficiente para classificá-lo, apenas habilita a continuidade do procedimento de calibragem. Aos decorrer de um número suficiente de testes, cujas normas relacionadas à categoria de deverão ser estudadas e posta em prática para prosseguimento deste projeto, avaliando para toda a faixa de operação que foi estabelecida para o wattímetro inicialmente. Exemplos de normas a serem estudadas e aplicadas: IEC 61000-4-7/15/30; IEEE 519/1159.

7 REFERÊNCIAS

- [1] LAT, *Guia de Experimentos - Laboratório de Equipamentos Elétricos: Transformadores de Potência*. Campina Grande, PB, BR: DEE/UFCG, 2008.
- [2] L. A. Barbosa, “Wattímetro microprocessado, trabalho de final de curso,” 2016.
- [3] J. M. Filho, *Manual de Equipamentos Elétricos*. Rio de Janeiro, RJ, BR: GEN, 2013.
- [4] “IEEE Standard Definitions for the Measurement of Electric Power Quantities under Sinusoidal, Non-Sinusoidal, Balanced, or Unbalanced Conditions,” *IEEE Std 1459-2000*, pp. 1–52, June 2000.
- [5] J. P. G. D. A. João Roberto Cogo, José Carlos de Oliveira, *Transformadores: Teoria e Ensaio*. São Paulo, SP, BR: Blucher, 1984.
- [6] F. das Chagas, *Notas de Aula - Transformadores de Corrente*. Campina Grande, PB, BR: DEE/UFCG, 2017.
- [7] YHDC, “Split Core Current Transformer,” 2013.
- [8] Espressif, “ESP8266 Technical Reference,” 2016.
- [9] Atmel, “Poly-Phase High-Performance Wide-Span Energy Metering IC 90e36a,” 2013.
- [10] P. de Brito Ponchet, “Avaliação do transformador de corrente sct-013 aplicado em medidor eletrônico de potência, trabalho de final de curso,” 2016.

A APÊNDICE

A.1 ESQUEMA DO CIRCUITO DO WATTÍMETRO



A.2 CÓDIGOS IMPLEMENTADOS E ADAPTADOS

```

1  /*-----UNIVERSIDADE FEDERAL DE CAMPINA GRANDE-----
2  %-----DEPARTAMENTO DE ENGENHARIA ELETRICA E INFORMATICA-----
3  %-----
4  %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
5  %Victor Hugo dos Santos Germano e-mail: victor.germano@e.ufcg.edu.br
6  %-----
7  %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
8  % WATTIMETRO
9  %%%%%%%%%%%%%%%%%%%%%%%%%%%%%*/
10
11 //Adaptado, contribuiçao de Plateny de Brito Ponchet
12
13 //*****DEFINIÇÕES*****
14 //*****
15
16 //Numeraç'2o dos pinos do ESP8266 para sincronizar comunicaç'2o entre perifericos
17 #define START_WATTIMETRO 0 //Iniciar Wattimetro
18 #define M0_CHIP_SELECT 15 //CS para o M0E32A
19 #define M0_DMA_CTRL 2 //DMA para dados dos ADCs fornecidos M0E32A
20 // #define M0_RESET 16 //Reset M0E32A
21 #define SCREEN_WIDTH 128 //OLED largura do display
22 #define SCREEN_HEIGHT 64 //OLED altura do display
23 #define OLED_RESET -1 //Resetar OLED
24
25 //Fatores de escala para o M0E36A
26 #define N 10
27 #define FACTOR 1
28 #define URES 0.0000390625
29 #define IRES 0.00000390625
30 #define POWRES 0.00390625
31 #define TOTALPOWRES 0.015625
32 #define PFRES 0.001
33 #define H_RES 0.006103515625
34 #define UF_RES 0.032656
35 #define IF_RES 0.0032656
36 #define F_RES 0.009767252604167
37 #define START 1
38 #define END 0
39 #define SAMPLE_RATE 8000
40
41 //Deseja calibrar sensores ? 1. SIM 0. NAO
42 #define CALIBRATE_VOLTAGE 1
43 #define CALIBRATE_CURRENT 1
44
45 //Definição bibliotecas para o Wattimetro
46 #include <SPI.h> //Comunicação com o M0E32A SPI;
47 #include "Thread.h"; //Ferramenta para organizar e sincronizar
48 #include "ThreadController.h"; // atividades simultaneas;
49 #include <ESP8266WiFi.h> //Servidor Web e sincronizar Time.h;
50 #include <ESP8266WebServer.h> //
51 #include <ESP8266mDNS.h> //
52 #include <Wire.h> //Ferramenta para comunicação com o OLED
53 #include <Adafruit_GFX.h> //Biblioteca da configurações do display
54 #include <Adafruit_SSD1306.h> //Biblioteca da classe OLED
55 #include "FS.h" //SPIFFS
56
57 //VARIÁVEIS E FUNÇÕES*****
58
59 //Definindo variáveis enumeradas para o M0E32
60 enum CHANNELS {UA, UB, UC, IA = 4, IB, IC, IN, TOTAL = -1};
61 enum CHECKSUM_NUMBER {CONFIG, ENERGY, FUNDAMENTAL_HARMONIC, MEASUREMENT};
62
63 //Definindo variáveis do Time.h
64 int timezone = -3 * 3600; //no Brasil a zona eh -3
65 int dst = 0; // horario de verao
66
67 //Definindo a classe OLED
68 Adafruit_SSD1306 display(SCREEN_WIDTH, SCREEN_HEIGHT, &Wire, OLED_RESET);
69
70 //Definindo a classe txt
71 FsFile Parametro_TXT;
72
73 //variáveis para Botao

```

```

74 int aux_med = 0;
75 bool estadoBotao, estadoBotaoAnterior, auxiliar;
76
77 //Conex2o Wreless: login e senha
78 //const char* ssid = "nomeDaRede";
79 //const char* password = "senha";
80
81 //Servidor Web
82 ESP8266WebServer server(80);
83
84 //Biblioteca de Funcoes para o CI
85 #include "M0E326.h"
86 //Biblioteca de Funcoes para o uC
87 #include "uC_CONFIG.h"
88
89 //Atividades do Microcontrolador
90 Thread TXT_MEDICOES; //Dados do ADC -> medicoes.txt
91 Thread OLED_PARAMETROS; //Mostrar parametros no OLED
92 Thread TXT_PARAMETROS; //Registra parametros em parametros.txt
93 Thread BUTTON; //Botao para sair ou iniciar leitura no OLED
94
95 //SETUP PRINCIPAL*****
96 void setup() {
97
98 //Iniciar Serial
99 Serial.begin(115200);
100 SPIFFS.begin();
101 //Conectado a rede wireless
102 WiFi.mode(WIFI_STA);
103 WiFi.begin(ssid, password);
104
105 //Definindo pinos de saida
106 //Nota: os pinos do SPI s2o automaticos exceto o CS
107 pinMode(M0_CHIP_SELECT, OUTPUT);
108 pinMode(M0_DMA_CTRL, OUTPUT);
109 pinMode(START_WATTIMETRO, INPUT);
110 //Iniciar OLED
111 OLED_START();
112
113 //M0E36A SETUP*****
114 //Verificando comunicacao com o CI
115 Serial.println("Medidor Inicializado.");
116 Serial.println("Calibrando...");
117
118 while (!pinCSetup()) {
119 Serial.println("Falha no M0E36A.");
120 }
121 display.print("\n 1. M0E36A OK.");
122 display.display();
123
124 //Verificando calibragem
125 measurementCalibraton(START);
126 energyCalibraton(START);
127 solitacao();
128 measurementCalibraton(END);
129
130 //WIFI SETUP*****
131
132 //Verificando comunicacao com o WiFi
133 while (WiFi.status() != WL_CONNECTED) {
134 delay(500);
135 Serial.print(".");
136 }
137 Serial.print("\nNodeMCU IP Address : ");
138 Serial.println(WiFi.localIP());
139 if (MDNS.begin("esp8266")) {
140 Serial.println("\nMDNS responder started");
141 }
142 display.print("\n 2. WiFi OK.");
143 display.display();
144
145 //TIME SETUP*****
146 //Obtendo hor@rio atual

```

```

147 configTime(timezone, dst, "time.nist.gov", "de.pool.ntp.org");
148 Serial.println("\nAguardando Wireless para o Time.h");
149 while(!time(NULLptr)){
150     Serial.print("*");
151     delay(200);
152 }
153 Serial.println("\nTime Class ....OK");
154 time_t now = time(NULL);
155 struct tm* p_tm = localtime(&now);
156 display.print("\n 3. Data: ");
157 display.print(p_tm->tm_mday); display.print("/");
158 display.print(p_tm->tm_mon + 1); display.print("/");
159 display.print(p_tm->tm_year + 1900);
160 display.display();
161 delay(1500);
162 display.clearDisplay();
163 display.drawBitmap(0, 0, LAT_Logo, 128, 64, WHITE);
164 display.display();
165 delay(3500);
166
167 //THREAD SETUP*****
168 //Mostrar dados dos ADCs
169 TXT_MEDI COES. setInterval(4000); //atue cada 4s
170 TXT_MEDI COES. onRun( outputDMA);
171
172 TXT_PARAMETROS. setInterval(2000); //atue cada 4s
173 TXT_PARAMETROS. onRun( fileWrite);
174
175 OLED_PARAMETROS. setInterval(2200); //atue cada 2.2s
176 OLED_PARAMETROS. onRun( OLED_READ);
177
178 BUTTON. setInterval(700); //atue cada 700ms
179 BUTTON. onRun( detectaBotao);
180
181 //WEBPAGE SETUP*****
182 server.on("/", HTTP_GET, servidor_Pag_Inicial);
183 server.on("/pag_inicial", HTTP_GET, servidor_Pag_Inicial);
184 server.on("/tensao", HTTP_GET, servidor_Pag_Tensao);
185 server.on("/corrente", HTTP_GET, servidor_Pag_Corrente);
186 server.on("/style", HTTP_GET, servidor_cssStyle);
187 server.on("/LAT", HTTP_GET, imagem_LAT);
188 server.on("/UFCEG", HTTP_GET, imagem_UFCEG);
189 server.on("/medicoes", medicoes);
190 server.on("/parametros", parametros);
191 server.onNotFound( handleNotFound);
192 server.begin();
193 }
194 //-----
195 void loop() {
196
197     server.handleClient();
198
199     if(OLED_PARAMETROS.shouldRun()){
200         OLED_PARAMETROS.run();
201     }
202     if(TXT_PARAMETROS.shouldRun()){
203         TXT_PARAMETROS.run();
204     }
205     if(TXT_MEDI COES.shouldRun()){
206         TXT_MEDI COES.run();
207     }
208     if(BUTTON.shouldRun()){
209         BUTTON.run();
210     }
211 }
212

```

```

1  /*-----UNIVERSIDADE FEDERAL DE CAMPINA GRANDE-----
2  %-----DEPARTAMENTO DE ENGENHARIA ELETRICA E INFORMATICA -----
3  %-----
4  %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
5  % Victor Hugo dos Santos Germano e-mail: victor.germano@ee.ufcg.edu.br
6  %-----
7  %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
8  % BIBLIOTECA DE FUNCOES M0E36A
9  %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%*/
10
11 //Adaptado, contribuiçao de Lenilson Andrade Barbosa e Plateny de Brito Ponchet
12
13 #ifndef M0E36A // Se a biblioteca nao estiver definida...
14 #define M0E36A //...defina.
15
16 //PROTOTIPO DE FUNCOES*****
17 uint16_t read16(uint16_t); //leitura dos dados
18 void write16(uint16_t, uint16_t); //escrita dos dados
19 bool pmcSetup(); //regs de configuracao
20 uint16_t csCalculator(byte, byte); //calcula valor para o reg de validacao
21 bool checkStatus(int); //teste de validacao do reg
22 uint16_t measurementGainCalibration(int, float); //recalcula e regos ganhos rms
23 uint16_t measurementOffsetCalibration(int); //recalcula e registra o offset rms
24 bool measurementCalibration(bool); //habilita a configuracao dos ganhos rms
25 bool energyCalibration(bool); //recalcula e registra os ganhos P,Q
26 void dftStart(); //
27 float frequency(); //retorna o valor da frequencia
28 uint32_t rms(int8_t); //retorna o valor rms
29 float realRms(int8_t); //retorna o valor rms com fator de escala
30 float neutralCurrentRms(); //retorna o valor da corrente no neutro
31 uint16_t harmonic(int8_t, int8_t); //retorna o valor da mag da harmonica i
32 float activeHarmonicPower(int8_t); //retorna o valor da P_harmonica
33 float activeFundamentalPower(int8_t); //retorna o valor da P_fundamental
34 uint16_t fundamental(int8_t); //retorna o valor da mag da fundamental
35 float activePower(int8_t); //retorna o valor da P ativa (total)
36 float reactivePower(int8_t); //retorna o valor da Q reativa (total)
37 float aparentPower(int8_t); //retorna o valor da S aparente (total)
38 float powerFactor(int8_t); //retorna o fator de potencia
39 int32_t twosComplementToDecimal(uint32_t, const int); //complemento de 2 adaptado
40 int32_t powOfTwo(int8_t); //potencia de 2 adaptado
41 void outputDMA(); //Uso do DMA (precisa ser revisado)
42
43 //*****
44 uint16_t read16(uint16_t address) {
45     uint16_t readData;
46     SPI.beginTransaction(SPI_Settings(1000000, MSBFIRST, SPI_MODE3));
47     digitalWrite(M0_CHIP_SELECT, LOW);
48     SPI.transfer16(0x8000 | address);
49     readData = SPI.transfer16(0x0000);
50     digitalWrite(M0_CHIP_SELECT, HIGH);
51     SPI.endTransaction();
52     delay(1);
53     return readData;
54 }
55 void write16(uint16_t address, uint16_t data) {
56     SPI.beginTransaction(SPI_Settings(1000000, MSBFIRST, SPI_MODE3));
57     digitalWrite(M0_CHIP_SELECT, LOW);
58     SPI.transfer16(address);
59     SPI.transfer16(data);
60     digitalWrite(M0_CHIP_SELECT, HIGH);
61     SPI.endTransaction();
62     delay(1);
63     return;
64 }
65
66 bool pmcSetup() {
67     SPI.begin();
68     digitalWrite(M0_CHIP_SELECT, HIGH);
69     //Hardware Reset
70     //digitalWrite(M0_RESET, LOW);
71     //delay(20);
72     //digitalWrite(M0_RESET, HIGH);
73     delay(500);

```

```

74 //Software Reset
75 write16(0x00, 0x789A);
76 delay(100);
77 //Configuration
78 write16(0x30, 0x5678); //Permitir Calibragem
79 write16(0x33, 0x1087); //Calibrando: fase - Num do ADC, frequencia ...
80 write16(0x3B, csCalculator(0x31, 0x3A)); //Composiçao do MSB e LSB dos reg.
81 // // 0x31 a 0x3A salva em 0x3B
82 write16(0x30, 0x8765); //Habilita checagem do checksum
83 delay(1000);
84 if(read16(0x33) != 0x1087) return false;
85 return !checksumStatus(CONFIG);
86 }
87
88 uint16_t measurementGainCalibration(int channel, float ref) {
89     uint8_t i;
90     uint16_t gain;
91     float meas, attenuatedRef;
92
93     for (meas = 0, i = 0; i < N; i++) {
94         meas += rms(channel);
95         delay(500);
96     }
97     meas = meas / N;
98
99     Serial.print("Valor medido: "); Serial.println(meas);
100    Serial.print("Valor referenciado: "); Serial.println(ref);
101
102    switch (channel) {
103        case UA: case UB: case UC:
104            meas = meas * URES;
105            gain = 52800 * ref / meas;
106            write16(0x61 + channel * 4, gain);
107            break;
108        case IA: case IB: case IC:
109            meas = meas * IRES;
110            gain = 30000 * ref / meas;
111            write16(0x62 + (channel - 4) * 4, gain);
112            break;
113    }
114    return gain;
115 }
116
117 uint16_t measurementOffsetCalibration(int channel) {
118     uint8_t i;
119     int16_t offset;
120     float meas;
121
122     for (meas = 0, i = 0; i < N; i++) {
123         meas += rms(channel);
124         delay(500);
125     }
126     meas = meas / N;
127
128     switch (channel) {
129         case UA: case UB: case UC:
130             offset = (~offset) + 0x01;
131             write16(0x63 + channel * 4, offset);
132             break;
133         case IA: case IB: case IC:
134             offset = (~offset) + 0x01;
135             write16(0x64 + (channel - 4) * 4, offset);
136             break;
137     }
138     return offset;
139 }
140
141 bool measurementCalibration(bool opt) {
142     switch (opt) {
143         case 1: 79;
144             write16(0x60, 0x5678);
145             return 1;
146         case 0:

```

```

147     write16(0x6F, csCalculator(0x61, 0x6E));
148     write16(0x60, 0x8765);
149     return !checksumStat us( MEASUREMENT);
150 }
151 }
152
153 void sol icitacao(){
154     #if CALIBRATE_VOLTAGE == 1
155     Serial . print("UA: "); Serial . println( measurement GainCal i brati on( UA, 100));
156     Serial . print("UB: "); Serial . println( measurement GainCal i brati on( UB, 100));
157     Serial . print("UC: "); Serial . println( measurement GainCal i brati on( UC, 100));
158     #else
159     write16(0x61, 52800); //UA Gain
160     write16(0x65, 52800); //UB Gain
161     write16(0x69, 56989); //UC Gain
162     #endif
163     #if CALIBRATE_CURRENT == 1
164     Serial . print("IA: "); Serial . println( measurement GainCal i brati on( IA, 1.07));
165     Serial . print("IB: "); Serial . println( measurement GainCal i brati on( IB, 1.07));
166     Serial . print("IC: "); Serial . println( measurement GainCal i brati on( IC, 1.07));
167     #else
168     write16(0x62, 30000); //IA Gain
169     write16(0x66, 30000); //IB Gain
170     write16(0x6A, 1566); //IC Gain
171     write16(0x6D, 0); //IN Gain
172     #endif
173
174     #if CALIBRATE_VOLTAGE_OFFSET == 1
175     Serial . print("Offset UA: "); Serial . println( measurement Off set Cal i brati on( UA));
176     Serial . print("Offset UB: "); Serial . println( measurement Off set Cal i brati on( UB));
177     Serial . print("Offset UC: "); Serial . println( measurement Off set Cal i brati on( UC));
178     #else
179     write16(0x63, 0); //UA Offset
180     write16(0x67, 0); //UB Offset
181     write16(0x6B, 0); //UC Offset
182     #endif
183     #if CALIBRATE_CURRENT_OFFSET == 1
184     Serial . print("Offset IA: "); Serial . println( measurement Off set Cal i brati on( IA));
185     Serial . print("Offset IB: "); Serial . println( measurement Off set Cal i brati on( IB));
186     Serial . print("Offset IC: "); Serial . println( measurement Off set Cal i brati on( IC));
187     #else
188     write16(0x64, 0); //IA Offset
189     write16(0x68, 0); //IB Offset
190     write16(0x6C, 0); //IC Offset
191     #endif
192     write16(0x4C, 1800); //UC configuracao do angulo de fase
193 }
194
195
196 bool energyCal i brati on( bool opt) {
197     switch (opt) {
198     case 1:
199         write16(0x40, 0x5678);
200         return 1;
201     case 0:
202         write16(0x4D, csCalculator(0x41, 0x4C));
203         write16(0x40, 0x8765);
204         return !checksumStat us( ENERGY);
205     }
206 }
207
208 void dftStart() {
209     write16(0x1D1, 0x0001);
210     return;
211 }
212
213 uint16_t csCalculator( byte startRegister, byte endRegister) {
214     int i;
215     byte lowCsByte = 0, highCsByte = 0;
216     uint16_t aux;
217     for (i = startRegister; i <= endRegister; i++) {
218         aux = read16(i);
219         lowCsByte += (byte)(aux >> 8) + (byte)(aux & (0x00FF));

```

```

220     highCsByte = highCsByte ^ (byte)(aux >> 8) ^ (byte)(aux & (0x00FF));
221 }
222 return ((uint16_t)lowCsByte) + (((uint16_t)highCsByte) << 8);
223 }
224
225 bool checksumStatus(int number) {
226     switch (number) {
227         case CONFIG:
228             return bitRead(read16(0x01), 14);
229         case ENERGY:
230             return bitRead(read16(0x01), 12);
231         case FUNDAMENTAL_HARMONIC:
232             return bitRead(read16(0x01), 10);
233         case MEASUREMENT:
234             return bitRead(read16(0x01), 8);
235     }
236 }
237
238 float frequency() {
239     return read16(0xF8) * F_RES;
240 }
241
242 uint32_t rms(int8_t channel) {
243     return (((uint32_t)read16(0xD9 + channel)) << 8) + ((read16(0xE9 + channel) >>
244     8));
245 }
246 float neutralCurrentRms() {
247     uint32_t readData = 0;
248     readData = ((uint32_t)read16(0xDC)) << 8;
249     return readData * IRES / FACTOR;
250 }
251
252 float realRms(int8_t channel) {
253     switch (channel) {
254         case UA: case UB: case UC:
255             return rms(channel) * URES;
256         case IA: case IB: case IC:
257             return rms(channel) * IRES / FACTOR;
258     }
259 }
260
261 uint16_t harmonic(int8_t channel, int8_t order) {
262     if (order < 2 || order > 32)
263         return -1;
264     else {
265         switch (channel) {
266             case UA: case UB: case UC:
267                 return read16(0x160 + channel * 0x20 + (order - 2));
268             case IA: case IB: case IC:
269                 return read16(0x100 + (channel - 4) * 0x20 + (order - 2));
270         }
271     }
272 }
273
274 float activeHarmonicPower(int8_t channel) {
275     uint32_t readData = 0;
276     float ahp;
277     readData = (((uint32_t)read16(0xD5 + channel)) << 8) + (read16(0xE5 + channel) >>
278     8);
279     ahp = twosComplementToDecimal(readData, 24);
280     return (channel == TOTAL) ? (ahp * TOTALPOWRES / FACTOR) : (ahp * POWRES / FACTOR);
281 }
282
283 float activeFundamentalPower(int8_t channel) {
284     uint32_t readData = 0;
285     float afp;
286     readData = (((uint32_t)read16(0xD1 + channel)) << 8) + (read16(0xE1 + channel) >>
287     8);
288     afp = twosComplementToDecimal(readData, 24);
289     return (channel == TOTAL) ? (afp * TOTALPOWRES / FACTOR) : (afp * POWRES / FACTOR);
290 }

```

```

290  uint16_t fundamental(int8_t channel) {
291      switch (channel) {
292          case UA: case UB: case UC:
293              return read16(0x1C1 + channel * 2);
294          case IA: case IB: case IC:
295              return read16(0x1C0 + (channel - 4) * 2);
296      }
297  }
298
299  float activePower(int8_t channel) {
300      uint32_t readData = 0;
301      float ap;
302      readData = (((uint32_t)read16(0xB0 + channel + 1)) << 8) + (read16(0xC0 + channel
+ 1) >> 8);
303      ap = twosComplementToDecimal(readData, 24);
304      return (channel == TOTAL) ? (ap * TOTALPOWRES / FACTOR) : (ap * POWRES / FACTOR);
305  }
306
307  float reactivePower(int8_t channel) {
308      uint32_t readData = 0;
309      float rp;
310      readData = (((uint32_t)read16(0xB4 + channel + 1)) << 8) + (read16(0xC4 + channel
+ 1) >> 8);
311      rp = twosComplementToDecimal(readData, 24);
312      return (channel == TOTAL) ? (rp * TOTALPOWRES / FACTOR) : (rp * POWRES / FACTOR);
313  }
314
315  float aparentPower(int8_t channel) {
316      float ap;
317      ap = (((uint32_t)read16(0xB8 + channel + 1)) << 8) + (read16(0xC8 + channel + 1)
>> 8);
318      return (channel == TOTAL) ? (ap * TOTALPOWRES / FACTOR) : (ap * POWRES / FACTOR);
319  }
320
321  float powerFactor(int8_t channel) {
322      float pf;
323      pf = twosComplementToDecimal(read16(0xBC + channel + 1), 16) * PFRES;
324      return pf;
325  }
326
327  int32_t twosComplementToDecimal(uint32_t complement, const int Nbits) {
328      int32_t output;
329      output = -1;
330      output *= (int32_t)bitRead(complement, Nbits - 1);
331      output *= powOfTwo(Nbits - 1);
332      for (int8_t i = 0; i <= Nbits - 2; i++) {
333          output += ((int32_t)bitRead(complement, i)) * powOfTwo(i);
334      }
335      return output;
336  }
337
338  int32_t powOfTwo(int8_t expoent) {
339      return ((int32_t)1) << expoent;
340  }
341
342  void outputDMA(){
343
344      FsFile Medicoes_TXT = SPIFFS.open("/medicoes.txt", "w");
345      Medicoes_TXT.close();
346      Medicoes_TXT = SPIFFS.open("/medicoes.txt", "w");
347
348      if (!Medicoes_TXT) {
349          Serial.println("Erro na escrita/abertura de medicoes.txt.");
350          return;
351      }
352      if (Medicoes_TXT) {
353
354          SPI.beginTransaction(SPISettings(1000000, MSBFIRST, SPI_MODE3));
355
356          int32_t data[7];
357
358          int cont2 = 0;
359          int resolucao = 300;

```

```

360
361 SPI.transfer16(0x0E);
362 SPI.transfer16(0x7F04); //Todos as ADC serao habilitadas, dados de 32 bits
363
364 while(cont2 < resolucao){
365     digitalWrite(M0_CHIP_SELECT, LOW);
366     digitalWrite(M0_DMA_CTRL, HIGH);
367     for(int cont = 0; cont < 7; cont = cont + 1){
368         switch (cont) {
369             case 0: case 1: case 3: case 5:
370                 data[cont] = SPI.transfer16(0x0000) * 0.01;
371             case 2: case 4: case 6:
372                 data[cont] = SPI.transfer16(0x0000) * 0.001;
373         }
374         switch (cont) {
375             case 0: case 1: case 3: case 5:
376                 data[cont] += SPI.transfer16(0x0000) * 0.01/256;
377             case 2: case 4: case 6:
378                 data[cont] += SPI.transfer16(0x0000) * 0.001/256;
379         }
380     }
381     digitalWrite(M0_DMA_CTRL, LOW);
382     digitalWrite(M0_CHIP_SELECT, HIGH);
383     SPI.endTransaction();
384     Medicoes_TXT.print((float) data[0]); Medicoes_TXT.print(", ");
385     Medicoes_TXT.print((float) data[1]); Medicoes_TXT.print(", ");
386     Medicoes_TXT.print((float) data[2]); Medicoes_TXT.print(", ");
387     Medicoes_TXT.print((float) data[3]); Medicoes_TXT.print(", ");
388     Medicoes_TXT.print((float) data[4]); Medicoes_TXT.print(", ");
389     Medicoes_TXT.print((float) data[5]); Medicoes_TXT.print(", ");
390     Medicoes_TXT.print((float) data[6]); Medicoes_TXT.print(", \n");
391     cont2++;
392 }
393 Medicoes_TXT.close();
394 }
395 }
396
397
398 #endif // Encerre definições da biblioteca
399

```

```

1  /*-----UNIVERSIDADE FEDERAL DE CAMPINA GRANDE-----
2  %-----DEPARTAMENTO DE ENGENHARIA ELETRICA E INFORMATICA-----
3  %-----
4  %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
5  % Victor Hugo dos Santos Germano e-mail: victor.germano@ee.ufcg.edu.br
6  %-----
7  %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
8  % Biblioteca de funcoes para o uC
9  %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%*/
10
11
12 #ifndef uC_CONFIG // Se a biblioteca nao estiver definida...
13 #define uC_CONFIG //...define.
14
15 void IMPRIMR(); //mostra os valores
16 void OLED_START(); //iniciar tela OLED
17 void OLED_READ(); //mostrar valores no OLED
18 void detectaBotao(); //botao para iniciar sair do modo leitura
19 void handleNotFound(); //alerta de servidor nao encontrado
20 void servidor_Pag_Inicial(); //chamada da pagina web principal
21 void servidor_Pag_Tensao(); //chamada da pagina web: grafico para tensao
22 void servidor_Pag_Corrente(); //chamada da pagina web: grafico para corrente
23 void servidor_cssStyle(); //estilo da pagina web
24 void imagem_LAT(); //imagem logo do LAT na pagina web
25 void imagem_UFCG(); //imagem logo da UFCG na pagina web
26 void medicoes(); //escrita de dados em medicoes.txt
27 void parametros(); //leitura de dados em parametros.txt
28 void fileWrite(); //escrita de dados em parametros.txt
29
30 //OLED: logo do LAT
31 const unsigned char LAT_logo[] PROGMEM = {
32 };
33 //OLED: borda da tela
34 const unsigned char BORDA [] PROGMEM = {
35 };
36 //OLED: borda de botao
37 const unsigned char BOTAO_logo [] PROGMEM = {
38 };
39
40 // Funcao que envia, pela porta serial, o nome de um dado acompanhado de seu valor
41 // int selector 1 - OLED, 2 - Parametros.txt, 3 - Medicoes.txt
42 void IMPRIMR(int8_t name, String parameter, int selector) {
43 float param;
44 switch (name) {
45 case TOTAL:
46 if (parameter == "RMS") param = realRms(name);
47 else if (parameter == "P_ATIVA") param = activePower(name);
48 else if (parameter == "Q_REATIVA") param = reactivePower(name);
49 else Serial.println("Nao Valido");
50 break;
51
52 case IA: case IB: case IC:
53 if (parameter == "RMS") param = realRms(name);
54 else if (parameter == "P_ATIVA") param = activePower(name);
55 else if (parameter == "Q_REATIVA") param = reactivePower(name);
56 else Serial.println("Nao Valido");
57 break;
58
59 case UA: case UB: case UC:
60
61 if (parameter == "RMS") param = realRms(name);
62 else if (parameter == "P_ATIVA") param = activePower(name);
63 else if (parameter == "Q_REATIVA") param = reactivePower(name);
64 else Serial.println("Nao Valido");
65 break;
66
67 if (parameter == "RMS") param = realRms(name);
68 else if (parameter == "P_ATIVA") param = activePower(name);
69 else if (parameter == "Q_REATIVA") param = reactivePower(name);
70 else Serial.println("Nao Valido");
71 break;
72
73 case IN:

```

```

74
75     if (parameter == "RMS") param = real Rms(name);
76     else Serial.println("Nao Valido");
77     break;
78 }
79
80 if(selector==1){display.print(param);}
81 if(selector==2){Parametro_TXT.print(param); Parametro_TXT.print(", ");}
82 }
83
84 void OLED_START() {
85     if(!display.begin(SSD1306_SWITCHCAPVCC, 0x3C)) {
86         Serial.println(F("SSD1306 allocation failed"));
87         for(;;); // loop eterno, em caso de falha
88     }
89     delay(1000);
90     display.clearDisplay();
91     display.setTextSize(1);
92     display.setTextColor(WHITE);
93     display.drawBitmap(0, 0, BORDA, 128, 64, WHITE);
94     display.println("\n Iniciando...\n");
95     display.display();
96     estadoBotao = HIGH;
97     auxiliar = HIGH;
98 }
99
100 void OLED_READ() {
101
102     display.clearDisplay();
103
104     if (auxiliar == LOW) {
105         display.setCursor(0, 0);
106         display.print("UC: "); IMPRIMIR(UC, "RMS", 1); display.print("V ");
107         display.println("IC: "); IMPRIMIR(IC, "RMS", 1); display.print("A ");
108         display.println("PC: "); IMPRIMIR(UC, "P_ATIVA", 1); display.print("W ");
109         display.println("QC: "); IMPRIMIR(UC, "Q_REATIVA", 1); display.print("W ");
110         display.drawBitmap(71, 48, BOTAO_logo, 48, 16, WHITE);
111         display.setCursor(84, 52);
112         display.print("SAIR");
113         display.display();
114     }
115     else if (auxiliar == HIGH) {
116         display.setCursor(24, 24);
117         display.print("WATTMETRO 3W");
118         display.drawBitmap(56, 48, BOTAO_logo, 48, 16, WHITE);
119         display.setCursor(59, 52);
120         display.print("INICIAR");
121         display.display();
122     }
123     estadoBotaoAnterior = estadoBotao;
124 }
125
126 void writeFile() {
127
128     Parametro_TXT = SPIFFS.open("/parametros.txt", "w");
129     if (!Parametro_TXT) {
130         Serial.println("Erro na escrita/abertura de parametros.txt.");
131         return;
132     }
133     if (Parametro_TXT) {
134         IMPRIMIR(UA, "RMS", 2);
135         IMPRIMIR(IA, "RMS", 2);
136         IMPRIMIR(UB, "RMS", 2);
137         IMPRIMIR(IB, "RMS", 2);
138         IMPRIMIR(UC, "RMS", 2);
139         IMPRIMIR(IC, "RMS", 2);
140         IMPRIMIR(IC, "RMS", 2);
141     }
142     Parametro_TXT.close();
143 }
144
145 void detectaBotao(){
146     estadoBotao = digitalRead(START_WATTMETRO);

```

```

147     if ( ( estadoBotao == LOW && ( estadoBotaoAnterior != estadoBotao) ) {
148         auxiliar = !auxiliar;
149     }
150 }
151
152 void handleNotFound() {
153
154     String message = "File Not Found\n\n";
155     message += "URI: ";
156     message += server.uri();
157     message += "\nMethod: ";
158     message += (server.method() == HTTP_GET) ? "GET" : "POST";
159     message += "\nArguments: ";
160     message += server.args();
161     message += "\n";
162     for (uint8_t i = 0; i < server.args(); i++) {
163         message += " " + server.argName(i) + ": " + server.arg(i) + "\n";
164     }
165     server.send(404, "text/plain", message);
166 }
167
168 void servidor_Pag_Inicial() {
169     FsFile file = SPIFFS.open("/pag_inicial.html", "r");
170     server.streamFile(file, "text/html");
171     file.close();
172 }
173
174 void servidor_Pag_Tensao() {
175     FsFile file = SPIFFS.open("/tensao.html", "r");
176     server.streamFile(file, "text/html");
177     file.close();
178 }
179
180 void servidor_Pag_Corrente() {
181     FsFile file = SPIFFS.open("/corrente.html", "r");
182     server.streamFile(file, "text/html");
183     file.close();
184 }
185
186 void servidor_cssStyle() {
187     FsFile file = SPIFFS.open("/style.css", "r");
188     server.streamFile(file, "text/css");
189     file.close();
190 }
191
192 void imagem_LAT() {
193     FsFile file = SPIFFS.open("/LAT.png", "r");
194     server.streamFile(file, "image/png");
195     file.close();
196 }
197
198 void imagem_UFCG() {
199     FsFile file = SPIFFS.open("/UFCG.png", "r");
200     server.streamFile(file, "image/png");
201     file.close();
202 }
203
204 void medicoes() {
205     FsFile file = SPIFFS.open("/medicoes.txt", "r");
206     server.streamFile(file, "text/plain");
207     file.close();
208 }
209
210 void parametros() {
211     FsFile file = SPIFFS.open("/parametros.txt", "r");
212     server.streamFile(file, "text/plain");
213     file.close();
214 }
215
216
217 #endif // Encerre definições da biblioteca
218

```

```

1  <!-------UNIVERSIDADE FEDERAL DE CAMPINA GRANDE-----
2  % ----- DEPARTAMENTO DE ENGENHARIA ELETRICA E INFORMATICA -----
3  % -----
4  %%%%%%%%%%%%%%%%%%%%%%%
5  % Victor Hugo dos Santos Germano e-mail: victor.germano@ee.ufcg.edu.br
6  % -----
7  %%%%%%%%%%%%%%%%%%%%%%%
8  % Pagina Web Principal
9  %%%%%%%%%%%%%%%%%%%%%%%*/
10 -->
11
12 <!DOCTYPE html>
13 <html>
14   <head>
15     <meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1">
16     <link rel="icon" href="data:,">
17     <link rel="stylesheet" type="text/css" href="style"/>
18   </head>
19
20   <script type="text/javascript">
21     var progresso = new Number();
22     var maximo = new Number();
23     var progresso = 0;
24     var maximo = 5;
25     function start(){
26       if ((progresso + 1) < maximo) {
27         progresso = progresso + 1;
28         document.getElementById("pg").value = progresso;
29         setTimeout("start();", 1000);
30       }
31     }
32   </script>
33
34   <body onload="start();">
35
36     <div id="interface">
37       <header>
38         <section align="right">
39           
40         </section>
41
42         <section margin="-15px">
43           
44         </section>
45       </header>
46       <header id="cabecalho">
47         <h1>IoT Watt</h1>
48         <h2 class="subtitulo">LAT - UFCG</h2>
49       </header>
50
51       <section id="areaBotao">
52         <button name="voltage" onclick="parent.location='tensao' "
53           class="botaoTensao" ><b>TENSÃO</b></button>
54         <button name="current" onclick="parent.location='corrente' "
55           class="botaoCorrente" ><b>CORRENTE</b></button>
56         <button name="aboutIW" onclick="parent.location='sobre' "
57           class="botaoSobre" ><b>SOBRE</b></button>
58       </section>
59
60       <script
61         src="//ajax.googleapis.com/ajax/libs/jquery/2.1.3/jquery.min.js"></sc
62         ript>
63       <script type="application/javascript">
64
65         $(document).ready(function() {
66           atualiza();
67         });
68
69         function atualiza() {
70           delete window.html;

```

```

68     var html = [];
69     $.get('parametros', function(data) {
70         times = data.split("\n");
71         for (var i in times) {
72             html = data.split(",");
73         }
74         document.getElementById("VaRMS").innerHTML = html[0];
75         document.getElementById("IaRMS").innerHTML = html[1];
76         document.getElementById("VbRMS").innerHTML = html[2];
77         document.getElementById("IbRMS").innerHTML = html[3];
78         document.getElementById("VcRMS").innerHTML = html[4];
79         document.getElementById("IcRMS").innerHTML = html[5];
80         document.getElementById("InRMS").innerHTML = html[6];
81     });
82     setTimeout('atualiza()', 2000);
83 }
84 </script>
85
86 <section id="corpo">
87     <fieldset><legend><h1>VALORES ATUALIZADOS</h1></legend>
88     <table onload="atualiza();" id="tabela"
89         style="text-align: center">
90         <tr>
91             <td><b>Tens2o</b></td>
92             <td><b>(Vrms)</b></td>
93             <td><b>Corrente</b></td>
94             <td><b>(Arms)</b></td>
95         </tr>
96
97         <tr>
98             <td>Van</td>
99             <td id="VaRMS"></td>
100            <td>Ia</td>
101            <td id="IaRMS"></td>
102        </tr>
103
104        <tr>
105            <td>Vbn</td>
106            <td id="VbRMS"></td>
107            <td>Ib</td>
108            <td id="IbRMS"></td>
109        </tr>
110
111        <tr>
112            <td>Vcn</td>
113            <td id="VcRMS"></td>
114            <td>Ic</td>
115            <td id="IcRMS"></td>
116        </tr>
117
118        <tr>
119            <td></td>
120            <td></td>
121            <td>In</td>
122            <td id="InRMS"></td>
123        </tr>
124    </table>
125 </fieldset>
126 </section>
127 <section>
128     <footer id="rodape">
129     <p>Copyright &copy; 2019 - by LAT@UFCG</p>
130     <a
131         href="https://sites.google.com/a/dee.ufcg.edu.br/gse/laboratrio-de-alt
132         a-tenso" target="_blank">WebLAT</a></p>
133     </footer>
134 </section>
135 </div>
</body>
</html>

```

```

1  <!-------UNIVERSIDADE FEDERAL DE CAMPINA GRANDE-----
2  % ----- DEPARTAMENTO DE ENGENHARIA ELETRICA E INFORMATICA -----
3  % -----
4  %%%%%%%%%%%%%%%
5  % Victor Hugo dos Santos Germano e-mail: victor.germano@ee.ufcg.edu.br
6  % -----
7  %%%%%%%%%%%%%%%
8  % Pagina Web: graficos para tensao
9  %%%%%%%%%%%%%%%
10 -->
11
12 <!DOCTYPE html>
13 <html>
14   <head>
15     <meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1">
16     <link rel="icon" href="data:,">
17     <link rel="stylesheet" type="text/css" href="style"/>
18   </head>
19
20   <script type="text/javascript">
21     var progresso = new Number();
22     var maximo = new Number();
23     var progresso = 0;
24     var maximo = 5;
25     function start(){
26       if ((progresso + 1) < maximo) {
27         progresso = progresso + 1;
28         document.getElementById("pg").value = progresso;
29         setTimeout("start();", 1000);
30       }
31     }
32   </script>
33
34   <script
35   src="http://ajax.googleapis.com/ajax/libs/jquery/2.1.3/jquery.min.js"></script>
36   <script type="application/javascript">
37     delete medicoes_data;
38     var opcao = 0;
39     var F5Botao;
40     var medicoes_data = [];
41     $(document).ready(function() {
42       atualiza();
43     });
44     function atualiza() {
45       $.get('medicoes', function(txtdata) {
46         times = txtdata.split("\n");
47         for (var i in times) {
48           medicoes_data = txtdata.split(",");
49         }
50       });
51       setTimeout("atualiza();", 4000);
52     }
53   </script>
54
55   <body onload="start();">
56
57     <div id="interface" onload="atualiza();">
58
59       <section align="right">
60         
61       </section>
62       <section margin="-15px">
63         
64       </section>
65
66       <header id="cabecalho">
67         <h1>IoT Watt</h1>
68         <h2 class="subtitulo">LAT - UFCG</h2>
69       </header>
70
71       <section id="areaBotao">

```

```

72     <button name="voltage" onclick="parent.location='tensao' "
73     class="botaoTensao" ><b>TENSÃO</b></button>
74     <button name="current" onclick="parent.location='corrente' "
75     class="botaoCorrente" ><b>CORRENTE</b></button>
76     <button name="aboutIW" onclick="parent.location='sobre' "
77     class="botaoSobre" ><b>SOBRE</b></button>
78 </section>
79 <section id="corpo" style="height: 500px">
80     <fieldset style="height: 500px"><legend><h1>OSCILOSCÓPIO</h1></legend>
81     <form method="get" enctype="multipart/form-data">
82         <a id="va" onclick="getva()"><b> <input type="button"
83         class="botaofaseA" name="leitura" value="VA" /> </b></a>
84         <a id="vb" onclick="getvb()"><b> <input type="button"
85         class="botaofaseB" name="leitura" value="VB" />
86         </b></a>
87         <a id="vc" onclick="getvc()"><b> <input type="button"
88         class="botaofaseC" name="leitura" value="VC" /> </b></a>
89         <a id="myImage_V" style="width: 800px; height: 600px;"></a>
90     </form>
91 </fieldset>
92 </section>
93
94 <script type="text/javascript"
95 src="https://www.gstatic.com/charts/loader.js"></script>
96 <script>
97     google.charts.load('current', {'packages':['corechart']});
98     google.charts.setOnLoadCallback(getva);
99     function getva() {
100         opcao = 1;
101         var aux = [];
102         var cont = 0;
103         for(var cont2 = 0 + 2; cont2 + 2 < medicoes_data.length; cont2 =
104         cont2 + 7){
105             aux.push([cont, parseFloat(medicoes_data[cont2])]);
106             cont = cont + 0.000125; //0.000125 eh o periodo de
107             amostragem do DMA
108         };
109
110     var data = new google.visualization.DataTable();
111     data.addColumn('number', 'tempo(s)');
112     data.addColumn('number', 'Va(V)');
113
114     data.addRow(aux);
115     delete aux;
116     var options = {
117         width: 600,
118         height: 300,
119         hAxis: {
120             title: 'Tempo (s)'
121         },
122         vAxis: {
123             title: 'Tens²o (V)',
124             viewWindow: {
125                 max: 700,
126                 min: -700
127             }
128         },
129         colors: ['#db1818']
130     };
131
132     var chart = new
133     google.visualization.LineChart(document.getElementById('myImage_
134     V'));
135     chart.draw(data, options);
136     atualizaBotao();
137 }
138 </script>

```

```

132 <script>
133     google.charts.load('current', {'packages':['corechart']});
134     google.charts.setOnLoadCallback(getvb);
135     function getvb() {
136         opcao = 2;
137         var aux = [];
138         var cont = 0;
139         for(var cont2 = 0 + 4; cont2 + 4 < medicoes_data.length; cont2 =
140             cont2 + 4){
141             aux.push([cont, parseFloat(medicoes_data[cont2])]);
142             cont = cont + 0.000125;
143         };
144
145         var data = new google.visualization.DataTable();
146         data.addColumn('number', 'tempo(s)');
147         data.addColumn('number', 'Vb(V)');
148
149         data.addRow(aux);
150         delete aux;
151         var options = {
152             width: 600,
153             height: 300,
154             hAxis: {
155                 title: 'Tempo (s)'
156             },
157             vAxis: {
158                 title: 'Tens2o (V)',
159                 viewWindow: {
160                     max: 700,
161                     min: -700
162                 }
163             },
164             colors: ['#2bc633']
165         };
166
167         var chart = new
168             google.visualization.LineChart(document.getElementById('myImage_
169             V'));
170         chart.draw(data, options);
171         atualizaBotao();
172     }
173 </script>
174
175 <script>
176     google.charts.load('current', {'packages':['corechart']});
177     google.charts.setOnLoadCallback(getvc);
178     function getvc() {
179         opcao = 3;
180         var aux = [];
181         var cont = 0;
182         for(var cont2 = 0 + 6; cont2 + 6 < medicoes_data.length; cont2 =
183             cont2 + 6){
184             aux.push([cont, parseFloat(medicoes_data[cont2])]);
185             cont = cont + 0.000125;
186         };
187
188         var data = new google.visualization.DataTable();
189         data.addColumn('number', 'tempo(s)');
190         data.addColumn('number', 'Vc(V)');
191
192         data.addRow(aux);
193         delete aux;
194         var options = {
195             width: 600,
196             height: 300,
197             hAxis: {
198                 title: 'Tempo (s)'
199             },
200             vAxis: {
201                 title: 'Tens2o (V)',
202                 viewWindow: {
203                     max: 700,

```

```

200         min: -700
201     }
202 },
203     colors: ['#5b187c']
204 };
205
206     var chart = new
google.visualization.LineChart( document.getElementById('myImage_
V'));
207     chart.draw(data, options);
208     atualizaBotao();
209 }
210 </script>
211
212 <script type="application/javascript">
213     function atualizaBotao() {
214         delete data; delete chart;
215         if(opcao == 1){opcao = 0; clearTimeout(F5Botao); F5Botao =
setTimeout("getva();", 4000);}
216         if(opcao == 2){opcao = 0; clearTimeout(F5Botao); F5Botao =
setTimeout("getvb();", 4000);}
217         if(opcao == 3){opcao = 0; clearTimeout(F5Botao); F5Botao =
setTimeout("getvc();", 4000);}
218     }
219 </script>
220
221
222 <button type="submit" name="back"
onclick="parent.location='pag_inicial'" class="botaoSair"
><b>SAIR</b></button>
223 <footer id="rodape">
224     <p>Copyright &copy; 2019 - by LAT@UFCG</p>
225     <a
href="https://sites.google.com/a/dee.ufcg.edu.br/gse/laboratorio-de-alt
a-tenso" target="_blank">WebLAT</a></p>
226 </footer>
227
228 </div>
229 </body>
230 </html>
231

```