

**Programa Institucional de Bolsas de Iniciação Científica – PIBIC
RELATÓRIO PARCIAL**

IDENTIFICAÇÃO DO PROJETO

Título do Projeto: Automação De Sistemas Elétricos De Potência Com Dispositivos Eletrônicos Inteligentes
 Local de Realização (Unidade/Instituto/Departamento/Laboratório): Escola de Engenharia / Universidade Federal Fluminense / Departamento de Engenharia Elétrica / Laboratório de Eletrônica
 Endereço: Rua Passo da Pátria, 156 – Bloco E, Sala 420A
 Bairro: São Domingos Cidade: Niterói UF: RJ CEP: 24210-240

DADOS DO ORIENTADOR

Nome: Rainer Zanghi
 Matrícula Siape: 2393115 CPF: 034205097-40
 Endereço: Rua Passo da Pátria, 156 – Bloco D, Sala 421
 Bairro: São Domingos Cidade: Niterói
 UF: RJ CEP: 24210-240 E-mail: rzanghi@id.uff.br
 Telefone 1: (21) 2629-5471 Telefone 2: ()

DADOS DO BOLSISTA

Nome: Lucas Abdalla Menezes
 Matrícula: 116.038.019 CPF: 171.509.847-11 CR: 6.9
 Curso/Departamento/Instituto: Engenharia Elétrica/ Departamento de Engenharia Elétrica – TEE/ Universidade Federal Fluminense - UFF
 Endereço: Rua Santa Rosa 91, Apto. 906
 Bairro: Santa Rosa
 Cidade: Niterói UF: RJ CEP: 24240225
 Email: lucasabdalla@id.uff.br Telefone 1: (21)971475033

1. TÍTULO

AUTOMAÇÃO DE SISTEMAS ELÉTRICOS DE POTÊNCIA COM DISPOSITIVOS ELETRÔNICOS INTELIGENTES

2. INTRODUÇÃO

Non-Intrusive Load Monitoring (NILM) ou Desagregação de Cargas — do inglês Load Disaggregation —, é o nome dado ao reconhecimento das cargas operantes de um circuito, por meio da análise das curvas de corrente e tensão, que são entregues a ele [1].

As discussões acerca de métodos para a conscientização e redução do desperdício de energia elétrica, são crescentes no contexto atual. Com o foco na área de eficiência energética e estudo de Smart Grids (redes elétricas inteligentes), esse projeto estuda a possibilidade de implementação de tecnologias de baixo custo, que sejam capazes de monitorar a rede de maneira não intrusiva. Esta pesquisa, diferentemente de [2], [3] trata-se de um estudo que foca na capacidade de dispositivos computacionais econômicos (normalmente micro controladores), de reconhecer e agrupar diferentes sinais presentes na rede.

Como mostrado por [4], os consumidores finais, mostram-se mais sensíveis a mudanças no hábito de consumo, quando conseguem visualizar seus gastos em tempo real, ao invés da maneira tradicional, a cada 30 dias. Por esses motivos, o uso de técnicas de NILM, mostra-se adequado para essa visualização de dados.

Segundo [5], existem diversos algoritmos e abordagens para realizar a desagregação das cargas, e esses podem ser divididos em quatro etapas. A primeira, consiste na detecção dos eventos. Com base nas variações da corrente que alimenta o circuito, pode-se identificar quando uma carga troca de estado Ligado para Desligado, ou vice-versa. Na segunda etapa, após a identificação de um novo evento, o sinal de corrente começa a ser armazenado e avaliado, até o término do evento. Considera-se o fim do evento quando é atingido um novo estado de regime permanente. Na terceira, é feita uma triagem desses eventos; onde aqueles com características similares, enquadram-se em um grupo, e eventos do mesmo grupo, correspondem à mesma carga. Por fim, é mostrado para o consumidor a energia consumida por cada carga em tempo real.

Nesse contexto, na primeira parte da pesquisa, descrita nesse relatório parcial, serão detalhadas as etapas de elaboração do ambiente de testes que será usado para as medições; criação do esquema do circuito para medição da corrente instantânea da rede; e desenvolvimento de um código para cálculo de seu valor com disponibilização em rede Ethernet para sua futura utilização na classificação das cargas.

3. METODOLOGIA

Etapa 1:

A primeira etapa do projeto foi destinada ao estudo de bibliografias recomendadas pelo orientador, sobre alguns aspectos iniciais que viriam a ser encontrados durante o desenvolvimento da pesquisa. Foram explorados os seguintes temas em artigos científicos, Teses e trabalhos de conclusão de curso de graduação: Técnicas para medição de corrente de maneira não intrusiva em sistemas de distribuição de energia elétrica e tipos de algoritmos da inteligência computacional para identificação de padrões. Após uma pesquisa feita em portais de publicações científicas (Springer Link, Science Direct, IEEE Xplore e Periódicos Capes), foram encontrados 814 artigos de conferências e revistas sobre a Desagregação de Cargas, publicados desde 1992. Mesmo com esse número expressivo de publicações sobre o tema, pouco foi discutido sobre a implementação de algoritmos e técnicas NILM para sistemas embarcados. Usando essas mesmas plataformas de pesquisa, foram encontrados apenas 117 trabalhos com as palavras chave "NILM" e "embedded" no contexto das Engenharias (filtrando com a palavra chave "-cytology"). Sendo os artigos [1] – [6], os mais relevantes para essa parte inicial do estudo.

Etapa 2:

Nessa etapa, foi iniciado o desenvolvimento do artigo sobre o uso de sistemas embarcados de baixo custo para o monitoramento não intrusivo de cargas ou NILM (do inglês Non-Intrusive Load Monitoring). Além do estudo de outras bibliografias para investigar os requisitos mínimos de software e hardware do sistema para o tratamento dos dados, de maneira que a proposta de reconhecimento de cargas não fosse afetada.

Etapa 3:

Na terceira etapa, foram estudadas plataformas computacionais de baixo custo disponíveis no mercado e sua aderência aos requisitos mínimos de software e hardware escolhidos na etapa anterior. Foi feito também, o projeto do circuito para medições de corrente, incluindo a seleção de componentes para montagem do circuito.

Etapa 4:

Nessa etapa, foi feita a montagem de um protótipo do circuito projetado, e iniciada a elaboração de um código que possa ser reaproveitado para os demais dispositivos que serão testados. Esse código é responsável por fazer a aferição de corrente de uma malha do circuito, e então faz o cálculo do seu valor RMS para cada instante, dado um certo tempo de amostragem.

4.RESULTADOS

4.1 Elaboração do ambiente de testes.

Após estudados métodos de extração dos valores de corrente, o primeiro passo foi montar uma esquemática do ambiente de testes. Na Figura 1, está ilustrado um diagrama com os elementos utilizados no ambiente de testes.

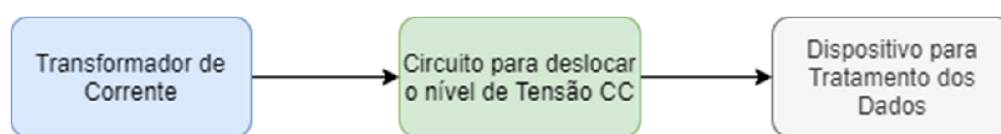


Figura 1: Diagrama do ambiente de testes

O esquema funciona da seguinte forma: Uma corrente atravessa o Transformador de Corrente (TC), que por sua vez, produz uma tensão no seu enrolamento secundário, que é proporcional ao fluxo enlaçado gerado por essa corrente. Essa tensão está contida no intervalo de $[-1,1]$ V. O micro controlador utilizado suporta apenas valores de tensão positivos entre 0 e $+3,3$ V na sua porta de leitura para o conversor analógico -digital (ADC). Portanto, foi criado um circuito, que fosse capaz de produzir um deslocamento (offset) do nível de tensão CC, que garanta que essa tensão seja estritamente positiva, e dentro destes limites. Nesse caso, considerando a maior variação possível do sensor, ficará entre $+0,65$ e $+2,65$ V. Depois disso, foi dado início ao desenvolvimento de um código que fosse capaz de ler a tensão na porta do dispositivo computacional (porta 34, no nosso caso), e que fizesse a conversão desse valor para equivalência em Amperes RMS que atravessa o TC. A descrição dos três elementos será feita nas seções a seguir.

4.1.1 Um sensor de corrente não-intrusivo - transformador de corrente (TC) tipo janela.

O TC tipo janela utilizado no protótipo está ilustrado na *Figura 2*. Com este instrumento, é possível medir a corrente elétrica em um circuito de maneira não-intrusiva, evitando a interrupção do circuito para instalação do medidor.



Figura 2: TC tipo janela não intrusivo (disponível em [13])

O esquema de funcionamento deste TC é ilustrado na *Figura 3*.

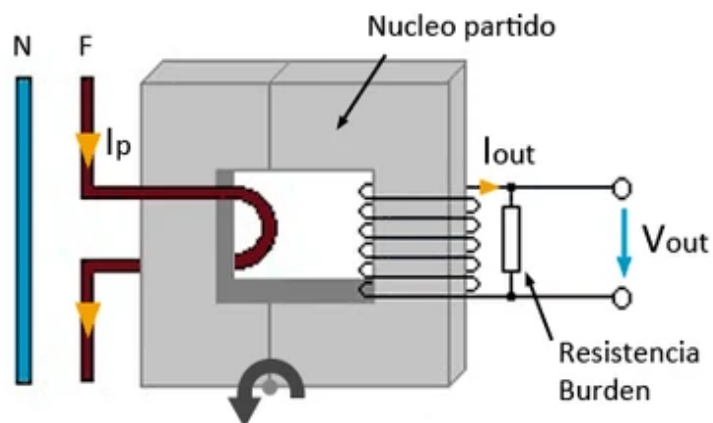


Figura 3: Esquema de funcionamento do TC (disponível em [13])

A relação de transformação do TC, pode ser descrita por (1).

$$\frac{I_r}{I_p} = \frac{V_p}{V_s} = \frac{N_p}{N_s} \quad (1)$$

Onde,

- I_r é a corrente produzida no enrolamento secundário;
- I_p é a corrente correspondente ao número de condutores que atravessam o núcleo magnético do TC;
- V_p é a tensão no terminal primário;
- V_s é a tensão produzida na Resistência Burden;
- N_p/N_s , representa a relação de transformação;

A tensão de saída fornecida pelo TC, que é proporcional à corrente que o atravessa, é entregue por meio de um conector P2 de 3,5 mm, que é muito comum no campo de áudio [13]. Onde a tensão de saída é dada pela diferença de potencial entregue pelos contatos, “Ponta” e “Capa” do conector ($V_{(3)} - V_{(1)}$), representada pela Figura 4.

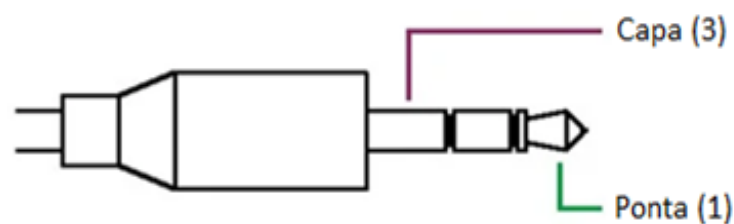


Figura 4: Conector P2 de 3,5 mm (disponível em [13])

4.1.2 Circuito de condicionamento do sinal do sensor – deslocamento de nível CC (Corrente Contínua).

Uma vez que a maioria dos microcontroladores só suporta tensão CC na sua entrada, para a criação do circuito, é necessário saber qual a corrente máxima que o TC é capaz de suportar sem que a sua saturação magnética seja atingida, a fim de tornar o sinal de saída estritamente positivo. Com os dados nominais do TC, temos que para uma corrente máxima de 30A, esse produz uma tensão de 1V. Dessa forma, foi montado um circuito que gera, pelo menos, 1Volt de Offset do sinal de entrada, já que essa seria a tensão máxima que o TC consegue reproduzir.

Seguindo [13] e [14], foi montado o circuito da Figura 5.

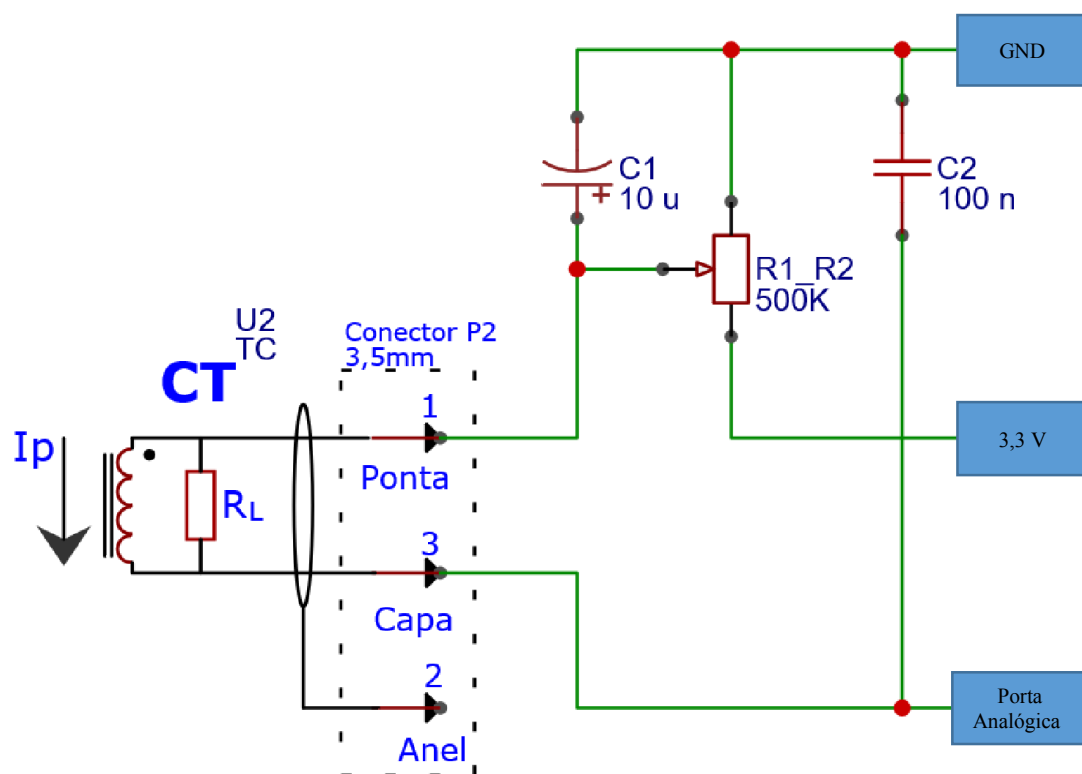


Figura 5: Esquema do circuito para o deslocamento CC do Sinal

O circuito foi alimentado com uma tensão de (3,3 V c.c.) para um offset de 1,65V (3,3 / 2 V). A tensão de saída V_{tc} é exatamente a tensão produzida pelo secundário do TC, deslocada de 1,65 V positivamente. Essa portanto, assume valores de (1,65 – 1) V até (1,65V +1) V, definindo o intervalo entre 0,65 V e 2,65 V. O valor de tensão na entrada do ADC dependerá do valor instantâneo de corrente demandado pela carga.

Os componentes utilizados neste circuito são:

- 1 Trimpot de 500kOhms, separada igualmente entre R1 e R2;
- 1 Capacitor de 10 uF / 50 V para filtrar sinais de alta frequência que chegam no circuito;
- 1 Capacitor de 100 nF / 250 V para filtrar sinais de alta frequência que chegam no dispositivo computacional;
- 1 Conector P2 para receber o sinal de saída do TC;
- 1 Fonte CC de 3,3 V proveniente da própria placa de desenvolvimento do dispositivo;

O circuito projetado foi montado em um protótipo, utilizando uma matriz de contatos (protoboard) e cabos conectorizados (jumpers) para as conexões. A Figura 6 ilustra o protótipo do circuito de condicionamento de sinais montado.

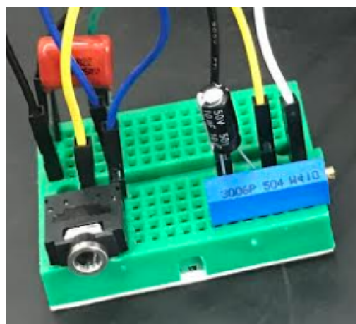


Figura 6: Circuito para o deslocamento CC do Sinal

4.1.3 Uma plataforma computacional de baixo custo que será responsável por receber e tratar o sinal de saída do circuito montado.

Para início da bancada de testes, foi utilizado um ESP32 [15]. pelas suas características um pouco mais robustas que outros sistemas computacionais de baixo custo na mesma faixa de preço como o Arduino UNO [16]. Para as principais diferenças entre os dispositivos, tendo como base o Datasheet dos fabricantes [15] e [16], apresenta-se uma comparação entre as características mais relevantes na Tabela 1.

Tabela 1: Comparação entre Arduino - ESP32

Descrição	Arduino UNO	ESP32
Pinos de I/O	23 pinos com 6PWM	34 pinos com 16 PWM
Memória RAM/SRAM	2 kB	520 kB
Temporizadores	3 timers de 16-bits e dois de 8bits	4 timers de 64-bits
Memória Flash	32 kB	4 MB
Memória ROM/EEPROM	4 kB	448 kB
Interface Ethernet Wi-fi	Não possui	Possui
Frequência de Operação	0 – 16 MHz	80 – 240 MHz
Preço (estimativa em fev. 2020)	R\$70,00	R\$40,00

Além da capacidade de processamento e memória, outro critério importante na escolha do ESP32 para esta solução foi a verificação nas referências bibliográficas [7]-[10] indicadas pelo professor, da inserção de dispositivos de IoT (Internet of Things) na automação de sistemas elétricos de potência. Segundo a norma IEC61850 [11], padrão utilizado para a automação de sistemas elétricos de potência, todos os dispositivos devem se comunicar através de uma interface Ethernet. Para a plataforma Arduino UNO, a interface Ethernet só é possível através de placas de circuito adicionais (shields).

Uma foto do dispositivo ESP32 utilizado, montado em uma placa de desenvolvimento ESP32 DEVKIT V4 [16] se encontra na Figura 7.



Figura 7: ESP32 DEVKIT V4

Na Figura 8, foram juntadas as três partes abordadas anteriormente, e postas num diagrama.

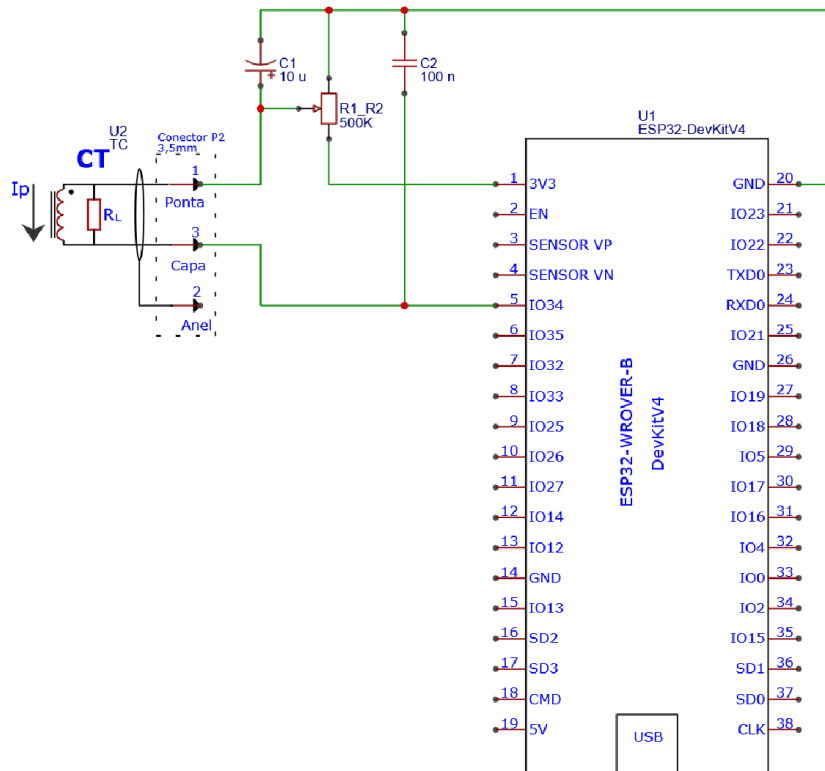


Figura 8: Diagrama do Ambiente de testes completo

Na Figura 9, é mostrada a montagem completa do diagrama descrito na Figura 8.



Figura 9 -Montagem do protótipo

4.2 Produção de um código para o ESP32 que faça a leitura da corrente que passa pelo TC.

4.2.1 Idealização do Código

Começando o desenvolvimento do código, o desafio se dá em identificar o quão rápido pode ser a frequência de amostragem, uma vez que o código seja escrito da forma mais otimizada possível.

O ESP32, consegue receber códigos de *Micro-Python* [12] ou *C/C++*, como os demais microcontroladores do mercado. Utilizando *Micro-Python*, foi verificado que esse era muito inconstante e pouco confiável para tempos exatos de amostragem. Um simples código como o descrito no pseudo-código a seguir, gerava valores muito distintos para cada valor da variável `TEMPO_DA_ITERACAO`, variando de [200, 500] *us*. Isso ocorre, por falta de otimização na conversão de *Micro-Python* para C e pela ausência de implementação de interrupções para esta plataforma computacional em *Micro-Python*.

```
VALORES_LIDOS = [];  
TEMPO_PERCORRIDO = 0;  
TEMPO_DA_ITERACAO;  
PARA I de 0 ATÉ 255 FAÇA PASSO 1  
    VALORES_LIDOS[I] = LER_PORTA_34();  
    TEMPO_DA_ITERACAO = TEMPO_ATUAL - TEMPO_PERCORRIDO;  
    ESCREVER(TEMPO_DA_ITERACAO);  
    TEMPO_PERCORRIDO = TEMPO_ATUAL;  
FIM PARA;
```

O mesmo código anterior escrito em C, ainda que gerando grandes variações, mostrou-se mais eficiente, sempre entregando valores dentro do intervalo de $[50, 200]\mu s$. Para a identificação das cargas, os intervalos entre amostras devem ser os mesmos, para que esses não interfiram no resultado. Dessa forma, foi necessário a utilização de outra ferramenta para aquisição dos dados, além de uma simples iteração usando FOR.

Para isso, foi utilizado um timer/cronômetro do próprio ESP32, o uso de Interrupção e Região Crítica do código. O timer é responsável simplesmente por armazenar o tempo decorrido desde que o dispositivo foi ligado. A interrupção faz como que o processador pause a sua tarefa atual, e a atenda o dispositivo que chamou a interrupção. Já a Região Crítica de um código, garante que apenas aquele trecho de código, seja executada naquele instante, tornando atômica a operação sobre recurso compartilhado. A programação foi feita na IDE Arduino, pois esta IDE possui todas as funções para ESP32, desenvolvidas pela própria ESPRESSIF (fabricante do ESP32), seguras para uso em interrupções.

Então, esse código funcionaria da seguinte forma: O ESP32 é ligado, o timer é disparado, e uma função de Interrupção é associada ao timer, para que a cada $100\mu s$ essa função seja chamada. Quando essa é chamada, o código entra numa Região Crítica, e garante que apenas a leitura da porta ligada ao ADC seja executada naquele instante, otimizando o processo de leitura. Depois, que alguns valores fossem armazenados, uma função para o cálculo RMS dos valores obtidos seria chamada, e o timer reiniciado para recomeçar o processo. Além disso, o último valor RMS calculado fica disponível num servidor web, hospedado no próprio ESP32, e que pode ser acessado por protocolo HTTP.

Além disso, os valores gerados pelo conversor A/D do ESP32, variam de 0 a 4095, e esses, correspondem à uma tensão CC que também varia de 0 a 3,3 V. Dessa forma o nosso código, deve fazer essa conversão do valor em contagens fornecido pelo conversor A/D para Amperes. Utilizando a relação do transdutor de 0-30 A para 0-1 V e (1), obtemos (2) para o cálculo da corrente instantânea que atravessa o TC.

$$I_P = \left(\frac{Counts}{Counts_{AD_{CC}}} - 1 \right) \cdot \left(\frac{I_{MÁX_{TC}}}{n} \right) \cdot V_{OFFSET_{CC}} \quad (2)$$

Onde:

- Counts: É o valor lido pela entrada analógica e convertido pelo conversor A/D;
- Counts_ad_cc: É o valor convertido pelo conversor A/D para a corrente de entrada do TC em 0;
- I_máx_tc: É o valor máximo de corrente medido pelo TC, especificado por [13];
- n: Constante que representa o número de condutores no interior do TC. Este recurso deve ser utilizado somente nos casos onde a corrente máxima medida é muito menor que o valor máximo de corrente do TC;
- V_offset_cc: É o valor de Tensão fornecido pelo circuito de deslocamento de nível de tensão.

4.2.2 Produção do código

A fim de otimizar e permitir o bom funcionamento do código, os seguintes artifícios foram usados:

- Definir constantes com **#define**, para reservar um espaço em memória que não precisa ser acessado.
- Declaração dos timers que serão utilizados. O **timer0** é responsável por pegar as amostras a cada 100 us e armazená-las num vetor de 256 espaços. O **timer1** é responsável por resetar o **timer0**, a cada 500 ms.
- Declaração das variáveis com o modificador **volatile**, para evitar otimizações indesejadas do compilador. E uso de **DRAM_ATTR** para garantir que a variável vá para a memória RAM, que tem acesso mais rápido que a Flash.
- Declaração de um semáforo que identifica se no momento está ocorrendo um evento de interrupção. Uso de **uint32_t**, para garantir 32 bits de memória para a variável.
- Associação de uma função ao **timer0** que é chamada a cada 100 us. Associando uma função ao **timer1** para disparar o **timer0**, que é chamada a cada 500 ms. O esquema fica como demonstrado na Figura 10.

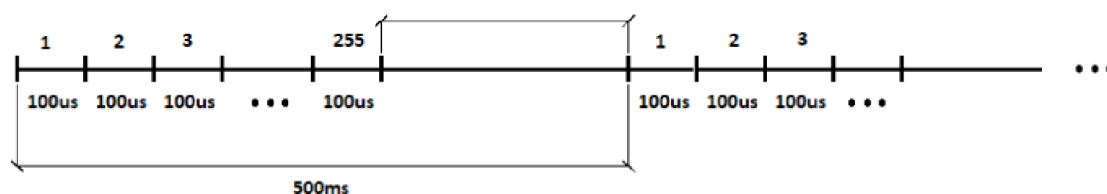


Figura 10: Funcionamento dos Timers para a leitura do sinal

- Criação de uma função que é chamada pela Interrupção, e entra em uma Região Crítica para fazer a leitura da porta 34 do controlador.
- Criação de uma função que é chamada pela Interrupção, a cada 500 ms para resetar o **timer0**.
- Função que retorna o último valor RMS calculado, ao acessar https://IP.DO.DISPOSITIVO/i_rms.

Em termos de pseudocódigo para a leitura do último valor rms, temos:

```

importar <WebServer>;

declarar timer0, timer1: hw_timer_t ;
declarar ADC_SAMPLES, I_RMS_VEC_SIZE: INT;
declarar isrCounter: uint32;
declarar i_rms_data: float;
declarar adc_data[]: uint32;
...
//função chamada pelo timer 0
funcao onTimer
    Entrada_na_zona_critica();
    adc_data[isrCounter] = leitura_porta_analogica(34);
    isrCounter++;
    Saida_da_zona_critica();
fim funcao
funcao onTimer1
    resetarTimer(timer0);
fim funcao

funcao calcularRms : float
...
retorna valorRms;
fim funcao
//Primeiro método chamado
funcao Setup
    Conectar.Wifi(login, senha);
    Definir.endpoints();
    AssociarFucoesAosEndpoints();
    Iniciar.servidor();
    Iniciar_timer0();
    Associar_leitura_a_interrupcao_do_timer0();
    Associar_chamada_do_timer0_ao_timer1();
fim Setup
funcao handleAcessoI_rms
    i_rms = calcularRms(buffer_adc_data[]);
    Servidor.enviar(i_rms);
fim funcao
//chamada após o Setup até o desligamento do dispositivo
funcao Loop
    se (timer_ja_disparado())
        entrar_em_zona_critica();
        isrCount = isrCounter++;
        sair_da_zona_critica();
    fim se
    se (isrCount>ADC_SAMPLES-1) E (timer0 ativo)
        parar_timer_0();
        entrar_em_zona_critica();
        isrCounter = 0;
        para (int i = 0; i < numero_maximo_de_amstras; i++)
            buffer_adc_data[i] = adc_data[i];
        fim para
        sair_da_zona_critica();
    fim se
    verificar_requisicoes_pro_servidor();
fim Loop

```

Após inserido o código compilado no ESP32, ao acessar <http://IP>, obtemos a última janela de valores completa usada para o cálculo do valor RMS. Acessando http://IP/i_rms, é obtido o valor RMS, para uma janela de intervalo de $100 * 255 \text{ us}$. E esse valor é atualizado a cada 500 ms . Além disso, utilizando o endpoint http://IP/i_rms_data, é possível obter os últimos 512 valores, RMS calculados.

Foi produzido então, um código em OCTAVE, por conta da praticidade, para uma primeira visualização dos gráficos, e confirmar se as curvas obtidas condizem com a variação das cargas do circuito.

Para testar o bom funcionamento, o TC foi colocado em uma das fases no quadro geral, monitorando a corrente das cargas ligadas a ele.

Acessando <http://IP>, foram passados para um gráfico os valores obtidos, ficando como mostrado na Figura 11. Os valores presentes no eixo y representam as contagens fornecidas pelo conversor A/D para uma amostra realizada em laboratório e são mostradas na Figura 11 apenas para fins de ilustração.

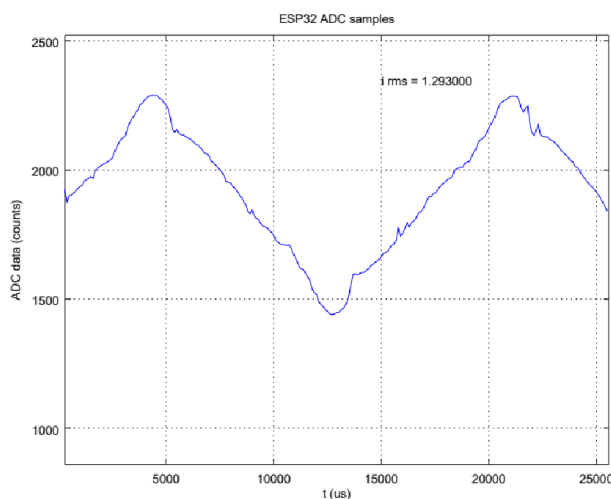


Figura 11: Exemplo de Curva de Corrente

Para testar a mudança de valores de corrente RMS com a variação de cargas residenciais, foram testados um micro-ondas e uma máquina de lavar em um circuito de distribuição monofásico de uma residência.

Primeiro, o endereço http://IP/i_rms_data foi acessado, com as cargas que serão monitoradas desligadas como na Figura 12.

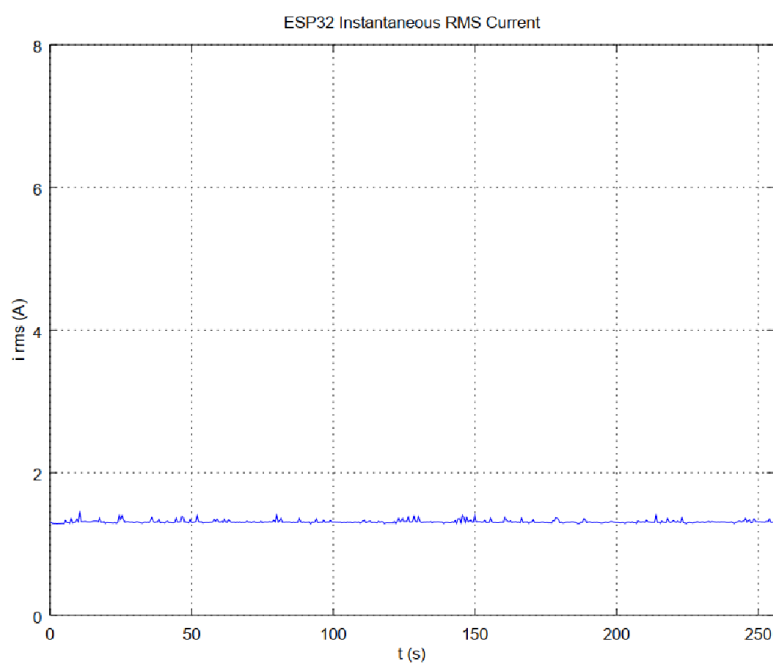


Figura 12: RMS no tempo para Micro-ondas e Máquina de lavar desligados

Logo em seguida foi iniciada a rotina de eventos de teste que teria as seguintes etapas e o tempo em que elas ocorreram:

- 1- Ligando a Máquina de lavar com o Micro-ondas desligado – $t = 0$ s;
- 2- Ligando a Micro-ondas – $t = 60$ s;
- 3- Desligando e Ligando o Micro-ondas rapidamente – $t = 160$ s;
- 4- Desligando e Ligando o Micro-ondas rapidamente – $t = 240$ s ;

Os resultados encontrados estão ilustrados na Figura 13.

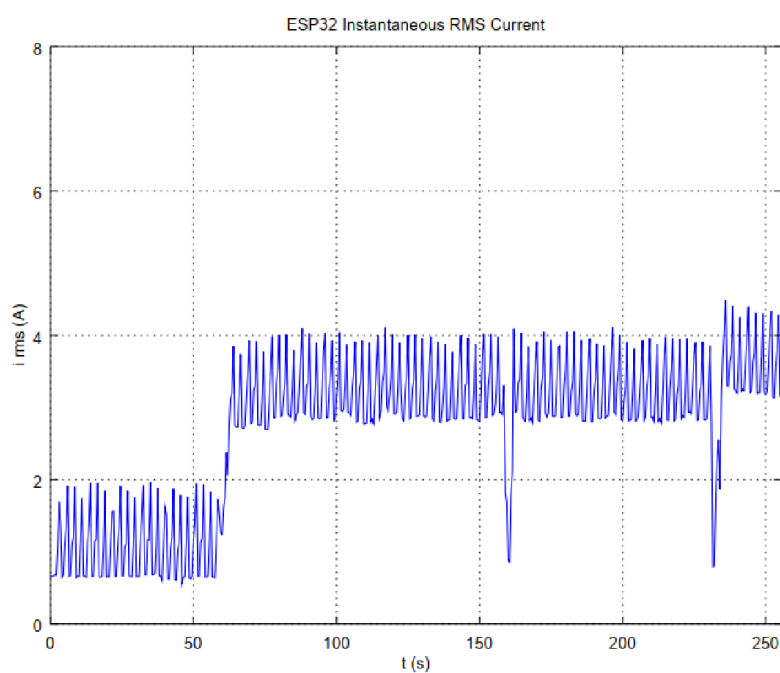


Figura 13: Parte 1 rotina de testes

Continuando os Testes anteriores:

- 5- Desligando e Ligando o Micro-ondas – $t = 50$ s;
- 6- Desligando e Ligando o Micro-ondas – $t = 120$ s;
- 7- Desligando o Micro-ondas – $t = 170$ s;
- 8- Desligando a Máquina de Lavar – $t = 190$ s;

Os resultados encontram-se na Figura 14.

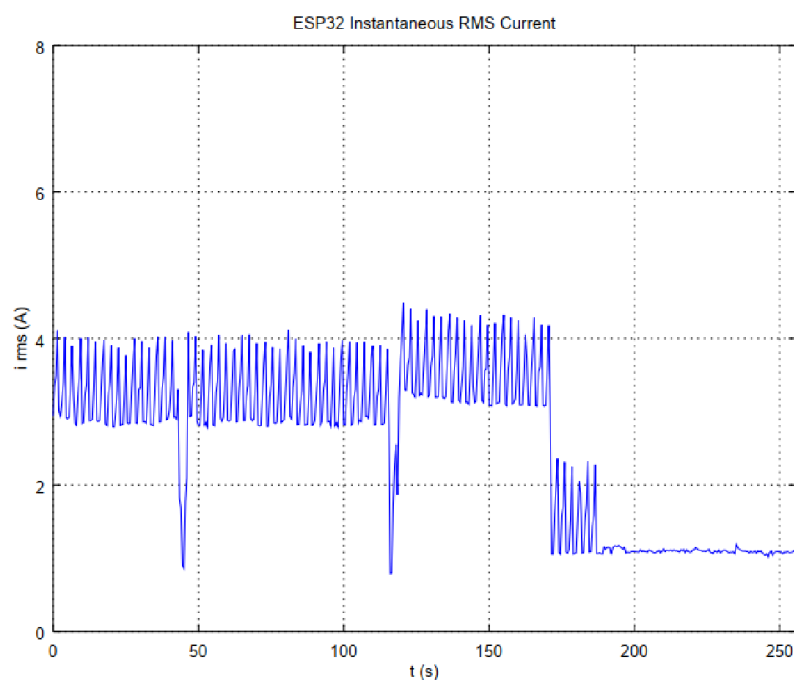


Figura 14: Parte 2 da rotina de testes

5. PRODUÇÃO TÉCNICO-CIENTÍFICA

Como citado na segunda etapa da parte de Metodologia desse documento, está sendo redigido um Artigo Científico, a partir de pesquisas feitas em base de dados como IEEE Xplore [17]. Esse artigo tem como intuito testar o funcionamento do monitorador de cargas que está sendo produzido, e a partir dos resultados encontrados, sugerir futuras aplicações da técnica.

6. CONCLUSÕES

O objetivo da primeira parte do projeto, era de levantamento bibliográfico sobre métodos de medição de corrente não intrusivos, levantamento das possíveis plataformas disponíveis no mercado, leitura de sinais em micro controladores, e métodos para o tratamento de dados, considerando as limitações técnicas dos dispositivos.

Para a medição de corrente foi necessária a criação de um circuito para a leitura dos dados pelo ESP32. Logo no início da elaboração do código que seria inserido no dispositivo, foram encontradas limitações e dificuldades de processamento e acesso à memória, que dispositivos maiores, como computadores de mesa, não teriam.

Encontrados esses problemas, foi necessário ser feito mais estudos, sobre técnicas que otimizassem o processo de leitura, tornando-a eficiente e confiável. Após a conclusão da montagem do circuito para o deslocamento de sinal CC, que viria do TC, e finalização do código para o dispositivo, foi criado um pequeno script em OCTAVE, para visualização dos gráficos de corrente, e a variação seu valor RMS no tempo. Concluídas essas etapas, é esperado que o sinal que chega ao dispositivo seja confiável.

Nessa última parte, serão utilizadas as abordagens de identificação de sinais, estudadas no início do projeto, para a implementação de um modelo computacional, que seja capaz de classificar as cargas ligadas ao sistema, bem como diferenciá-las entre si.

7. REFERÊNCIAS BIBLIOGRÁFICAS

- [1] **Hart, G. W.** (1992). Non-Intrusive appliance Load Monitoring, Proceedings of the IEEE.
- [2] **Pereira, E. L., C. Cavalieri, D. e Resende, C. Z.** (2017). Método de identificação de cargas elétricas as técnicas Short-Time Fourier e Kernel PCA, XIII Simpósio Brasileiro de automação Inteligente
- [3] **Wild, B., Barsim, K. S. e Yang, B.** (2015). A New Unsupervised Event Detector for Non-Intrusive Load Monitoring, Global Symposium on Signal Processing Applications in Smart Buildings
- [4] **Darby, S.** (2006). The effectiveness of feedback on energy consumption, DEFRA Journal of the Literature on Metering, Billing and direct Displays, vol. 486
- [5] **Azzini, H. A. D., Torquato, R. and da Silva, L.C. P.** (2014). Event Detection Methods for Non-Intrusive Load Monitoring, IEEE
- [6] **Meziane, M. N., Ravier, P., Lamarque, G., Bune-tel, J.-C. L. and Raingeaud, Y.** (2017). HighAccuracy Event Detection for Non-IntrusiveLoad Monitoring, IEEE.
- [7] **JPEMBEDDED**, “Integrating IoT with IEC 61850”, disponível em:
<https://www.jpembedded.eu/en/integrating-iot-with-iec-61850>
(Acessado em 07 de fevereiro de 2020)
- [8] **Kok Tong Lee, Hou Kit Mun** (2019) Real-time power monitoring using field-programmable gate array with IoT technology, IET Journals Vol. 13
- [9] **Dragan Mlakić, Srete Nikolovski, Hamid Reza Baghaee** (2019) An Open-Source Hardware/Software IED based on IoT and IEC 61850 Standard, 2nd International Colloquium on Smart Grid Metrology (SMAGRIMET)
- [10] **Markel Iglesias-Urkia, Diego Casado-Mansilla, Simon Mayer, Josu Bilbao, and Aitor Urbieto** (2019) Integrating Electrical Substations Within the IoT Using IEC 61850, CoAP, and CBOR, IEEE IoT Journal

-
- [11] **IEC 61850-SER** (2018), Communication Networks and Systems in Substations
- [12] **MicroPython ORG**, “Getting started with MicroPython on the ESP32”, disponível em:
<https://docs.micropython.org/en/latest/esp32/tutorial/intro.html>
(Acessado em 07 de fevereiro de 2020)
- [13] **YHDC**, “SENSOR NÃO INVASOR: YHDC SCT013-000 CT USADO COM ARDUINO. (SCT-013)”, disponível em:
<https://www.poweruc.pl/blogs/news/non-invasive-sensor-yhdc-sct013-000-ct-used-with-arduino-sct-013>.
(Acessado em 07 de fevereiro de 2020)
- [14] **LEARN OPEN ENERGY MONITOR**, “CT Sensors - Interfacing with an Arduino”, disponível em:
<https://learn.openenergymonitor.org/electricity-monitoring/ct-sensors/interface-with-arduino>.
(Acessado em 9 de fevereiro de 2020)
- [15] **ESPRESSIF**, “ESP32 DEVKIT V4 Hardware Overview”, disponível em:
<https://www.espressif.com/en/products/hardware/esp32/overview>.
(Acessado em 9 de fevereiro de 2020)
- [16] **Arduino**, “Arduino UNO specs overview”, disponível em:
<https://www.farnell.com/datasheets/1682209.pdf> . (Acessado em 10 de fevereiro de 2020)
- [17] **IEEE Xplore**, disponível em:
<https://ieeexplore.ieee.org/Xplore/home.jsp>
(Acessado em 12 de fevereiro de 2020)

8. AUTO-AVALIAÇÃO DO ALUNO

Com o aumento da relevância de sistemas de automação de redes elétricas de potência, a demanda de dispositivos responsáveis pela supervisão inteligente da rede, também aumentou. É interessante participar de um projeto que estuda a aplicabilidade de micro controladores, para o monitoramento da rede. Esse projeto traz para o meu contexto de graduando, a usabilidade de ferramentas computacionais para solução de problemas de Engenharia Elétrica. Foi possível também entender melhor conceitos de medição, funcionamento de circuitos integrados, e a partir da etapa atual do projeto, o uso de modelos computacionais focados em classificação. Além disso, o projeto está sendo importante para mostrar o funcionamento de metodologias científicas e como deve ser feito o planejamento para a realização da pesquisa. Por fim, pude ter contato com alguns temas que são vistos muito rápido durante o curso de graduação em Engenharia Elétrica, e aplicar alguns dos conceitos que já foram aprendidos nas cadeiras cursadas.