

Capítulo III - Acionamento e Supervisão de Motores Elétricos Via WiFi, Abordando o Conceito de Internet das coisas (IoT)

Giancarlo Piazza Bellafronte¹²

Renato Kazuo Miyamoto¹³

Fábio R. Milanez¹⁴

Wesley Candido da Silva¹⁵

RESUMO

A Internet das Coisas (IoT) tem oferecido, uma variedade de recursos, com objetivo de facilitar comunicação e gerenciamento de dispositivos. Assim, este trabalho apresenta o desenvolvimento de um protótipo, com objetivo de efetuar o acionamento e o monitoramento de sinais de corrente elétrica e de potência em um motor de indução trifásico por meio de uma rede sem fio (WiFi). O método de partida utilizado foi do tipo estrela triângulo e o acionamento foi executado por uma página de servidor web. Para efetuar a coletas de dados da corrente trifásica consumida da rede elétrica, utilizou-se o sensor de corrente não invasivo SCT-013. Os dados coletados do monitoramento foram enviados e armazenados a uma nuvem da *ThingSpeak*, da empresa *Matlab*. Nesta nuvem, foram gerados gráficos dos valores da corrente e da potência elétrica, para possíveis análises de anormalias do funcionamento do motor.

Palavras-chave: comunicação rede *WiFi*, nuvem monitoramento, Internet das coisas

DEVELOPMENT OF A PROTOTYPE IN THE OPERATION AND MONITORING OF ELECTRIC MOTORS APPLIED IN THE CONCEPT OF THE INTERNET OF THINGS (IoT)

ABSTRACT

The Internet of Things (IoT) has offered a variety of features to facilitate communication and device management. That way, this article presents a prototype development, with the intention of facing a three-phase electric drive motors and the electric current data monitoring, consumed by wireless WiFi network. The used start system was the triangle star, which where the electric drive executed by a web server page, where contained the on and off commands. For collect data of these consumed currents of phases R, S, T were used a noninvasive current sensor SCT-013. The monitoring collected data were sent and stored inside a *Matlab's ThingSpeak Cloud*, in this cloud graphs were generated representing the current and power's values, for possible analysis of motor operating abnormalities.

Key-words: WiFi network communication, cloud monitoring, Internet of Things

¹² Pós-graduação em Engenharia de Automação Industrial; Faculdade da Indústria SENAI Londrina. E-mail: giancarlo.piazza10@hotmail.com

¹³ Prof. MS.c. Faculdades da Indústria Senai Londrina. E-mail: renato.miyamoto@sistemafiep.org.br

¹⁴ Prof. Faculdades da Indústria Senai Londrina. E-mail: fabio.milanez@sistemafiep.org.br

¹⁵ Prof. Faculdades da Indústria Senai Londrina. E-mail: wesley.candido@sistemafiep.org.br

1 INTRODUÇÃO

O desenvolvimento de novas tecnologias em sistemas eletrônicos embarcados e na área da tecnologia de informações, contribuiu para a impulsão da 4^o revolução industrial, mais conhecida como indústria 4.0. Um dos principais pilares da indústria 4.0 é a Internet das Coisas (IoT - *Internet of Things*) que permite que circuitos eletrônicos embarcados sejam responsáveis por efetuar comandos, coleta de dados de processos de fabricação e monitoramento remoto das máquinas (PEREIRA et al., 2016, p 20, 21).

O conceito de Internet das Coisas (IoT) integra o sistema físico de aquisição de dados com uma rede internet e realiza o envio dessas informações na nuvem onde são armazenadas. Outro benefício que o IoT propicia é a utilização do sistema M2M (Machine-to-Machine) que tem como objetivo tornar as fábricas inteligentes por meio de sensores e atuadores (NETO et al., 2018, p. 10, 11, 12).

Diante da evolução desta tecnologia e de sua implantação no chão de fábrica, houve o surgimento dos custos elevados, para efetuar a comunicação dos equipamentos, com uma central de controle e monitoramento, por meio de cabos ou fibra ópticas.

Deste modo, percebe-se a relevância de pesquisas que empregam a comunicação sem fio via WiFi, como uma alternativa de baixo custo na integração de IoT em meios físicos, e como consequência, há uma redução significativa da necessidade do uso de cabos ou fibra ópticas (PILON, 2009, p. 9).

Em Maestrelli, Napoleão (2018) foi proposto um projeto de automação residencial, cujo objetivo foi de efetuar comandos das cargas (lâmpadas) e a aquisição de dados do consumo de corrente e tensão elétrica por meio de sensores conectados com um módulo ESP 8266 da Espressif. Tal módulo é responsável por efetuar os comandos, conectar com a rede de internet e enviar os dados coletados para a nuvem.

De acordo com Aita (2017) foi realizado uma automatização de sistemas de irrigação na agricultura com comunicação sem fio. O projeto utilizou o módulo ESP 32 NodeMCU para efetuar a coleta de dados de umidade e PH do solo. A ação de controle consiste no acionamento de um sistema de irrigação de acordo com a leitura de dados de umidade. O projeto não utilizou sistema IoT e armazenamento de dados na nuvem.

A proposta deste trabalho consiste no desenvolvimento de um protótipo, para utilização em ambiente industrial, capaz de realizar acionamentos de motores elétricos por meio de servidor web e coleta de dados. Assim, utiliza-se dois módulos de comunicação Wifi do modelo ESP8266 NodemCU. Os dados coletados: corrente e potência, serão enviados na nuvem da ThingSpeak com uma interface gráfica contendo leitura instantânea e histórico.

O projeto traz uma solução de baixo custo para integração com a indústria 4.0 e manufatura enxuta com a redução da quantidade de cabos e/ou fibra ópticas na instalação. A aquisição dos dados contribui para o processo de manutenção preditiva e monitoramento de anomalias na máquina.

Assim para a realização deste projeto, foram utilizados conceitos de sistemas embarcados, microcontroladores e de sistemas de comunicação sem fio. Tais aspectos serão abordados no tópico 2.

2. FUNDAMENTAÇÃO TEÓRICA

Nos dias atuais podemos encontrar circuitos eletrônicos, sendo utilizados em veículos automotores, aviões, máquinas industriais entre outros equipamentos com a finalidade de efetuar funções específicas, pois possuem microcontroladores em sua construção podendo ser programados (OLIVEIRA, ANDRADE, 2012, p, 17).

2.1 SISTEMAS EMBARCADOS

De acordo com Oliveira; Andrade e Souza (2012, p. 17), os sistemas embarcados podem ser compostos por uma infinidade de unidades de processamento, como componentes programáveis, ou seja, componentes que necessitam que seu software (firmware) seja alocado interiormente.

Os microcontroladores são circuitos integrados programáveis, que podem ser conectados a sensores, atuadores em seus pinos de entrada e saída. Segundo Oliveira; Andrade e Souza (2012, p. 34), os microcontroladores possuem uma grande quantidade de periféricos podendo controlar atuadores, efetuar aquisição de sinais digitais e analógicos. Dessa forma, os microcontroladores têm um papel fundamental no sistema de automação.

Com o avanço da eletrônica embarcada, podemos encontrar no mercado uma ampla variedade circuitos eletrônicos, contendo em sua construção microcontroladores de última geração, com capacidade de efetuar comandos e se conectar a uma rede de WiFi. A Figura 01 ilustra o módulo de WiFi.

Figura 01: Módulo de WiFi ESP8266 NodemCU



Elaborado pelo autor (2019)

O ESP8266 NodemCU pode ser programado em linguagem Lua e pela IDE Arduino que se utiliza a Linguagem C. No quadro 01 ilustra suas especificações.

Quadro 01: Características do ESP8266 NodemCU

WiFi	802.11b/g/n - 2,4 Ghz
Processador Tensilica LX106	160Mhz
Memória RAM	96KBytes
Memória ROM boot	64KBytes
Memória flash	4MB
Conversor ADC	1 Canal

Elaborado pelo autor (2019)

A conexão de microcontroladores à sensores é usual em sistemas de controle. Os sensores escolhidos definem a variável de controle a ser analisada. Os sensores não invasivos (transformadores de corrente) são dispositivos, utilizados para medir a corrente que circula por um condutor sem a necessidade de seccionar o cabo de alimentação, possuem um núcleo de ferrite bipartidos ou inteiriços, contendo um enrolamento.

Segundo Granadeiro (2017, p. 30) afirma que: “[...] agem como indutores, uma vez respondem ao campo magnético em torno de um condutor com corrente. A figura 02 ilustra o sensor não invasivo SCT-013.

Figura 02: Sensor não invasivo SCT-013



Elaborado pelo autor (2019)

O sensor SCT-013 pode medir corrente alternadas de até 100 amperes. O quadro 02 ilustra suas características.

Quadro 02: Características do sensor não invasivo SCT-013

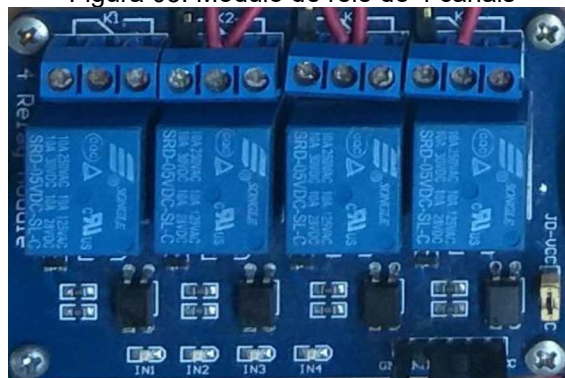
Corrente de entrada	0-100A
Corrente de saída	0-50mA

Elaborado por YHDC

O sensor SCT-013 pode ser conectado ao condutor sem a necessidade de efetuar o seccionamento, pois esse tipo de sensor possui núcleos de ferrite bipartido, proporcionando abertura do sensor para a introdução do condutor.

Por fim, para um sistema de controle em malha fechada, é necessário o emprego de atuadores. O módulo de relés de 4 canais é um dispositivo, utilizado para acionar cargas que possuem uma elevada potência por meio de microcontroladores. LUZ (2018, p. 14) esclarece que: “O relé é um interruptor eletromecânico aplicado em comandos elétricos, tendo a função de ligar ou desligar dispositivos. A Figura 03 ilustra o módulo de relés de 4 canais.

Figura 03: Módulo de relé de 4 canais



Elaborado pelo autor (2019)

O acionamento do relé é efetuado por um optoacoplador, assim isolando o circuito de comando dos relés. Os relés efetuam o fechamento dos contatos com sinal de nível baixo e a interrupção do circuito em nível alto.

2.2 SISTEMA DE COMUNICAÇÃO WiFi

Os avanços na tecnologia na área de sistemas de comunicação contribuíram para o desenvolvimento de vários sistemas de redes que não necessitam de fios ou cabos para efetuar comunicação como por exemplo a rede de WiFi (Wireless Fidelity).

A rede de WiFi possibilita o acesso de um smartphone a serviços da internet, efetuar a acionamento e monitoramento de sistemas de automação residenciais e nas indústrias. Pilon (2009, p. 09) afirma que “Os sistemas de comunicação sem fio possuem emprego crescente em sistemas de automação na indústria, no comércio e até mesmo em residências.

A grande falta de confiança de se utilizar WiFi nas indústrias está relacionada ao ambiente hostil com interferências eletromagnéticas, podendo vir a interferir na coleta de dados, para evitar tal problema é implantado alguns procedimentos para efetuar a correção.

Segundo Rocha (2005, apud PILON, 2009, p. 10) “[...] mas isto pode ser evitado através da reprogramação e pelo uso da técnica spread spectrum, um combate a faixas estreitas.

2.3 SISTEMA IoT

O sistema IoT (Internet of Things-Internet das Coisas) possibilitou a conexão das pessoas com equipamentos através da internet onde se pode efetuar o controle remotamente e monitoramento por meio de smartphones, notebook e computadores. Também é possível por meio de aplicativos efetuar acionamento de sistemas de irrigação em jardins de residências e bem como monitorar a umidade do solo.

O sistema IoT segundo Oliveira (2017, p.17) “Não é somente ligar as “coisas” pela internet, mas também torná-las inteligentes, capazes de coletar e processar informações do ambiente ou das redes às quais estão conectadas”.

A sua implantação no chão de fábrica, traz variedades de recursos para a melhoria da produção e na automação do processo.

Com os recursos da internet é possível criar páginas na web, onde o cliente pode requisitar um serviço ao servidor. Segundo Didone e Felipe-Chaulet (2016, p. 15) afirma que: “Os sistemas cliente/servidor têm como objetivo oferecer, através de um computador robusto uma série de serviços que ficam acessíveis aos clientes das mais diversas formas possíveis.

Dessa forma podemos criar páginas na Web com imagens de objetos como botões, para acionar algum dispositivo via Servidor Web.

2.3.1. Nuvem da internet

Com a evolução da tecnologia na área de automação industrial, acarretou uma grande quantidade de envio e recebimento de dados do processo produtivo a serem armazenados. Segundo Oliveira (2017, p. 75), salientou a importância de armazenar esses dados com um baixo custo, a alternativa foi a utilização de servidores externos mais conhecidos como Cloud (Nuvem).

Hoje podemos encontrar muitas empresas que oferecem, esse tipo de serviço como Dropbox, Google Drive, ThingSpeak da Matlab etc. O próximo Capítulo apresenta os métodos para o desenvolvimento do protótipo.

3 METODOLOGIA

A proposta deste trabalho emprega metodologia de estudo quantitativo, onde são coletados dados do consumo de corrente elétrica do motor e da potência, e são enviados a nuvem para geração de gráficos para monitoramento. Os testes foram realizados em uma empresa localizada na cidade de Londrina e as etapas para a implementação são descritas nas Seções 3.1 a 3.2.

3.1 Confeção do circuito impresso e Montagem do painel de comando

Para a elaboração do esquema elétrico e a placa PCB, foi utilizado um software dedicado a projetos de circuitos eletrônicos Proteus. Foram elaborados dois circuitos

elétricos de acionamento e monitoramento onde foram fixados os módulos de WiFi ESP8266 NodemCU e sensores SCT-013. (Conforme Apêndice A).

No software ARES foram elaboradas as placas PCB de monitoramento e acionamento (Conforme Apêndice B). O Primeiro passo foi elaboração do esquema elétrico do sistema de partida estrela triângulo (Conforme Apêndice C).

O módulo de Wifi ESP8266 NodemCU programado para efetuar a partida estrela triângulo, foi conectado ao módulo de relés de 4 canais e suas saídas ligadas as respectivas entradas das bobinas eletromagnéticas das chaves contadoras responsáveis pelo acionamento do motor. Os sensores SCT-013 foram conectados individualmente a cada módulo de comunicação, programados a enviar os dados a nuvem, e cada sensor foi inserido a rede de alimentação do motor, para monitorar as fases R, S, T.

Após a montagem do painel foi realizada a conexão do motor, alimentação do painel e conexão dos sensores SCT-013 cada um em sua fase definida (Conforme Apêndice D).

3.2 Programação do servidor web

O módulo de WiFi ESP8266 NodemCU do acionamento do motor foi programado, pela IDE Arduino em linguagem C. Foi utilizado um exemplo de Web Server, contido no arquivo do software de programação da IDE Arduino, responsável por criar uma página na Web, em que foi implementado o código fonte da partida estrela triângulo (Conforme Apêndice F). O exemplo de código fonte utilizado apresenta na tela os comandos liga e desliga requisitados pelo cliente, e efetua comandos de conexão com a rede WiFi pela senha e nome da rede a ser utilizada (Conforme Anexo A).

3.3.1 Programação de monitoramento

Para execução do monitoramento pelos módulos WiFi ESP8266 utilizando, sensores SCT-013 e o envio de dados a nuvem, foi utilizado o exemplo de código fonte de rotina para efetuar a conexão com a cloud da ThingSpeak do livro Internet das Coisas (Conforme Anexo B).

No código fonte de comunicação com ThingSpeak, com aplicação Datalog na nuvem, foi implementado partes de um exemplo de programação, responsável por efetuar a aquisição dos valores da corrente consumida e a potência do motor. Disponível em:< <https://portal.vidadesilicio.com.br/sct-013-sensor-de-corrente-alternada/>> (Conforme Anexo C). Assim, foi necessário a implementação de comando para possibilitar o envio de dados a nuvem e gerar gráficos (Conforme Apêndice F).

3.3.2 Nuvem ThingSpeak

A nuvem a ser usada para gerar gráficos e armazenar os dados é da empresa Matlab ThingSpeak, onde foi necessário criar uma conta para ter acesso. Após a confirmação da conta, foram criados 3 canais e 6 campos para a geração dos gráficos de corrente e potência (Conforme Apêndice G).

Cada canal é identificado por uma numeração e chaves escritas, esses dados são inseridos no código fonte para efetuar a comunicação com a nuvem. Para armazenar os dados coletados à nuvem, são exportados manualmente em formato de arquivo JSON, XML, orCSVdata para uma região de armazenamento. Os resultados do teste do protótipo do comando do servidor web e aquisição dos dados da corrente e potência serão abordados no tópico 4.

4 APRESENTAÇÃO E DISCUSSÃO DOS RESULTADOS

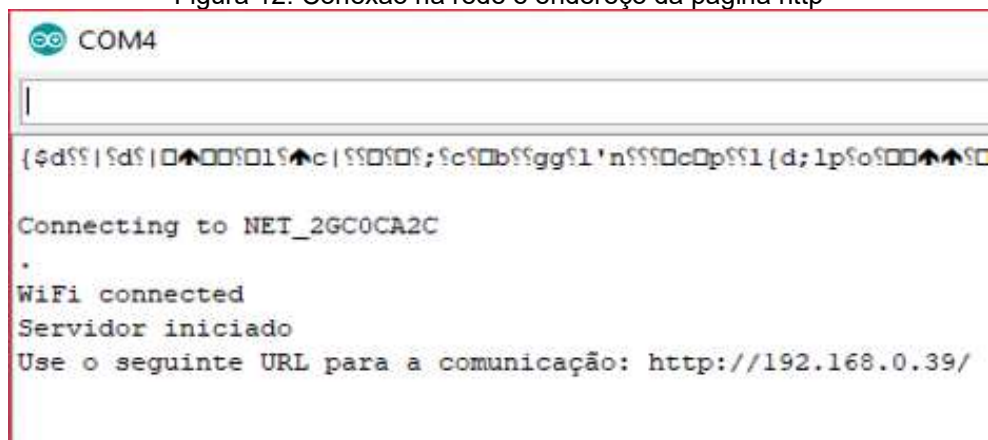
Neste tópico são apresentados, os resultados de acionamento e o envio de dados monitorados a nuvem, obtidos com os testes aplicados no protótipo apresentado neste trabalho.

4.1 Acionamento do motor

A comunicação do módulo ESP8266 NodemCU, com a rede de WiFi ocorreu sem nenhum tipo de interferência, mesmo implementado em ambiente fabril, onde pode ocorrer efeitos de indução eletromagnética. O monitor serial da IDE Arduino,

gerou o endereço da página da Web Server local. A Figura 12 ilustra a página de conexão na rede e endereço da Web Server local.

Figura 12: Conexão na rede e endereço da página http



Elaborado pelo autor (2019)

O serial monitor informa o nome da rede conectada, o servidor iniciado e o endereço de acesso ao servidor local. Executando o acesso a página do servidor é gerado a tela de comando do acionamento do motor de acordo com a Figura 13.

Figura 13: Servidor



Elaborado pelo autor (2019)

Após o acesso a página foi efetuado o acionamento do motor, onde a conexão da rede não houve queda de sinal e se manteve estabilizada.

4.2 Monitoramento e aquisição de dados

Os dados coletados pelo circuito de monitoramento foram enviados a nuvem e gerados os gráficos de corrente consumida e potência das fases R, S, T em tempo real, onde também foi utilizado um alicate amperímetro para comparação dos dados coletados. A Figura 14 ilustra o gráfico da corrente da fase R.

Figura 14: Gráfico da corrente na fase R



Elaborado pelo autor (2019)

A partir da análise da Figura 14, observa-se que para os dados coletados de corrente elétrica é informado, o horário do acionamento e valor eficaz. O valor da corrente é de 3.5 Amperes. Buscando validação nos sinais coletados, foi utilizado um alicate amperímetro com o propósito de efetuar, comparações com o valor registrado no gráfico. A Figura 15 ilustra o valor registrado da corrente no alicate amperímetro.

Figura 15: Valor da corrente fase R

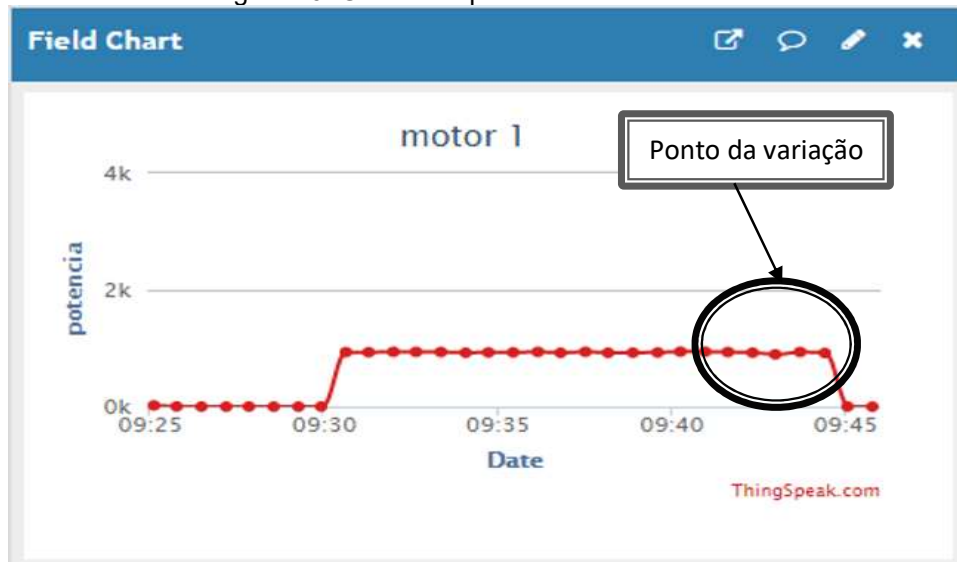


Elaborado pelo autor (2019)

O valor gerado no gráfico de potência apresentou-se inferior a 1000Watts, devido ao fato do motor operar à vazio. Foi registrado uma pequena variação na

potência após as 9:40 am. A Figura 16 ilustra o gráfico da potência do motor na fase R.

Figura 16: Gráfico da potência do motor fase R



Elaborado pelo autor (2019)

O surgimento da variação da potência da fase R, pode ser justificado por uma possível variação na tensão da rede. Pois outras máquinas estavam em funcionamento assim podendo gerar oscilações na rede.

No canal 2, referente a fase S, houve um aumento na corrente de 0,7 amperes, mas no gráfico não sofreu alterações que podem ser notadas. A Figura 17 ilustra o valor da corrente no alicate amperímetro na fase S.

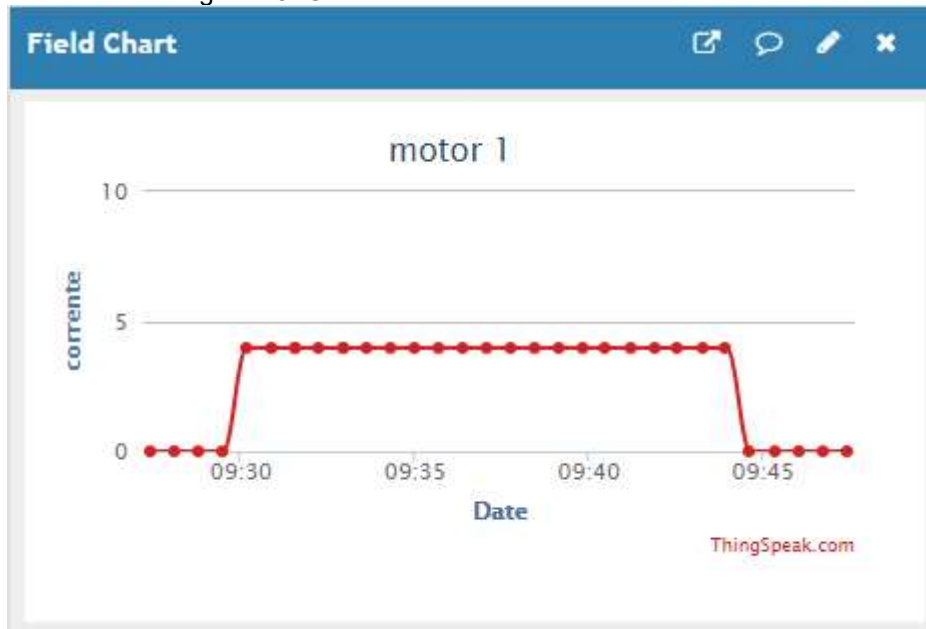
Figura 17: Valor da corrente na fase S



Elaborado pelo autor (2019)

Efetuada a medição da corrente na fase S, foi observado o aumento em relação do valor da fase R. O valor mensurado da fase S apresentou 4.2 amperes. A figura 18 ilustra o gráfico da corrente na fase S.

Figura 18: Gráfico da corrente do motor na fase S



Elaborado pelo autor (2019)

O gráfico da potência da fase S também apresentou pontos de variações. A Figura 19 ilustra o gráfico da potência em relação a fase S.

Figura 19: Gráfico da potência da fase S



Elaborado pelo autor (2019)

Na fase S foram registrados, dois pontos de variações em que as 9:35 am houve um aumento da potência, ao contrário de minutos antes do desligamento do motor onde foi menor. O gráfico da fase T do canal 3, não registrou variações na corrente, se manteve estabilizado. A Figura 20 ilustra o gráfico da corrente da fase T.

Figura 20: Gráfico da corrente na fase T



Elaborado pelo autor (2019)

O valor da corrente na fase T apresentou um aumento de 0,4 amperes em relação a fase S. A Figura 21 ilustra o valor da corrente.

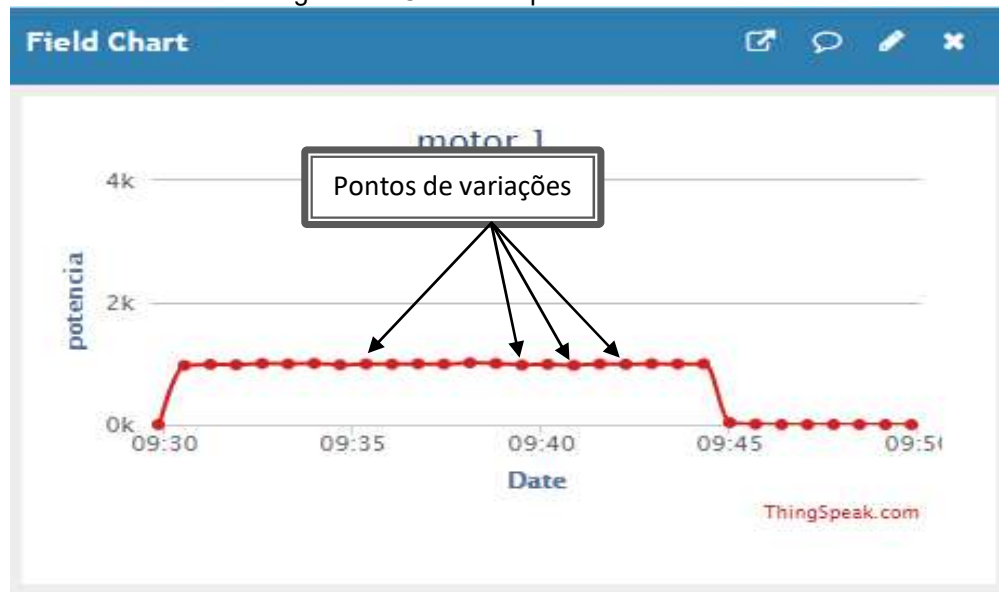
Figura 21: Valor da corrente na fase T



Elaborado pelo autor (2019)

Analisando o gráfico, foi verificado vários pontos de variação no valor da potência na fase T. A Figura 22 ilustra os pontos de variações no gráfico. A causa dessas variações na fase T pode ser justificada pelo funcionamento de outro equipamento que estava conectado à mesma fase.

Figura 22: Gráfico da potência na fase T



Elaborado pelo autor (2019)

Assim após efetuar o teste do protótipo e verificar seu funcionamento foi possível validar este projeto, de modo a comprovar a eficácia da implementação do sistema de automação com a utilização de IoT em ambiente fabril. O Capítulo 5 aborda as considerações finais deste trabalho.

5 CONSIDERAÇÕES FINAIS

A construção protótipo de comunicação WiFi apresentou resultados desejados, onde a conexão se manteve estável e sem interrupção. O acionamento do motor ocorreu corretamente. Os valores obtidos no gráfico de corrente tiveram um desvio padrão de 94mA em relação as medições efetuadas pelo alicate amperímetro.

O envio de comando de acionamento e de dados da corrente consumida pelo motor e potência a nuvem (Cloud) apresentaram resultados satisfatórios. Deixando como sugestão de pesquisas futuras a utilização do módulo WiFi ESP32 por possuir entradas analógicas de 12 Bits de resolução, onde que nesta pesquisa não pode ser

utilizado, por incompatibilidade do programa do sensor SCT-013, em que não foram obtidos resultados satisfatórios.

REFERÊNCIAS

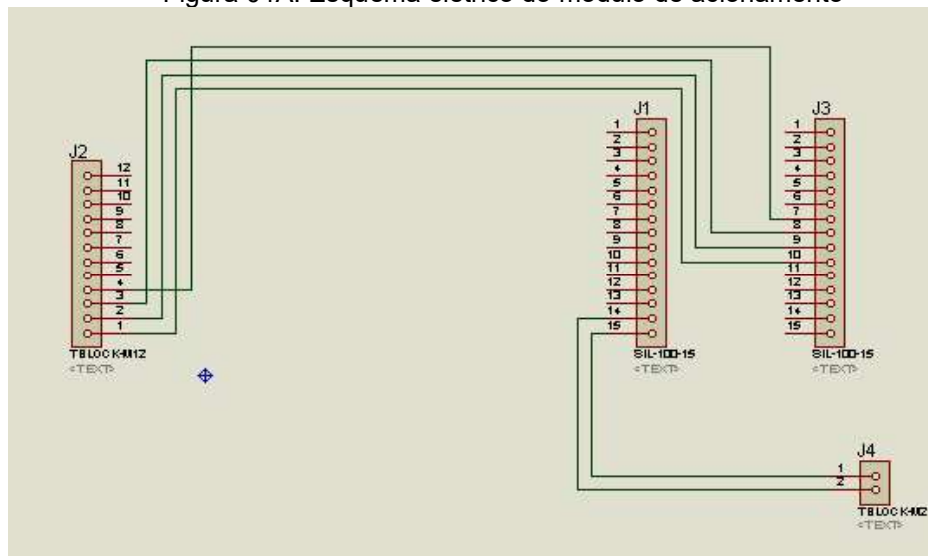
- AITA, R.; **Sistema de irrigação localizada e automatizada**, 2017 [online]. Disponível em: < <http://www.politecnica.pucrs.br/conclusao/files/20172-ricardo-hahn-aita-VOLUME-2644.pdf> > Acesso em 01/12/2018.
- DIDONÉ, D.; CHAULET, F. **Implantação e Administração de Serviços web**, 2016 [online]. Disponível em: < https://www.ufsm.br/unidades-universitarias/ctism/cte/wp-content/uploads/sites/413/2018/12/arto_implantacao_administracao_servicos_web.pdf > Acesso em 10/01/2019.
- EZEQUIEL D. **SCT-013-Sensor de Corrente Alternada com Arduino**, 2017 [online]. Disponível em: < <https://portal.vidadesilicio.com.br/sct-013-sensor-de-corrente-alternada/> >. Acesso em 24/01/2019 às 19 h 45 min.
- GRANADEIRO, M., Sistema de monitorização energética e comportamental **SMEC, Mestrado – Universidade de Lisboa**, 2017.
- LUZ, H. SISTEMA DE AUTOMAÇÃO RESIDENCIAL/PREDIAL UTILIZANDO ARDUINO E SISTEMA OPERACIONAL ANDROID. **Monografia – Universidade Federal de Ouro Preto**, 2018.
- MAESTELLI, G.; NAPOLEÃO, G. SISTEMA SUPERVISÓRIO PARA MONITORAMENTO DE ENERGIA ELÉTRICA RESEDENCIAL. **Monografia – Universidade Tecnológica Federal do Paraná**, 2018.
- NETO, E. et al., **A Internet das Coisas na Indústria**, 2018 [online]. Disponível em: < <http://rafaelkrolow.info/wp-content/uploads/2018/07/WP-IoT-01.pdf> > Acesso em 15/11/2018.
- OLIVEIRA, A.; ANDRADE, F. **Sistemas Embarcados hardware e Firmware na Prática**, São Paulo: Editora Ética, 2012.
- OLIVEIRA, S. **INTERNET DAS COISAS com ESP8266, ARDUINO E RASPBERRY PI**, São Paulo: Editora Novatec, 2017.
- PEREIRA, A. et al., **INTEGRAÇÃO DE INFORMAÇÃO NO PROCESSO DE DESENVOLVIMENTO DE PRODUTO SOB A ÓTICA INDÚSTRIA 4.0**, 2016. Disponível em: < <http://lyceumonline.usf.edu.br/salavirtual/documentos/2752.pdf> > Acesso em 08/12/2018.

PILON, V. ESTUDO PARA APLICAÇÃO DE REDES SEM FIO NO AMBIENTE INDUSTRIAL. **Monografia – Universidade Tecnológica Federal do Paraná, 2009.**

APÊNDICE A

A figura 04A apresenta o esquema elétrico do módulo de acionamento.

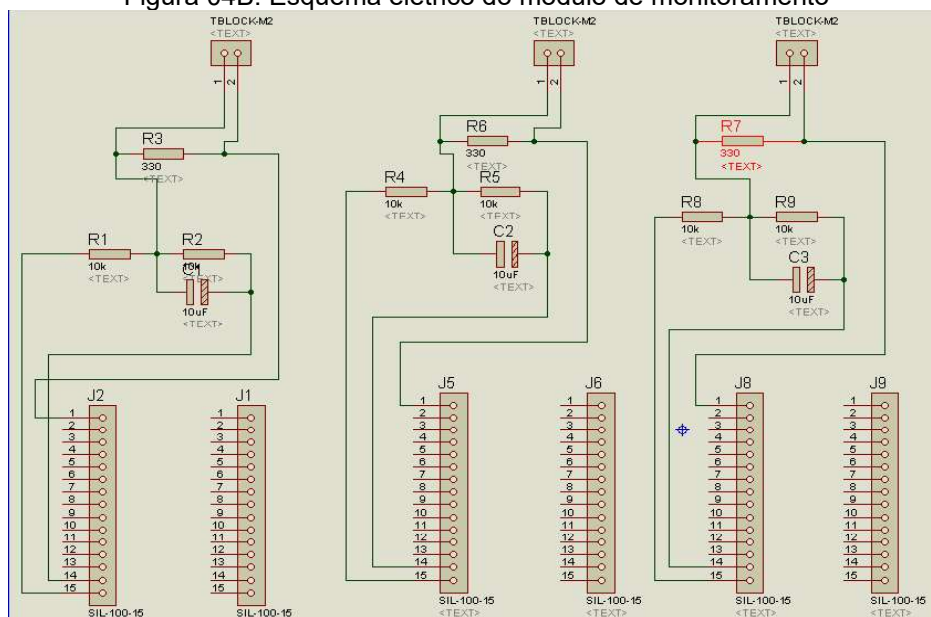
Figura 04A: Esquema elétrico do módulo de acionamento



Elaborado pelo autor

O esquema elétrico do circuito do módulo de monitoramento é apresentado na figura 04B.

Figura 04B: Esquema elétrico do módulo de monitoramento

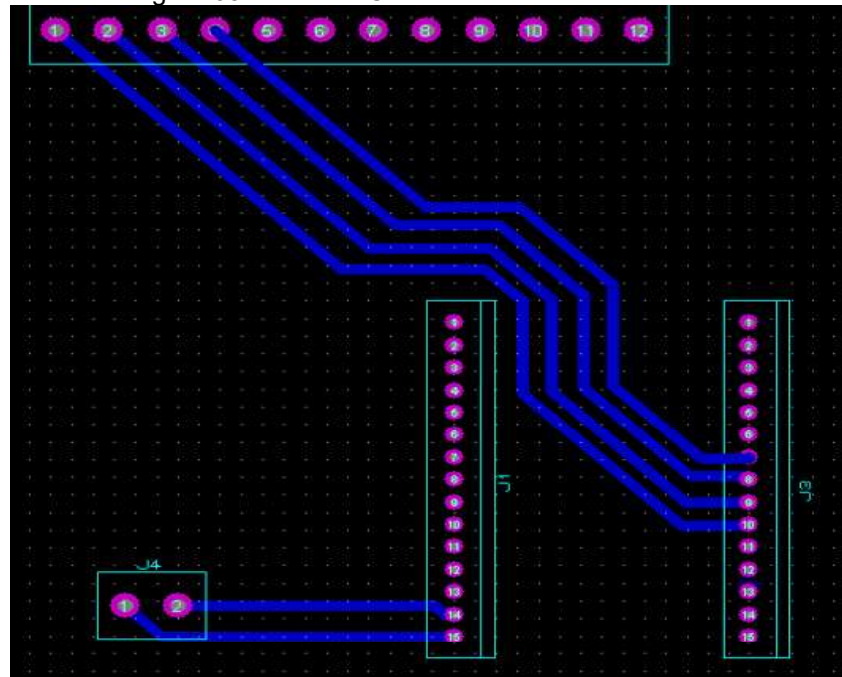


Elaborado pelo autor

APÊNDICE B

O circuito PCB do módulo de acionamento é apresentado na figura 05A.

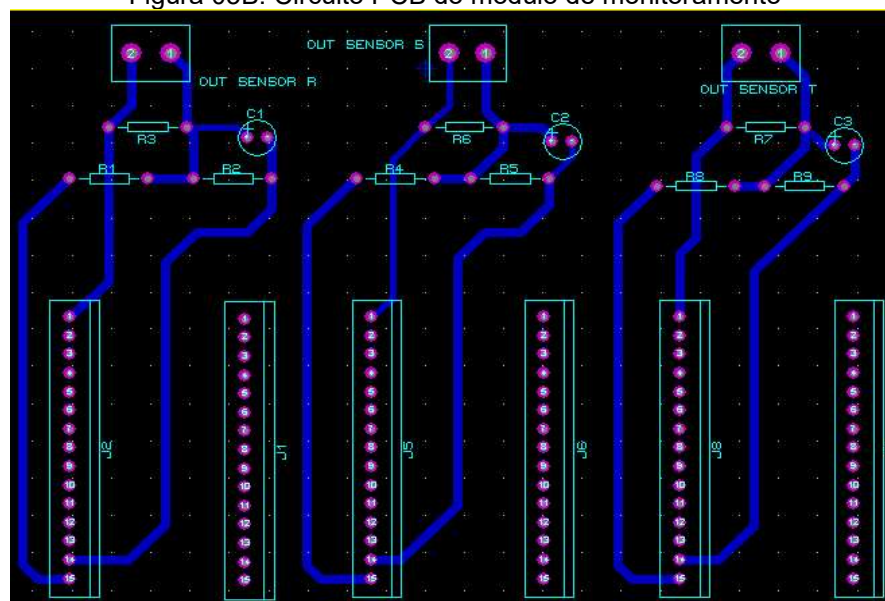
Figura 05A: Placa PCB do Módulo de acionamento



Elaborado pelo autor

A figura 05B mostra a placa PCB do módulo de monitoramento.

Figura 05B: Circuito PCB do módulo de monitoramento

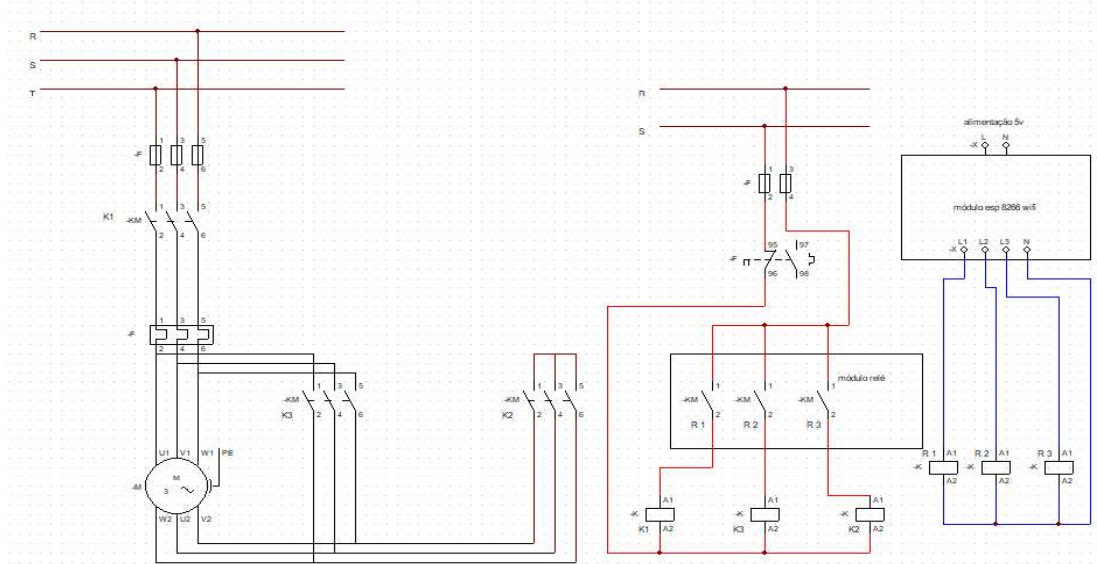


Elaborado pelo autor

APENDICE C

A figura 06 ilustra o diagrama elétrico do painel.

Figura 06: Esquema elétrico do painel de comando

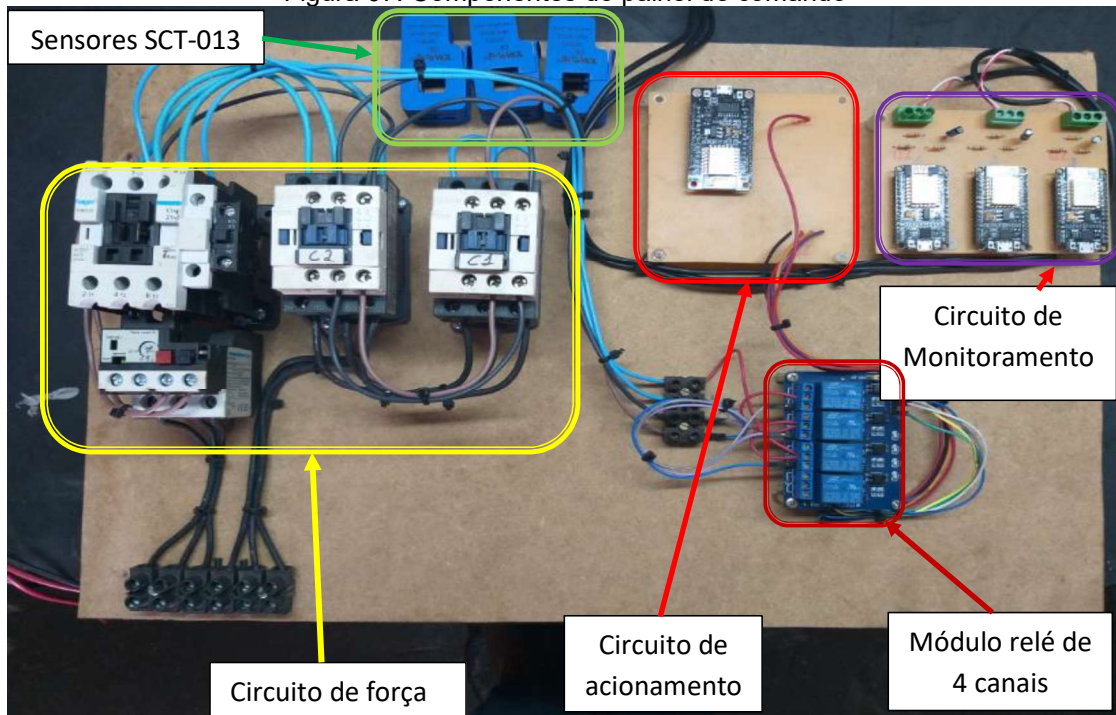


Elaborado pelo autor

APÊNDICE D

A figura 07 ilustra o painel de comando e seus componentes.

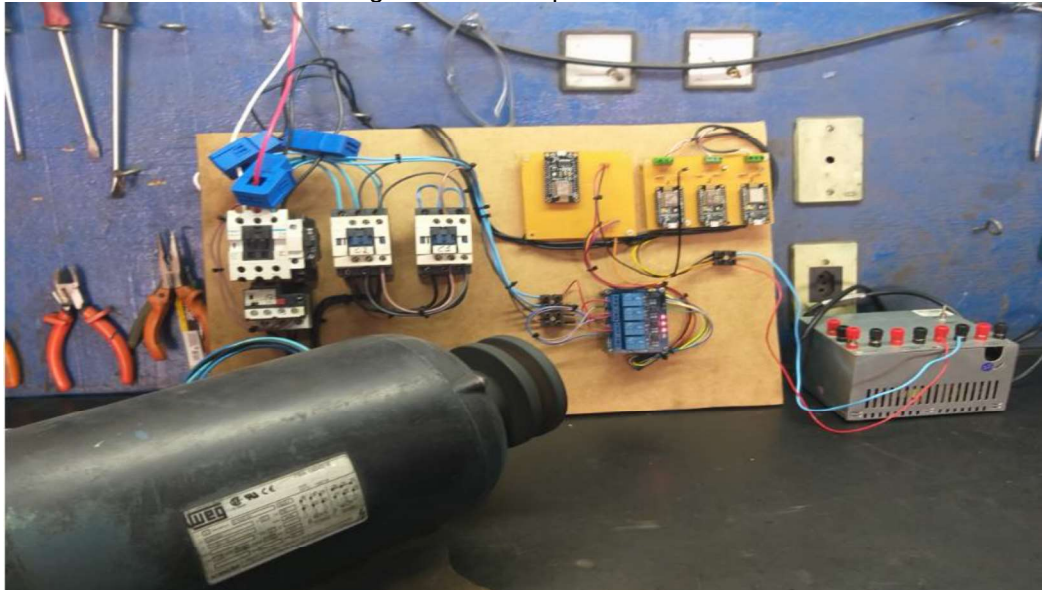
Figura 07: Componentes do painel de comando



Elaborado pelo autor

A figura 08 apresenta o painel de comando finalizado e pronto para efetuar os testes.

Figura 08: Protótipo finalizado



Elaborado pelo autor

APÊNDICE E

A figura 09 apresenta bloco do código fonte, responsável por efetuar a partida estrela triângulo.

Figura 09: Bloco do código fonte comando da partida

```

153
154 digitalWrite(relePin1, HIGH);
155 digitalWrite(relePin2, LOW);
156 digitalWrite(relePin3, HIGH);
157 delay(2000);
158 digitalWrite(relePin1, HIGH);
159 digitalWrite(relePin2, LOW);
160 digitalWrite(relePin3, LOW);
161 delay(3000);
162 digitalWrite(relePin1, HIGH);
163 digitalWrite(relePin2, HIGH);
164 digitalWrite(relePin3, LOW);
165
166 value = HIGH;
167 }
168 if (request.indexOf("/RELE=OFF") != -1) {
169
170 digitalWrite(relePin1, LOW); [
171 digitalWrite(relePin2, LOW);
172 digitalWrite(relePin3, LOW);
173 value = LOW;

```

Elaborado pelo autor

APÊNDICE F

A figura 10 apresenta a implantação de variáveis da linha de programação 09 e 38 para executar o envio dos dados a nuvem.

Figura 10: Implantação das variáveis no código fonte

```
9 int Irmsl;  
38 Irmsl = potencia/tensao;
```

Elaborado pelo autor

APÊNDICE G

A figura 11 apresenta os canais com suas datas de criação.

Figura 11: Informações dos canais

Nome	Criada em	Updated
motor 1 r Private Públicos Settings Sharing API Keys Data Import / Export	2019-02-15	2019-03-06 00:56
motor 1 s Private Públicos Settings Sharing API Keys Data Import / Export	2019-02-22	2019-03-06 03:45
motor 1 T Private Públicos Settings Sharing API Keys Data Import / Export	2019-02-22	2019-03-14 01:35

Elaborado pelo autor

ANEXO A

O exemplo de código fonte de criação de um servidor web utilizado da IDE Arduino.

/*

This sketch demonstrates how to set up a simple HTTP-like server.

The server will set a GPIO pin depending on the request
 http://server_ip/gpio/0 will set the GPIO2 low,
 http://server_ip/gpio/1 will set the GPIO2 high
 server_ip is the IP address of the ESP8266 module, will be
 printed to Serial when the module is connected.

```

*/
#include <ESP8266WiFi.h>
#include <ESP8266WebServer.h>
const char* ssid = "**** _*****";
const char* password = "*****";

int relePin1 = 12;
int relePin2 = 13;
int relePin3 = 14;
WiFiServer server(80);
void setup() {
  Serial.begin(115200);
  delay(10);
  pinMode(relePin1, OUTPUT);
  digitalWrite(relePin1, LOW);
  pinMode(relePin2, OUTPUT);
  digitalWrite(relePin2, LOW);
  pinMode(relePin3, OUTPUT);
  digitalWrite(relePin3, LOW);
  Serial.println();
  Serial.println();
  Serial.print("Connecting to ");
  Serial.println(ssid);
  WiFi.begin(ssid, password);
  while (WiFi.status() != WL_CONNECTED) {
    delay(500);
    Serial.print(".");
  }
}

```

```

Serial.println("");

Serial.println("WiFi connected");
server.begin();
Serial.println("Servidor iniciado");
Serial.print("Use o seguinte URL para a comunicação: ");
Serial.print("http://");
Serial.print(WiFi.localIP());
Serial.println("/");
}
void loop() {
WiFiClient client = server.available();
if (!client) {
return;
}
Serial.println("novo cliente");
while(!client.available()){
delay(1);
}

String request = client.readStringUntil('\r');
Serial.println(request);
client.flush();
int value = LOW;
if (request.indexOf("/RELE=ON") != -1) {
digitalWrite(relePin1, HIGH);
digitalWrite(relePin2, LOW);
digitalWrite(relePin3, HIGH);
delay(2000);
digitalWrite(relePin1, HIGH);
digitalWrite(relePin2, LOW);
digitalWrite(relePin3, LOW);
delay(3000);
}
}

```

```

digitalWrite(relePin1, HIGH);
digitalWrite(relePin2, HIGH);
digitalWrite(relePin3, LOW);
value = HIGH;
}
if (request.indexOf("/RELE=OFF") != -1) {
digitalWrite(relePin1, LOW);
digitalWrite(relePin2, LOW);
digitalWrite(relePin3, LOW);
value = LOW;
}
client.println("HTTP/1.1 200 OK");

client.println("Content-Type: text/html");
client.println(""); // do not forget this one
client.println("<!DOCTYPE HTML>")
client.println("<html>");
client.print("Estado do RELE: ");

if(value == HIGH) { // Se está ligado apresenta "on"
client.print("On");
} else {
client.print("Off");// Se está desligado apresenta "Off"
}
client.println("<br><br>");
client.println("<a href=\"/RELE=ON\"><button>Turn On </button></a>");
client.println("<a href=\"/RELE=OFF\"><button>Turn Off </button></a><br
/>");
//client.println("<a href=\"/LED1=ON\"><button>Turn On </button></a>");
//client.println("<a href=\"/LED1=OFF\"><button>Turn Off </button></a><br
/>");
client.println("</html>");
delay(1)

```

```
Serial.println("Cliente desconectado");
Serial.println("");
}
```

ANEXO B

Neste anexo apresenta o código fonte para executar a comunicação e monitoramento e envio dos dados da corrente consumida do motor, com ThingSpeak.

```
#include <ESP8266WiFi.h>
#include <ThingSpeak.h>
#include <EmonLib.h>
EnergyMonitor SCT013;
int pinSCT = A0; //Pino analógico conectado ao SCT-013
int tensao = 220;
int potencia ;
int Irms1;
WiFiClient cliente;
const unsigned long NumeroCanal = 710347;
const char * ChaveEscrita = "RV7I1KNXXDNS285Q";
void setup() {
  SCT013.current(pinSCT, 6.0606);
  Serial.begin(115200);
  delay(250);
  Serial.println();
  Serial.println("Iniciando.....");
  WiFi.begin("*** _ ****", "*****");
  int tentativas = 0;
  while ((WiFi.status())!= WL_CONNECTED)&& tentativas++ <20){
    delay(500);
    Serial.print(".");
  }
  ThingSpeak.begin(cliente);
```

```

}
void loop() {
int pinSCT = (A0);
double Irms = SCT013.calcIrms(1480);
potencia = Irms* tensao;
Irms1 = potencia/tensao;
ThingSpeak.writeField(NumeroCanal,1,Irms1,ChaveEscrita);
delay(15000);
ThingSpeak.writeField(NumeroCanal,2,potencia,ChaveEscrita);
delay(15000);
}

```

ANEXO C

Partes do código fonte da aquisição da corrente elétrica utilizadas para implementar na programação da nuvem da ThingSpeak.

```

EnergyMonitor SCT013;
int pinSCT = A0; //Pino analógico conectado ao SCT-013
int tensao = 127;
int potencia;
void setup()
{
SCT013.current(pinSCT, 6.0606)
Serial.begin(9600);
}
void loop()
{
double Irms = SCT013.calcIrms(1480); // Calcula o valor da Corrente
potencia = Irms * tensao; // Calcula o valor da Potência Instantânea
}

```