



COMPARATIVO ENTRE PLATAFORMAS DE DESENVOLVIMENTO DE SISTEMAS MICROCONTROLADOS PARA APLICAÇÕES EM INTERNET DAS COISAS

Artur de Almeida Rios*¹, Darizon Alves de Andrade¹, Luciano Coutinho Gomes¹, Henrique José Avelar² e Frederico Duarte Fagundes²

¹FEELT – Universidade Federal de Uberlândia

²CEFET-MG – Centro Federal de Educação Tecnológica de Minas Gerais

Resumo - Seguindo as tendências da indústria 4.0, que está inserindo a Internet das Coisas (*IoT - Internet of Things*) em ambientes industriais, este trabalho se concentra em analisar três plataformas de desenvolvimento de sistemas microcontrolados para aplicações em *IoT*, são elas: ESP32, ESP8266 e MKR1000. Essas plataformas foram escolhidas por serem plataformas comerciais e possuírem um módulo *WiFi* integrado, pois um dos pontos do trabalho é salvar os dados coletados em nuvem. Foram realizadas comparações teóricas, entre as principais especificações das plataformas, e práticas como a velocidade de processamento, por meio de uma rotina de monitoramento e transmissão de um sinal analógico de temperatura, intensidade do sinal, consumo de energia e a verificação do uso de sensor hall interno do ESP32. As tabelas e figuras ao longo desse trabalho apresentam resultados que auxiliam na escolha da plataforma mais indicada para a aplicação em questão.

Palavras-Chave - Internet das Coisas, Microcontroladores, Sistemas Embarcados.

COMPARATIVE BETWEEN DEVELOPMENT PLATFORMS OF MICROCONTROLLED SYSTEMS FOR APPLICATIONS ON THE INTERNET OF THINGS

Abstract – Following the trends of industry 4.0, which is inserting the Internet of Things (IOT) into industrial environments, this paper focuses on analyzing three microcontroller systems development platforms for IoT applications: ESP32, ESP8266 and MKR1000. These platforms were chosen because they are commercial platforms and have an integrated WiFi module, since one of the points of the work is to save the data in the cloud. Theoretical comparisons were carried out between the main specifications of the platforms and practices such as processing speed, through a routine of monitoring and transmission of an analogue signal of temperature, signal

intensity, energy consumption and verification of use of the ESP32 internal sensor. The tables and figures throughout this work present results that help in choosing the most suitable platform for the application in question.

Keywords - Internet of Things, Microcontrollers, Embedded Systems.

I. INTRODUÇÃO

Sistemas de aquisição de baixo custo baseados em sistemas embarcados vem sendo pesquisados nos últimos anos [1]. Trabalhos anteriores, já demonstravam a importância de métodos de monitoramento em tempo real [2].

O uso de plataformas de desenvolvimento de sistemas microcontrolados é uma solução para adquirir dados dos sensores, comunicar com outros dispositivos, armazenar informações localmente ou na nuvem e alertar o usuário quando falhas forem detectadas, como demonstrado em trabalhos anteriores, voltados para a indústria, com o monitoramento em tempo real das condições de motores de indução trifásicos, como vibração e temperatura, utilizando-se de sistemas microcontrolados e *IoT* [3].

Dentre as várias plataformas comercialmente encontradas destacam-se três, por possuírem *WiFi* integrado, as quais são: ESP8266, ESP32 e MK1000.

ESP8266 é um microcontrolador projetado pela *Espressif Systems*, empresa chinesa sediada em Xangai. Ele foi lançado como uma solução para realizar uma ponte entre os microcontroladores existentes e o *WiFi*. O ESP32 é sua versão melhorada, pois dentre outros quesitos, além de comunicação *WiFi* também possui comunicação *Bluetooth* [4].

MKR1000 é uma plataforma projetada pelo grupo *Arduino*, que iniciou seus projetos na cidade de Ivrea, Itália, em 2005. Essa plataforma de prototipagem eletrônica *open-source* (Código Aberto) combina as funcionalidades de modelos anteriores, além de dispor de *WiFi* integrado. É baseada em *hardware* e *software* flexíveis e fáceis de usar [5].

A Internet das Coisas (*IoT - Internet of Things*) pode ser definida como a conexão entre o mundo físico e o digital. Tal tecnologia permite que objetos cotidianos, pessoais ou industriais, sejam acionados pela Internet, desde que sejam

dotados de computação embarcada e unicamente identificados, trocando dados entre si e também com pessoas por meio da rede, permitindo a criação de serviços e experiências de uso adaptadas ao contexto [6].

Com o avanço das tecnologias envolvendo *IoT*, a indústria tem demonstrado interesse em se adaptar a esse cenário. A indústria 4.0, considerada a nova revolução industrial, é um bom exemplo da inserção da *IoT* em ambientes industriais [7]. Um protocolo largamente utilizado em aplicações de *IoT* é o *MQTT* (*Message Queue Telemetry Transport*), protocolo de comunicação leve e rápido utilizado principalmente na comunicação *M2M* (*Machine-to-machine*) [8].

Microcontrolador é um CI (Circuito Integrado) capaz de efetuar processos lógicos com rapidez, por englobar os circuitos do microprocessador, memória, conversão analógico-digital (AD) e portas digitais de entrada e saída, como também circuitos PWM e de comunicação dedicados. A facilidade de programação aliada à quantidade de circuitos internos dedicados, permite seu uso em uma vasta gama de tarefas.

O conhecimento e familiarização com essas tecnologias se tornam de suma importância, assim como o estudo e comparação entre novas plataformas microcontroladas que surgem a cada dia, visando acompanhar as tendências da indústria 4.0 [9].

Sistema embarcado pode ser definido como um circuito integrado, equipamento ou sistema dotado de capacidade computacional especializado que faz parte de uma máquina ou sistema maior [10].

O NodeMCU é um *firmware* de código aberto, consistindo de um kit de desenvolvimento para auxiliar na criação de protótipos de produtos voltados para *IoT*, de forma semelhante aos kits de desenvolvimento com códigos abertos – DevKits, que geralmente possuem o chip ESP8266 ou ESP32 [11].

Para a programação das plataformas MKR100, ESP32 e ESP8266, foi utilizado o Ambiente de Desenvolvimento Integrado (*IDE - Integrated Development Environment*), da Arduino, que é de código aberto, desenvolvido em Java, e projetado para facilitar a escrita de programas e o *upload* para as plataformas de desenvolvimento [12].

II. PROPÓSITO

O objetivo deste trabalho é definir, através de testes comparativos, a plataforma de desenvolvimento mais adequada, dentre as três selecionadas, para se trabalhar com aplicações envolvendo *IoT*.

A partir da revisão da literatura, sabe-se que a velocidade de processamento do microcontrolador, a intensidade do seu sinal sem fio e o consumo são de suma importância para trabalhar com *IoT*, sendo estes os parâmetros utilizados nas comparações entre as plataformas neste trabalho.

III. METODOLOGIA

Foi realizada a comparação entre três plataformas de sistemas microcontrolados, sendo uma plataforma Arduino modelo MKR1000, cujo *chip* é o ATMEL SAMW25, e duas plataformas da fabricante Espressif Systems, onde a primeira é denominada NodeMCU-32S, a qual utiliza o *chip* ESP-WROOM-32, ou apenas ESP32, e a segunda é o LoLin

NodeMCU V3, a qual utiliza o *chip* ESP8266MOD, ou ESP8662, conforme mostra a Figura 1.

Figura 1: Plataformas de desenvolvimento de sistemas microcontrolados: ESP32, MKR1000 e ESP8266, da esquerda para direita.



Foram confrontadas as principais especificações de cada microcontrolador e de cada plataforma, além de realizar testes práticos de velocidade de processamento e de intensidade do sinal para fins de comparação entre as mesmas.

Para comparar a velocidade de processamento foi desenvolvido um código no IDE da Arduino e carregado em cada microcontrolador, que será explicado no item B da seção IV.

Quanto a comparação da intensidade do sinal, as três plataformas, que possuem *WiFi* integrado, foram programadas como *Access Point* (Ponto de Acesso), e realizadas as medições, este teste será explicado no item C da seção IV.

No item D da seção IV, foram realizados os testes de consumo. Para tal foi utilizado um adaptador *USB* (*USB Tester*), que realiza a leitura do consumo em amperes da porta *USB* à qual está conectada a plataforma sob teste.

Além dos testes citados anteriormente, também realizamos no ESP32, testes com o sensor *hall*, a fim de verificar a possibilidade de utilização desse sensor para adquirir variáveis de campo magnético em motores sem utilização de sensores adicionais, os resultados estão no item E da seção IV.

IV. RESULTADOS

Foram realizados as comparações teóricas e os testes práticos de velocidade de processamento, intensidade do sinal *WiFi* e teste de consumo de energia entre as três plataformas. E também foi realizado o teste no sensor *hall* do ESP32.

A. Especificações das Plataformas

Foram feitas comparações entre as principais especificações de cada plataforma, baseadas nos *datasheets* de seus respectivos fabricantes conforme apresentadas na Tabela I [13] [14] [15].

Já as principais especificações dos microcontroladores utilizados em cada plataforma são apresentadas na Tabela II, também baseadas nos *datasheets* de seus fabricantes [16] [17] [18].

Tabela 1: Comparativo entre plataformas.

Plataformas			
Nome Comercial	ESP32	ESP8266	MKR1000
Plataforma de Prototipagem	ESP32 DevKitC	NodeMCU v3 Lolin	Arduino MKR1000
GPIO pinos	32	17	8
FLASH	Típico 16Mb Max 32Mb	Típico 512K Max 16 MB	256 Kb
ADC pinos	16	1	7
DAC pinos	2	0	1
Vcc	3.3V	3.3V	3.3V
Consumo	80 mA	80 mA	20 mA
Dimensões	54,0 x 27,5 mm	58,0 x 31,0 mm	61,5 x 25,0 mm
Preço médio	R\$ 42,45	R\$ 38,16	R\$ 135,12

Tabela 2: Comparativo entre microcontroladores.

Microcontroladores			
Nome comercial	ESP32	ESP8266	MKR1000
Chip	ESP WROOM-32	ESP8266 MOD	ATMEL ATSAMW25
Nº Processadores	2	1	1
Processador	Tensilica Xtensa LX6	Tensilica Xtensa LX106	SAMD21 ARM@ Cortex -M0+
Arquitetura	32 Bit	32 Bit	32 Bit
CPU Freq. (Max)	240 MHz	160 MHz	48 MHz
Bluetooth	sim	não	não
SRAM	520 Kb	50 Kb	32 Kb
FLASH	4 MB	4 MB	256 Kb
GPIO pinos	36	17	8
ADC pinos	16	1	7
DAC pinos	2	-	1
ADC Resolução	12 bits	10 bits	12 bits
DAC Resolução	8 bits	-	10 bits
Interfaces	SPI, I2C, UART, I2S, CAN	SPI, I2C, UART, I2S	SPI, I2C, UART
Sensor Touch	10	-	-
Sensor Hall	1	-	-
Sensor Temperatura	1	-	-

Foram encontradas muitas informações discordantes sobre as especificações das plataformas, devido ao fato de existirem plataformas diferentes com os mesmos chips. Desta forma, é interessante destacar que o comparativo da Tabela I se refere às plataformas ilustradas na Figura 1.

Foi verificado que o microcontrolador ESP32 é superior em vários quesitos, além de possuir uma maior frequência de *clock*, entre os mais notórios é que ele possui dois processadores, enquanto o ESP8266 e o MKR1000 possuem apenas um. Em poucas categorias o ESP8266 e o MKR1000 são similares ao ESP32. Em termos gerais, o ESP8266 sobressai frente ao MKR1000, assim, pode-se inferir que as plataformas da *Espressif Systems* são superiores às plataformas da *Arduino*.

Por outro lado, o MKR1000 tem a indicação de menor consumo de energia, conforme será apresentado neste trabalho.

Além disso, o ESP32 possui sensores *on-chip* que os demais não possuem, como um sensor hall, um sensor de temperatura e dez entradas para sensores de toque (*touch*).

Assim, a escolha da melhor plataforma deve levar em consideração as características mais relevantes para a aplicação a ser desenvolvida.

B. Teste de velocidade de processamento

O objetivo deste teste foi calcular o tempo de resposta de cada microcontrolador, executando códigos com a mesma finalidade. É importante salientar que os microcontroladores possuem arquiteturas diferentes, e com isso bibliotecas e funções diferentes, sendo assim foram desenvolvidos programas distintos, porém semelhantes em relação aos objetivos a serem executados.

Contudo, foi utilizado o mesmo software de desenvolvimento, *Arduino IDE*, para todos os microcontroladores, a fim de obter uma comparação mais coerente. Inclusive é importante salientar que foi utilizado apenas um núcleo do ESP32.

O código realiza os seguintes comandos: primeiramente conecta o microcontrolador a rede sem fio, depois realizada a leitura de um sensor de temperatura externo (LM35), do fabricante *Texas Instruments*, pela entrada analógica do microcontrolador [19]. Dentro de um loop é realizada a função “*if*” e de acordo com os níveis de temperaturas pré-estabelecidos (alta ou muito alta) o microcontrolador irá enviar mensagens via aplicativo *Telegram* para um *bot* (abreviação de “*robot*”) cadastrado em um *smartphone* e também irá disponibilizar os dados em forma de gráfico no servidor da *ThinkSpeak*, que pode ser acessada via navegador *Web* [20] [21]. Tanto o *Telegram*, quanto o *ThinkSpeak* possibilitam o armazenamento de dados em nuvem, gratuitamente. Dentre outras funções do código, como reconhecimento de alarmes, ele também executa um código para buscar a informação do horário via protocolo de sincronização de relógios NTP [22]. Ao final, o código mede o tempo em milissegundos que cada microcontrolador necessitou para executar todas as iterações e comandos dentro de um *loop*. Vale salientar que foi utilizado no código um *delay* de 500 milissegundos, ou seja, 0,5 segundo.

Os tempos medidos foram inseridos na Tabela III, onde consta o tempo medido com o programa sendo executado sem nenhuma iteração e os tempos medidos com o programa realizando iterações. Foram por volta de 40 iterações (envio e recebimento de mensagens via *Telegram*), onde tivemos por volta de 30 mensagens enviadas ao *Telegram* pelo microcontrolador e 10 mensagens recebidas no microcontrolador e enviadas via *Telegram*.

Não foi possível monitorar outras variáveis analógicas, como corrente e tensão, pois o ESP8266 possui apenas uma entrada analógica, assim causando uma limitação no código.

Tabela 3: Tempo dos microcontroladores.

Plataforma	Tempo gasto (ms)		
	ESP32	ESP8266	MKR1000
Sem iterações	3634	4499	4215
Com iterações	7832	9039	8949

C. Testes de intensidade do sinal Wi-Fi

O objetivo deste teste foi medir a intensidade do sinal das três plataformas em diferentes condições, e com os resultados

realizar uma análise comparativa. Para isso, as três foram programadas com a função de *Access Point*, cada uma transmitindo em um canal diferente para evitar interferências, ligadas de maneira simultânea e realizando as medições. A primeira medição foi feita sem obstáculos a 2 metros de distância das plataformas em um ambiente interno, onde foi registrado na Figura 2.a. as amplitudes dos sinais em certo instante de tempo, e na Figura 2.b. as oscilações dos sinais durante um intervalo de tempo de aproximadamente 3 minutos.

A segunda medição seguiu o mesmo modelo da primeira, porém a uma distância de 10 metros das plataformas, com uma parede como obstáculo. A medição foi realizada com a plataforma em um ambiente interno e o *smartphone* em um ambiente externo. Para este cenário, registrou-se na Figura 3.a. as amplitudes dos sinais em certo instante de tempo e na Figura 3.b. as oscilações dos sinais durante um intervalo de tempo de aproximadamente 3 minutos.

Para a realização dos testes foi utilizado o aplicativo para *smartphone*, *WiFiAnalyzer* (Disponível para *download* na *Play Store*), que apresenta a intensidade do sinal em *dBm* (unidade de medida de intensidade do sinal). Quanto menor o valor encontrado em *dBm*, maior a potência do sinal.

Cada categoria de teste foi repetida dez vezes, a fim de se verificar a constância dos resultados obtidos. A seguir são relatados os pontos levantados sobre cada plataforma.

O ESP32, no teste a 2 metros de distância demonstrou melhores resultados, sem oscilações e apresentou um nível de amplitude de sinal superior aos demais. No teste a 10 metros e com obstáculos, o ESP32 foi um pouco inferior ao MRK1000, mantendo a constância e uma amplitude satisfatória do sinal.

O MKR1000, no teste a 10 metros de distância e com obstáculo, obteve os melhores resultados, não apresentando muitas oscilações, além de obter um nível de amplitude superior aos demais. No teste a 2 metros e com obstáculos, apresentou amplitude de sinal próxima àquela do ESP32, mantendo uma constância e amplitude desejáveis.

Quanto ao ESP8266, apresentou comportamento semelhante aos demais, tanto a 2 metros quanto a 10 metros de distância com obstáculo, observando que a 10 metros a potência do sinal foi naturalmente menor. Apresentou resultados adequados no teste de amplitude, porém resultados bastante insatisfatórios na constância do sinal, pois apresentou grandes oscilações de amplitude.

Figura 2: (a) Amplitude a 2 metros. (b) Oscilações a 2 metros.

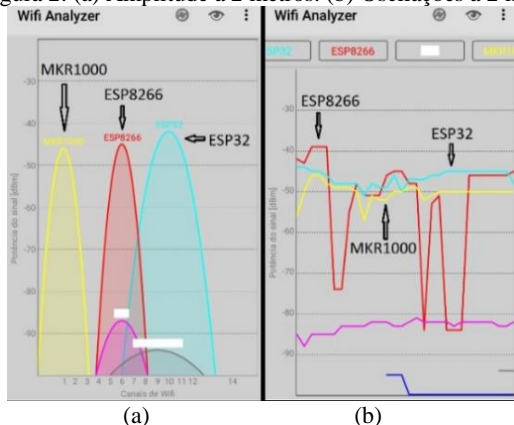
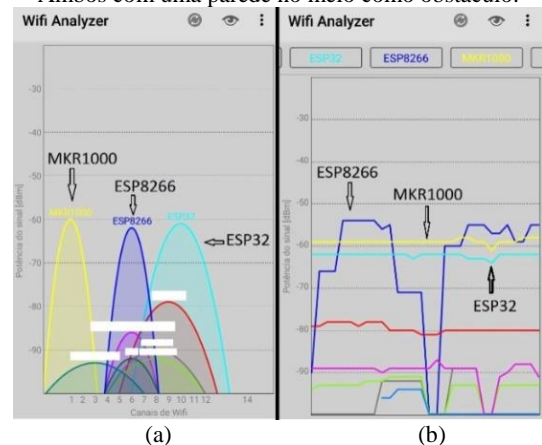


Figura 3: (a) Amplitude a 10 metros. (b) Oscilações a 10 metros. Ambos com uma parede no meio como obstáculo.



Resumindo, o ESP32 e o MKR1000 apresentaram resultados semelhantes no teste de intensidade de sinal. Já o ESP8266 apresentou grandes oscilações na intensidade do seu sinal, comportamento não desejado para a aplicação em questão.

D. Teste de consumo de energia

Com o intuito de verificar os dados de consumo inseridos na Tabela I, foram realizados testes para verificação do consumo das plataformas, sob as mesmas condições de uso, utilizando um dispositivo (*USB Tester*) que mede a tensão a corrente da porta *USB* à qual as plataformas serão conectadas. Para medir o consumo de energia de cada plataforma elas foram programadas com a função de *Access Point* e realizada as leituras, conforme Figura 4.

As plataformas MKR1000 e ESP8266 apresentaram a mesma leitura de corrente, 50 mA. Já a plataforma ESP32 apresentou uma leitura mais alta, 110 mA.

Esses valores não coincidem exatamente com os valores dos *datasheets*, já que o consumo de cada plataforma depende do programa que está sendo executado. Os resultados observados mostram que, programados como *Access Point*, o MKR1000 e o ESP8266 apresentam menor consumo, enquanto que o ESP32 apresenta um maior consumo. Essa questão se dá pelo fato de o ESP32 possuir um maior número de funções.

Figura 4: Teste de consumo utilizando o USB Tester.



E. Testes sensor hall ESP32

Este teste foi realizado apenas com o ESP32, pois é o único dos três que possui sensor hall integrado.

Ainda utilizando o *software* Arduino IDE, foi carregado o programa que faz a leitura do sensor *hall* através da função *hallRead()*. O sensor *hall* é um dispositivo que, sob o efeito de um campo magnético, produz uma tensão proporcional [23].

Com esse programa em funcionamento foram realizados vários testes para verificar como o sensor *hall* do ESP32 se comporta na presença de campos magnéticos, inclusive se consegue captar interferências de campos magnéticos de motores.

Para este teste foi utilizado um intervalo de 1 segundo entre medições.

As primeiras leituras foram realizadas sem a aplicação de campo magnético ao ESP32, cujos resultados podem ser observados na Figura 5.

Em seguida foi aplicado um campo magnético utilizando um ímã permanente. Ao se aproximar do sensor, o polo norte do ímã, nota-se que os valores das leituras variam positivamente. Ao se aproximar o polo sul do ímã, nota-se que os valores das leituras se tornam negativos. Em ambos os casos um maior valor em módulo representa um maior valor de campo aplicado, conforme Figura 6.

Figura 5: Leituras do sensor hall sem a aplicação de campo magnético.

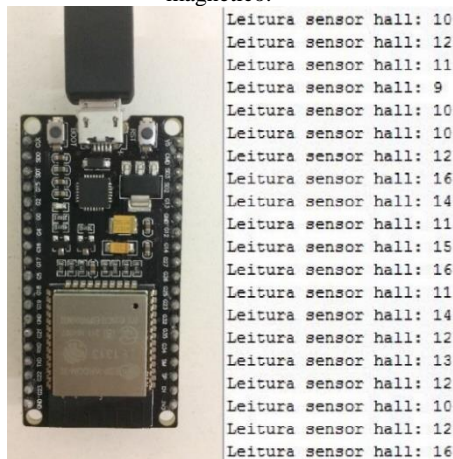


Figura 6: Leituras do sensor hall com a aplicação de campo magnético através de um ímã permanente, variando seus polos.

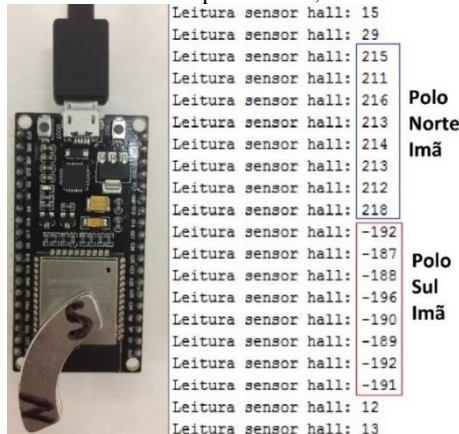


Figura 7: Leituras do sensor hall com a aplicação de campo magnético através do acionamento de um motor.



Também foi realizado um teste para verificar a capacidade do ESP 32 detectar campo magnético gerado por um motor elétrico. Para tal, foi acionado um motor de 1/2 CV sob o ESP32, porém as leituras de campo magnético não foram alteradas, conforme Figura 7. Acredita-se que o sensor não alterou sua leitura por se encontrar externo a carcaça do motor.

V. DISCUSSÕES

Baseado nos resultados dos testes e análises feitas entre as três plataformas *WiFi* (ESP32, ESP8266 e MKR100), o ESP32 mostrou-se mais promissor para ser utilizado em aplicações com *IoT*. Além de possuir melhores especificações na maioria dos quesitos, este apresentou um bom desempenho no teste de velocidade, mesmo utilizando apenas um processador, e demonstrou ótimos resultados no teste de intensidade do sinal, semelhante aos resultados do MKR1000, sendo este último inferior nos testes de velocidade de processamento e nos recursos disponíveis na plataforma. No teste de consumo de energia, o ESP32 demonstrou um maior consumo que os demais, algo que já era esperado, devido ao mesmo possuir um número maior de funções.

Os testes com o sensor hall do ESP32 não apresentaram resultados satisfatórios que justifiquem sua utilização para adquirir variáveis de campo magnético de motores.

Além do sensor hall o ESP32 também possui outros sensores internos ao chip, como sensores de toque (*touch*) e de temperatura, que não foram testados neste trabalho.

Em se tratando de disponibilidades de informações técnicas, o MKR1000 apresenta um número maior de fontes para pesquisa, além de maior facilidade de programação. Já as informações das plataformas da *Espressif Systems* são mais escassas e divergentes, dificultando a busca por informações.

VI. CONCLUSÕES

Conclui-se que, a partir dos dados levantados nesse trabalho, a plataforma ESP32 apresenta um conjunto satisfatório de recursos e um ótimo desempenho relativo para uso em aplicações de *IoT*.

Novas plataformas e microcontroladores são desenvolvidos a cada ano e novos testes podem ampliar as comparações obtidas nesse trabalho, como por exemplo o monitoramento de um número maior de variáveis.

REFERÊNCIAS

- [1] Andreola, A. T., Senter, M. J. D., Todero, E. L. and Cardoso, G. “Low cost data acquisition system for wind prospecting”. *12th IEEE International Conference on Industry Applications (INDUSCON)*, Curitiba, pp. 1-6, 2016.
- [2] Carvalho, D. P. D. Monitoramento wireless de eficiência e condição de operação de motores de indução trifásicos. Dissertação de Mestrado (UFU). Uberlândia, MG, p. 116, 2010.
- [3] Goundar, S. S., Pillai, M. R., Mamun, K. A., Islam, F.R. and Deo, R. “Real Time Condition Monitoring System for Industrial Motors”. *2nd Asia-Pacific World Congress on Computer Science and Engineering (APWC on CSE)*, Nadi, pp. 1-9, 2015.
- [4] Kolban, N. *Kolban's Book on ESP32 & ESP8266*. Texas, USA, October, 2016.
- [5] Arduino home page (2019). *What is Arduino?*. Acedido em 21 de Março de 2019, em: <http://www.arduino.cc>.
- [6] Oliveira, V. G., Niizu, F. Y. and Basseto, F., *Internet das Coisas*, Revista O Setor Elétrico, 132ª Edição, São Paulo/SP, 2017.
- [7] Cunha, M. J., Almeida, M. B., Fernandes, R. F. and Carrijo, R. S. “Proposal for an IoT architecture in industrial processes”. *12th IEEE International Conference on Industry Applications (INDUSCON)*, Curitiba, pp. 1-7, 2016.
- [8] Lampkin, V., Leong, W. T., Olivera, L., Rawat, S., Subrahmanyam, N., Xiang, R., *Building smarter planet solutions with mqtt and ibm websphere mq telemetry*, Redbooks, 7ª Edição, IBM, 2012.
- [9] Souza, D. J. *Desbravando o PIC: Baseado no microcontrolador PIC 16F84*, Érica, 5ª Edição, São Paulo, 2000.
- [10] Cunha, A. *Sistemas Embarcados*, Revista Saber Eletrônica, 414ª Edição, Brasil, 2007.
- [11] NodeMCU home page (2018), *NodeMcu Connect Things EASY*. Acedido em 22 de Março de 2019, em: https://www.nodemcu.com/index_en.html.
- [12] Arduino software page (2019), *Software*, Acedido em 22 de Março de 2019, em: <https://www.arduino.cc/en/software>.
- [13] Espressif Systems (2019), *Datasheet: ESP-32 DevKitC*, Acedido em 22 de Março de 2019, em: <https://www.espressif.com/en/products/devkits/esp32-devkitc/resources>.
- [14] Espressif Systems (2016). *Datasheet: NodeMCU v3 LoLin*, Acedido em 22 de Março de 2019, em: http://handsontec.com/pdf_learn/esp8266-V10.pdf.
- [15] Arduino store page (2019). *Arduino MKR1000 WiFi*, Acedido em 22 de Março de 2019, em: <https://store.arduino.cc/usa/arduino-mkr1000>.
- [16] Espressif Systems (2018). *ESP8266EX Datasheet*, Acedido em 20 de Março de 2019, em: https://www.espressif.com/sites/default/files/documentation/0a-esp8266ex_datasheet_en.pdf.
- [17] Espressif Systems (2019b). *ESP32 WROOM Series*, Acedido em 20 de Março de 2019, em: <https://www.espressif.com/en/products/modules/esp32/resources>.
- [18] Arduino uploads (2016). *MKR1000 Schematic*, Acedido em 22 de Março de 2019, em: <https://www.arduino.cc/en/uploads/Main/MKR1000-schematic.pdf>.
- [19] Texas Instruments (2017). *Datasheet: LM35 Precision Centigrade Temperature Sensors*, Acedido em 22 de Março de 2019, em: <https://www.ti.com/lit/ds/symlink/lm35.pdf>.
- [20] Telegram home page (2019). *Telegram: a new era of messaging*, Acedido em 20 de Março de 2019, em: <https://telegram.org>.
- [21] ThingSpeak home page (2019). *Understand Your Things: The open IoT platform with MATLAB analytics*, Acedido em 20 de Março de 2019, em: <https://thingspeak.com>.
- [22] NTP home page (2019). *Saiba mais sobre o NTP*, Acedido em 20 de Março de 2019, em: <https://ntp.br>.
- [23] Paun, M. A., Sallèse, J. M. and Kayal, M. “Hall Effect Sensors Design, Integration and Behavior Analysis”. *Journal of Sensor and Actuator Networks*, Basileia, Suíça, 2013.