



## Desenvolvimento de um Acionamento de Motor de Aeromodelo “brushless” utilizando a Plataforma do Arduino e Protocolo MQTT

Lucas Corrêa (TIA: 31613705), Aleixo Afonso (TIA: 31852246), Willian França

Universidade Presbiteriana Mackenzie (UPM)  
Rua da Consolação, 930 Consolação, São Paulo - SP, 01302-907 – Brazil

[lgmpcorrea@gmail.com](mailto:lgmpcorrea@gmail.com) , [aleixoafonso@hotmail.com](mailto:aleixoafonso@hotmail.com), [willian@mackenzie.br](mailto:willian@mackenzie.br)

**Abstract.** *This project deals with the development of a brushless model motor drive that uses the communication protocol (MQTT) to control the motor speed via wi-fi. For the execution of the work, the Arduino setup was used, together with an ESC driver and a model airplane engine for the construction of the drive. For data communication, a NodeMCU development kit was used, which plays the role of Publisher, and a Wi-Fi internet module ESP-8266 that acts as a Subscriber.*

**Resumo.** *Esse projeto trata do desenvolvimento de um acionamento de motor de aeromodelo brushless que utiliza o protocolo de comunicação (MQTT) para o controle de velocidade do motor via wi-fi. Para a execução do trabalho, foi utilizada a plataforma do Arduino, junto a um driver ESC e um motor de aeromodelo para a construção do acionamento. Para a comunicação de dados, foi utilizado um kit de desenvolvimento NodeMCU que desempenha o papel de Publisher, e um modulo de internet wi-fi ESP-8266 que atua como Subscriber.*

### 1. Introdução

O projeto de qualquer acionamento de motor geralmente é realizado em função das características do motor que se pretende acionar. O motor utilizado neste projeto é um motor síncrono de ímãs permanentes (“brushless DC Motor”) que possui as seguintes características: alta eficiência, maior densidade de potência, maior simplicidade de construção maior confiabilidade de operação, ampla faixa ampla de operação, controle preciso de velocidade, baixa pulsação de torque, sendo utilizados em inúmeras aplicações, tanto em baixa, como em alta potência, que vão desde braços robóticos, até geradores de energia eólicos. Outra vantagem associada às máquinas síncronas é a excitação do circuito

de campo, que é realizada por ímãs localizados nas superfícies dos polos do rotor, não havendo a necessidade de enrolamentos no rotor, anéis coletores ou escovas [1].

Como desvantagem, estas máquinas não podem operar diretamente da rede elétrica, ou seja, elas necessitam de um acionamento ou “driver” que estabeleça uma estratégia de comutação para as fases do motor. Associado a isso, existe ainda um complicador, dada as inúmeras novas aplicações para este tipo de motor na atualidade, principalmente no que tange a dispositivos portáteis, tais como drones, aeromodelos, ferramentas portáteis, etc, faça-se necessário à concepção de acionamentos com maior poder de conectividade, de modo a atender requisitos de melhor controle de velocidade e posição, telemetria de dados, e desempenho na autonomia de carga de baterias. Dentro deste contexto, se estabelece este trabalho que visa o desenvolvimento de um acionamento de motor, utilizando a plataforma do Arduino de modo que o mesmo possa operar de maneira satisfatória, atendendo as devidas condições de eficiência e potência do motor, fornecendo maior poder de conectividade através protocolo de comunicação MQTT (*Message Queue Telemetry Transport*) [1-2].

O MQTT consiste em um protocolo de mensagens simples, criado para comunicação M2M (*Machine to Machine*) que exige poucos recursos em termos de processamento e banda, sendo indicado para dispositivos embarcados que utilizam a tecnologia de Internet das Coisas IoT (*Internet of Things*). De modo geral, uma comunicação MQTT é composta por três partes: publishers (quem disponibiliza informações), subscribers (quem recebe informações) e Broker (servidor MQTT, localizado na nuvem). Os publishers enviam informação para o Broker, e os subscribers recebem informação do Broker, sendo que o Broker gerencia a troca de informações entre os mesmos. Ou seja, a parte complexa do trabalho fica a cargo do Broker, deixando os demais sistemas embarcados livres para gerenciar outras operações [3-4].

## **2. Materiais e Métodos**

Para a construção do acionamento e das interfaces de comunicação foram utilizados os componentes enumerados a seguir:

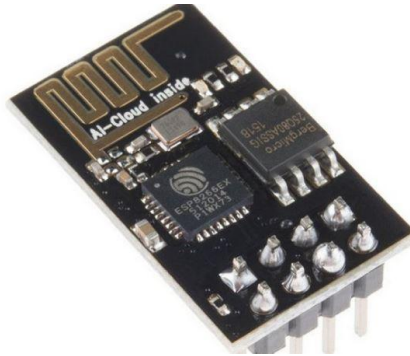
### **2.1. Materiais**

- 1) **Placa Arduino Uno:** a placa do Arduino é um kit de desenvolvimento equipado com o microcontrolador ATMEGA328 e utiliza de uma linguagem #C++ (IDE Arduino) para que objetos interativos possam ser manipulados.



**Figura 1. Placa do Arduino Uno [5]**

- 2) **Módulo ESP8266:** é um módulo wireless que permite a conexão da placa do Arduino às redes wi-fi padrão.



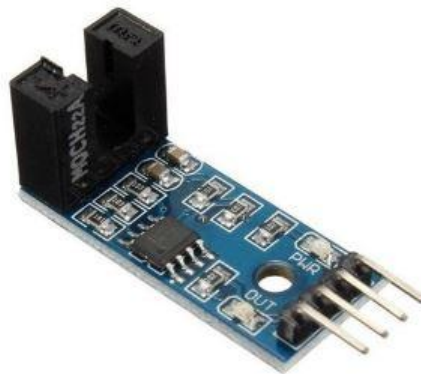
**Figura 2. Módulo ESP-8266 [6]**

- 3) **Modulo LCD I2C:** é utilizado para otimizar o numero de pinos disponíveis na placa do Arduino. Além disso, com este módulo o controle do display LCD será realizado através de apenas 2 pinos.



**Figura 3. Módulo LCD I2C [7-8]**

- 4) Sensor de velocidade:** é utilizado para realizar medições de rotação de motores, contagem de pulsos e como controlador de posicionamento de servo motores. Pode ser utilizado com um disco encoders, onde as aberturas do disco irão interromper o feixe de luz infravermelho e enviar um sinal ao microcontrolador.



**Figura 4. Sensor de Velocidade [9]**

- 5) Display LCD 16 x 2:** possui 16 colunas por 2 linhas e backlight verde. É equipado com o controlador HD44780.



**Figura 5. Display LCD 16 x 2 [10]**

- 6) **Protoboard:** é uma ferramenta para a montagem de circuitos eletrônicos. Possui 400 pontos e em sua parte inferior, sendo 100 pontos de distribuição e 300 pontos de conexão terminal.



**Figura 6. Protoboard [11]**

- 7) **Motor Brushless A2212 2200KV:** é um motor de corrente contínua, sem escovas, também conhecido como Motor BLDC. Pesa apenas 40g, sendo eficiente, com alta capacidade de carga, livre de manutenção, ruído reduzido, e vida útil mais prolongada. Possui alto torque, com a capacidade de girar hélices diretamente, sem o uso de redução mecânica, sendo ideal para projetos de aeromodelismo e drones.



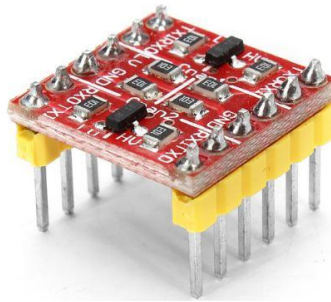
**Figura 7. Motor Brushless A2212 2200KV [12]**

- 8) Potenciômetro multivoltas 10k $\Omega$ :** é um componente de precisão, pois possibilita giros maiores que 360° em seu eixo. Desta forma, sua resistência varia de 0 a 10 k $\Omega$  lentamente, possibilitando maior precisão no ajuste.



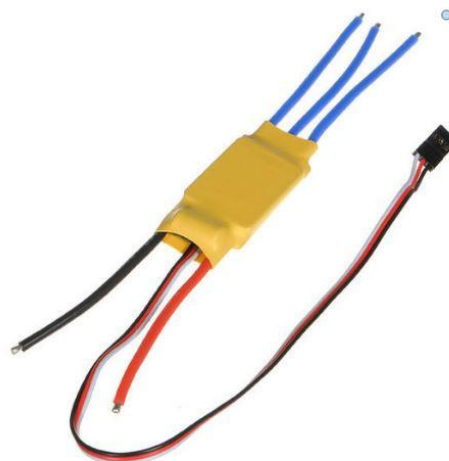
**Figura 8. Potenciômetro multivoltas 10k $\Omega$  [13]**

- 9) Conversor de tensão bidirecional 5 V / 3,3 V:** Módulo com função bidirecional que reduz de 5 V para 3,3 V e aumenta de 3,3 V para 5 V, possuindo duas entradas e duas saídas em cada lado.



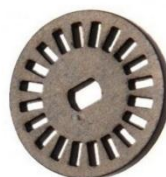
**Figura 9. Conversor de tensão [14]**

**10) Driver ESC (Electronic Speed Control) 12 Volts:** é uma placa eletrônica equipada com um microcontrolador e um regulador de tensão dedicados para o controle de motores brushless, utilizados em aeromodelos, quadcopteros e projetos em geral.



**Figura 10. Driver ESC 12 volts [15]**

**11) Disco de Encoder:** deve ser utilizado junto a um sensor de velocidade e/ou chave óptica, possibilitando que os mesmos realizem a leitura do número de voltas executadas pelo eixo do motor. Este disco é utilizado nos mais diversos equipamentos para definir a velocidade do deslocamento de um objeto ou de uma engrenagem.



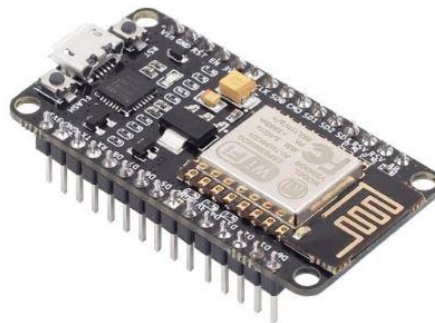
**Figura 11. Disco de Encoder [16]**

- 12) Adaptador USB para Módulo Wi-Fi ESP8266 ESP-01:** é utilizado para conectar o módulo wi-fi ESP8266 ao computador, de modo que possa ser realizada a programação do mesmo.



**Figura 12. Adaptador USB para Módulo Wi-Fi ESP8266 ESP-01 [17]**

- 13) Módulo Wi-Fi ESP8266 NodeMcu ESP-12:** é um kit de desenvolvimento que combina o chip ESP-12 com módulo Wi-Fi ESP8266. A programação do kit pode ser realizada através da linguagem LUA ou através da IDE do Arduino, utilizando a comunicação via cabo micro-usb.



**Figura 13. Módulo Wi-Fi ESP8266 NodeMcu ESP-12 [18]**

## **2.2 Métodos**

O acionamento de motor proposto é composto por uma placa do Arduino Uno conectada a um driver ESC de 12 V e, este por sua vez, está conectado a um motor de aeromodelo brushless, modelo A2212 2200KV. Além disso, o acionamento é composto por um display LCD (16x2) e um Módulo LCD I2C, ambos inseridos numa protoboard,



utilizados para indicação da velocidade do motor. Para a medição da rotação do eixo do motor, o acionamento utiliza um conjunto de um disco de encoder, com 10 aberturas, e sensor de velocidade.

O controle de velocidade do motor ocorre de maneira remota (wireless) e prevê a utilização do protocolo MQTT para sua execução. De modo geral, uma comunicação MQTT é composta por três partes: publishers (quem disponibiliza informações), subscribers (quem recebe informações) e Broker (servidor MQTT, localizado na nuvem). Os publishers enviam informação para o Broker, e os subscribers recebem informação do Broker, sendo que o Broker gerencia a troca de informações entre os mesmos.

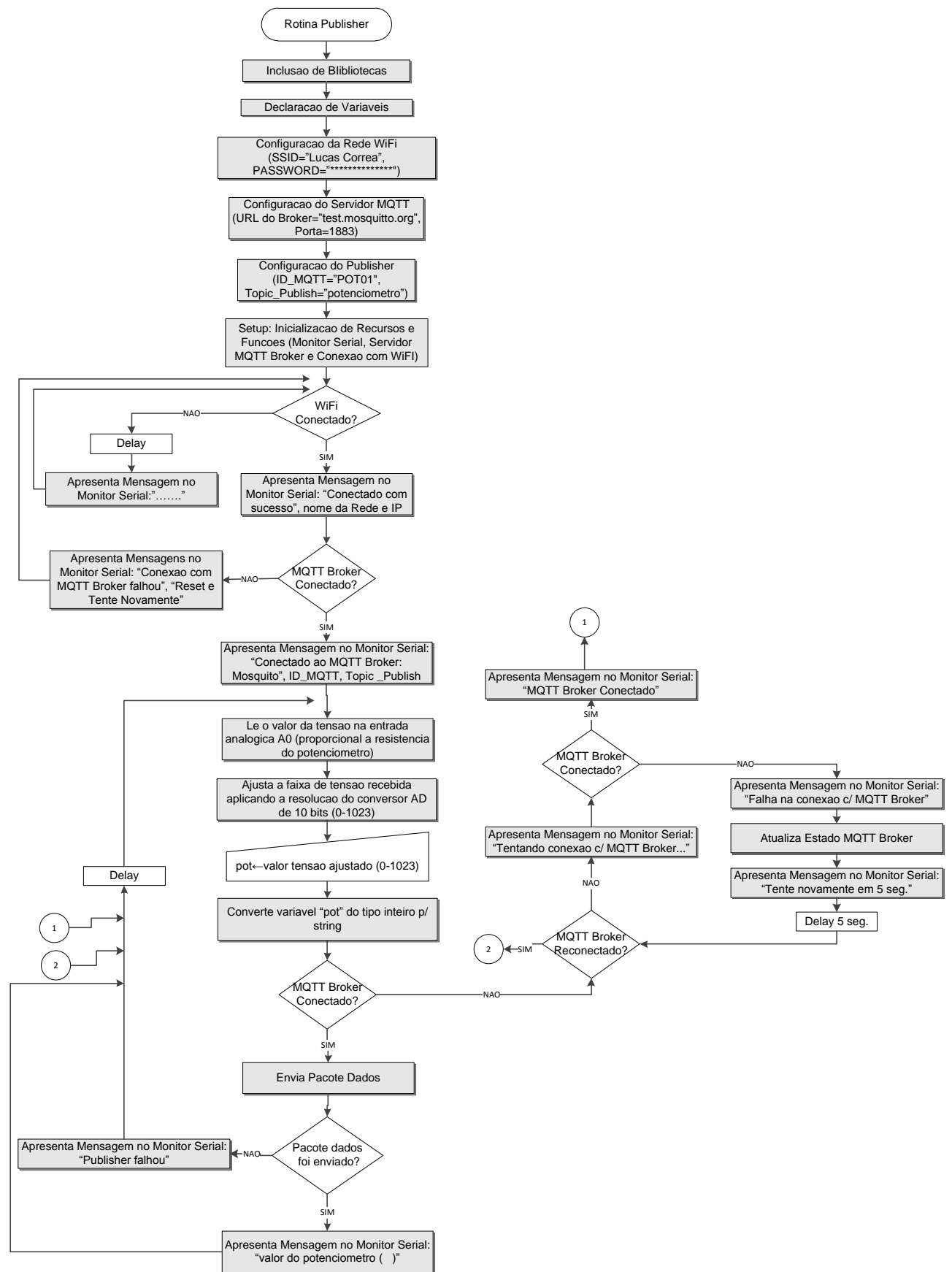
Para a comunicação de dados do projeto, foi utilizado um kit de desenvolvimento NodeMCU ESP-12F associado a modulo wi-fi ESP8266, e um potenciômetro multivoltas de 10 k $\Omega$  que desempenham o papel de Publisher. Além disso, foi utilizado um modulo de internet wi-fi ESP-8266 e um adaptador USB que atua como Subscriber, e é alimentado por um conversor de tensão de 5,0 V / 3,3 V. No caso do Broker, foi utilizado um serviço de nuvem publico (Broker Mosquitto), de modo a reduzir os custos do trabalho.

A implementação do hardware foi dividida em três blocos, denominados: Placa do Acionamento de Motor, Placa do Publisher e Placa do Subscriber, cujos componentes já foram detalhados anteriormente e são apresentados mais adiante na sessão 3 (Resultados). De maneira similar ao hardware, para o desenvolvimento do software, foram desenvolvidos três scripts diferentes, destinados a cada um dos dispositivos programáveis (microcontrolados) integrantes do projeto, tais como: Arduino Uno, NodeMCU e ESP-8266. Desta forma, foi desenvolvido um script para a função de Publisher (Placa do Publisher “NodeMCU”), um script para a função de Subscriber (Placa do Subscriber “ESP-8266”) e um script para o acionamento de motor propriamente dito (Placa do Acionamento de Motor “Arduino Uno”). A programação de todos os dispositivos programáveis foi realizada em linguagem #C++, através da IDE do Arduino.

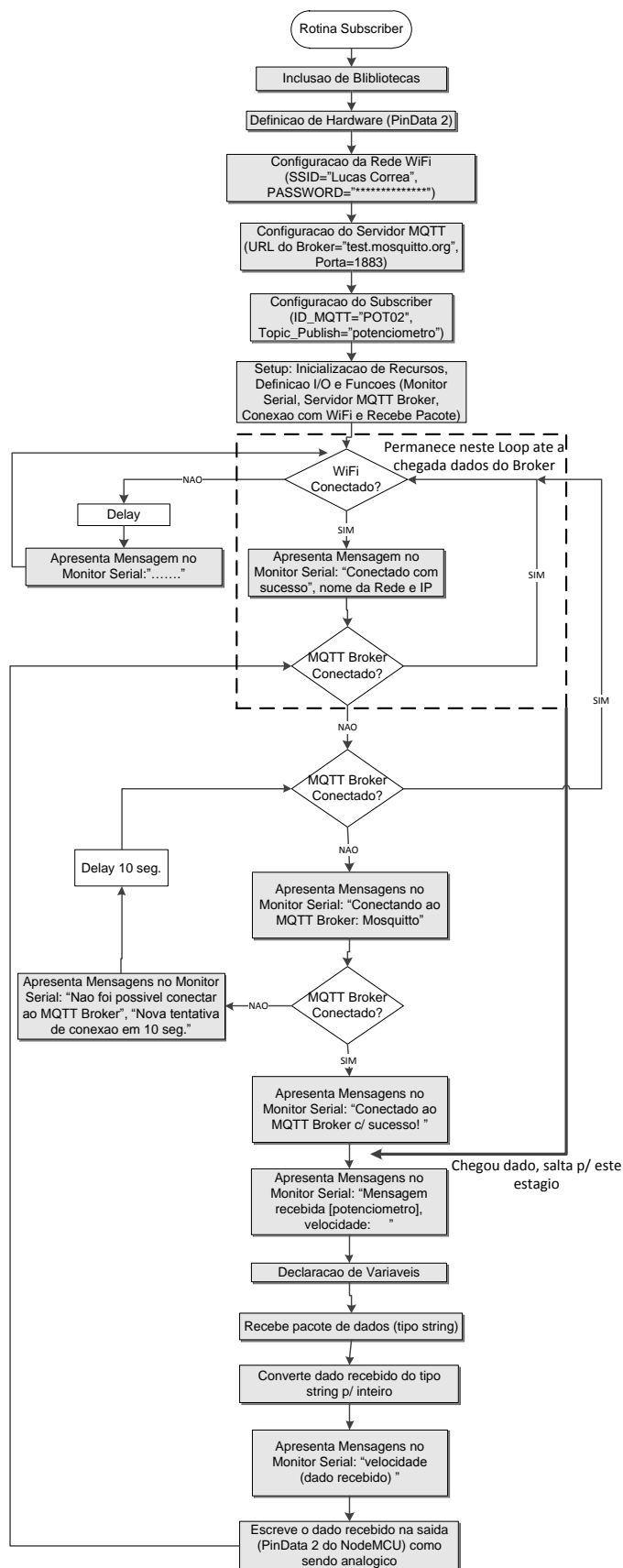
De modo sucinto, o software do projeto pode ser descrito do seguinte modo: a rotina do Publisher (NodeMCU) estabelece inicialmente a comunicação com rede Wifi e com Broker escolhido. A seguir, esta rotina estabelece um loop que lê os dados do potenciômetro na entrada analógica A0 do NodeMCU (valor da tensão convertida pelo AD de 10 bits “0-1023”), converte o dado para tipo string e envia os mesmos via wireless para o Broker. A rotina do Subscriber (modulo wi-fi ESP8266) também estabelece inicialmente a

comunicação com rede Wifi e com Broker escolhido e permanece neste loop, verificando a comunicação constantemente. Quando um dado chega do Broker, a rotina de comunicação é interrompida, o dado é lido e convertido para o tipo inteiro e colocado a disposição na entrada analógica A0 da placa do Arduino Uno, onde será recebido pela rotina de Acionamento de Motor. A rotina de Acionamento de Motor estabelece uma rotina de interrupção externa (int1) que monitora os pulsos no pino 3 do Arduino (sensor de velocidade e disco do encoder). Desta forma, esta rotina estabelece uma rotina de loop que cria uma base de tempo de 1 seg., onde dentro desta base de tempo, ela calcula a velocidade instantânea do eixo do motor, através da contagem dos pulsos do encoder, lê o valor do dado recebido do Subscriber (valor da tensão convertida pelo AD de 10 bits “0-1023”), realiza o escalonamento dos valores lidos para valores aceitáveis pelo driver ESC, carrega estes valores para o driver ESC, fazendo com que o motor entre em operação. Todas essas rotinas do software podem ser analisadas em maiores detalhes, respectivamente, nas Figuras 14, 15 e 16.

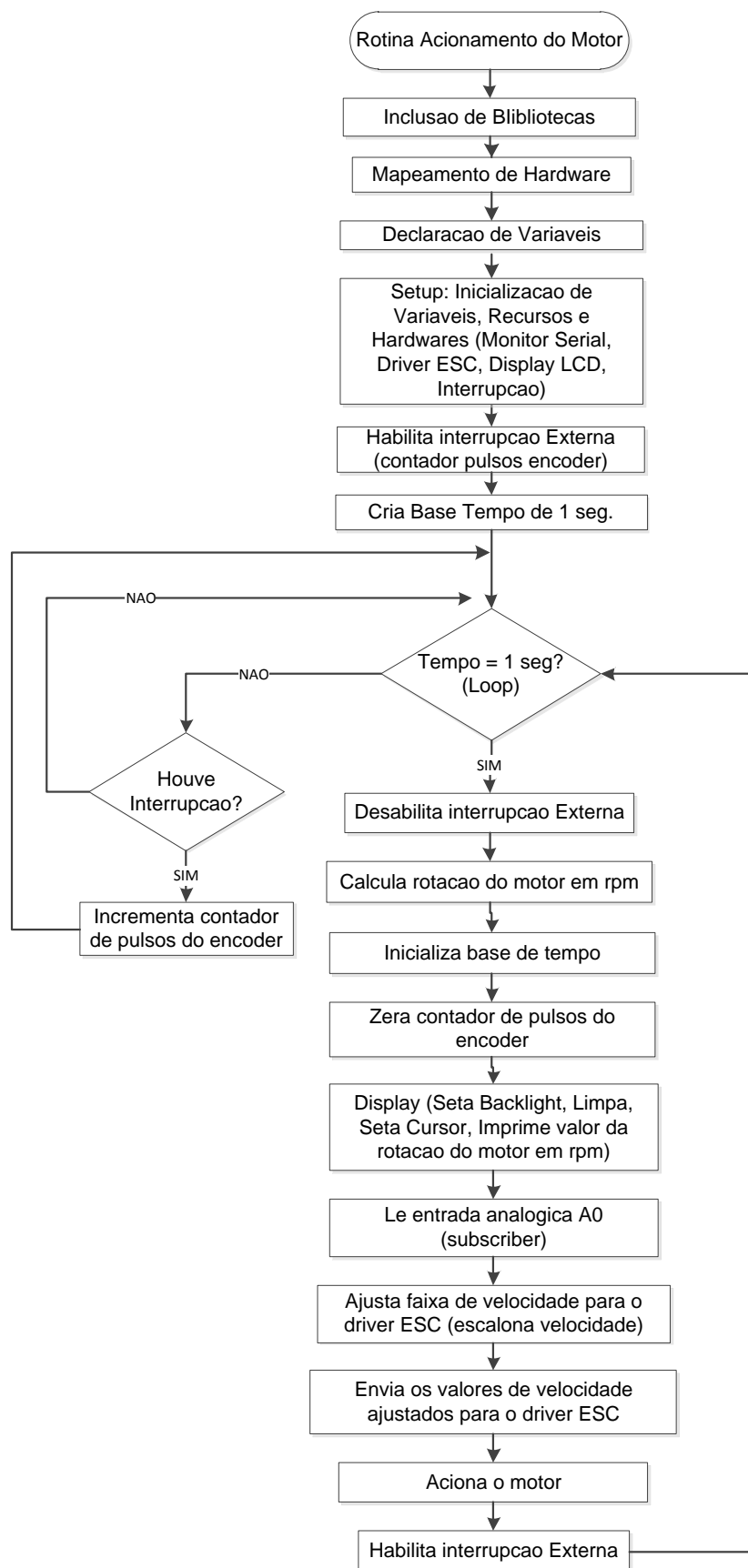
A Figura 17 apresenta um diagrama esquemático de interligação das interfaces utilizadas no projeto.



**Figura 14. Rotina do Publisher (NodeMCU).**



**Figura 15. Rotina do Subscriber (ESP-8266).**



**Figura 16. Rotina do Acionamento do Motor (Arduíno Uno).**

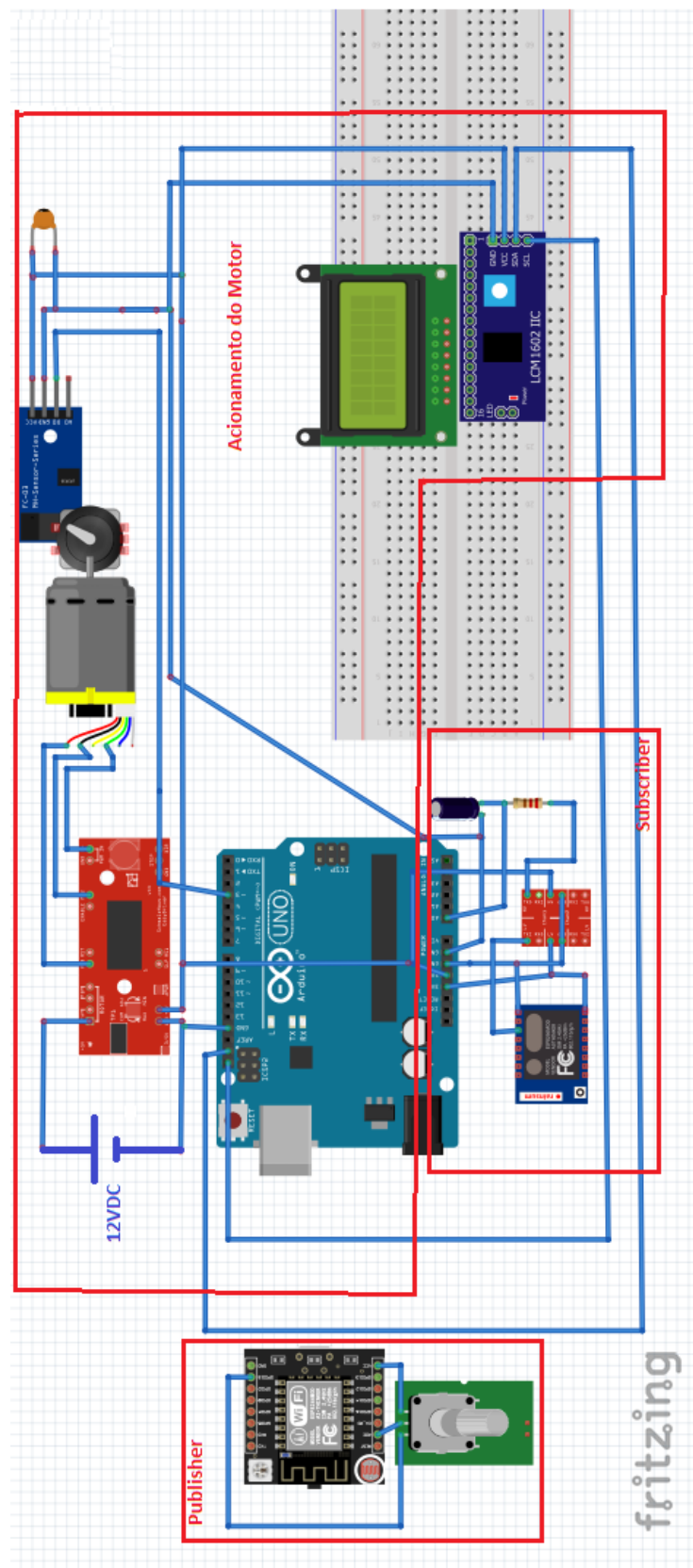
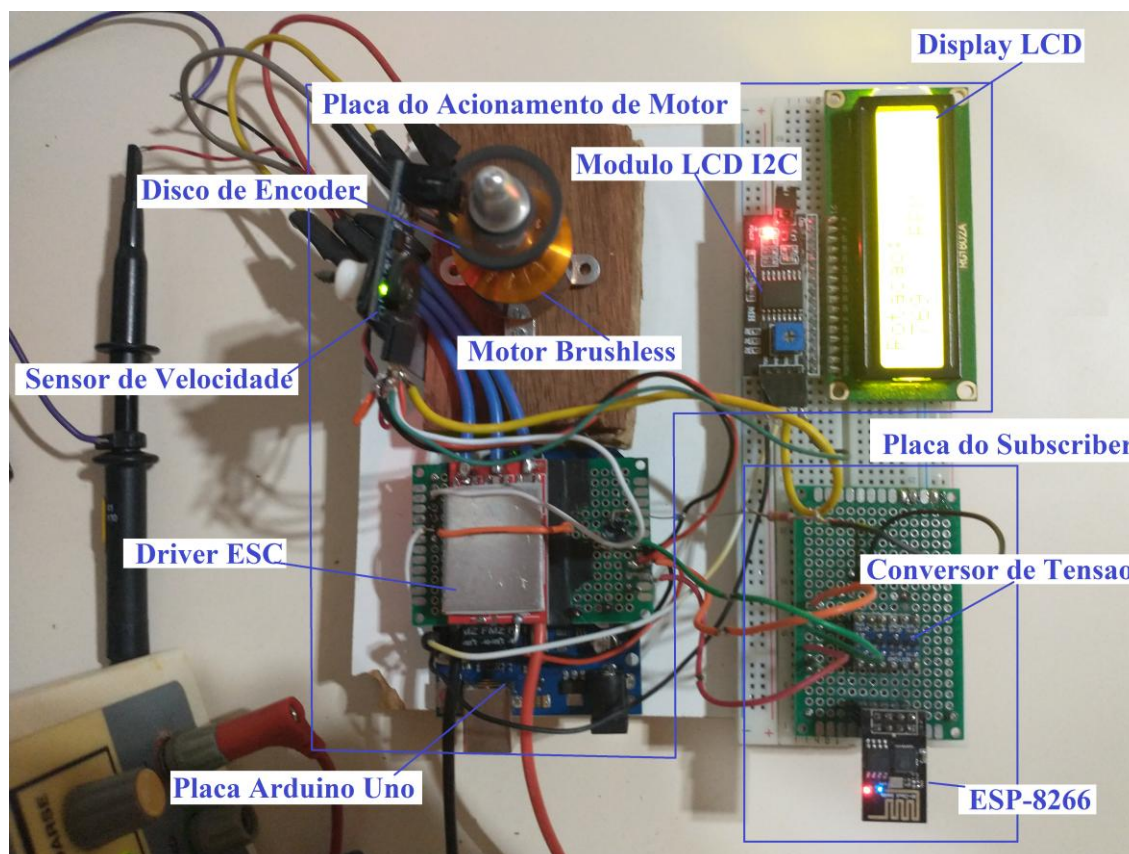


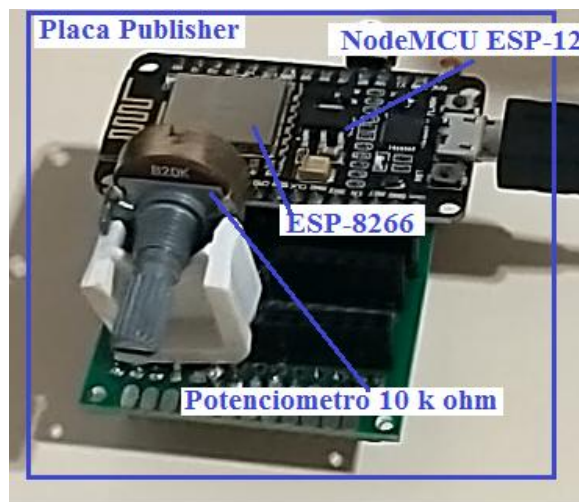
Figura 17. Diagrama Esquemático de Interligação das Interfaces.

### 3. Resultados

Foi implementada a construção do hardware do acionamento proposto que utiliza os dispositivos e componentes mencionados e, já detalhados, na seção 2 (Materiais e Métodos). Conforme descrito anteriormente, o acionamento de motor completo pode ser dividido em três blocos: Placa do Acionamento de Motor, Placa do Publisher e Placa do Subscriber. As Figuras 18 e 19 apresentam as fotos do acionamento de motor completo implementado, dividido em blocos, com os principais dispositivos e componentes utilizados.

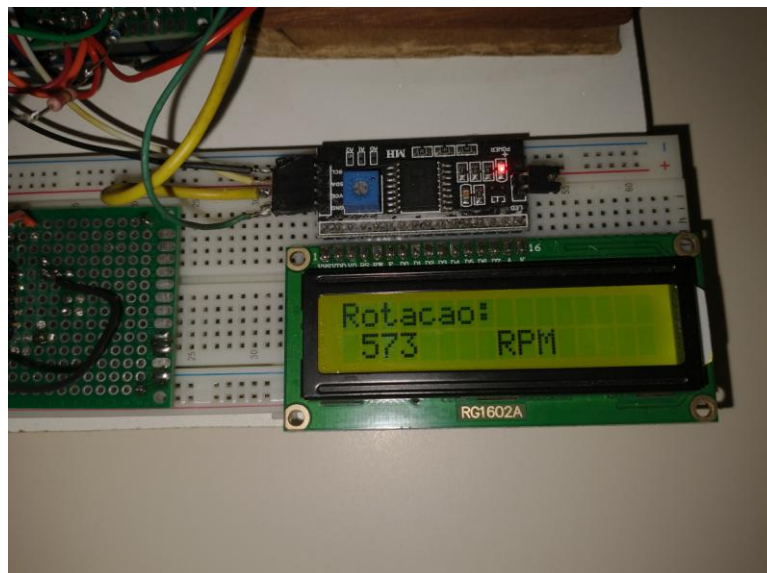


**Figura 18. Acionamento de Motor implementado (no detalhe: placa do Acionamento de Motor, placa do Subscriber, principais dispositivos e componentes).**



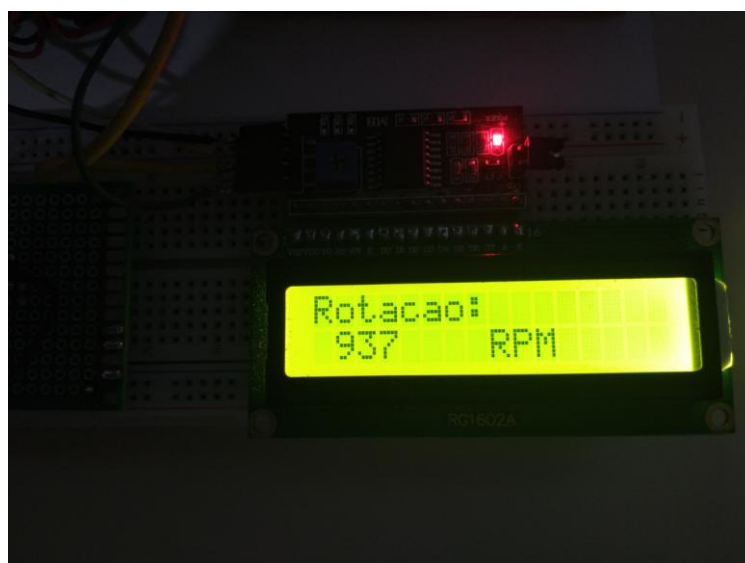
**Figura 19. Acionamento de Motor implementado (no detalhe: placa do Publisher, principais dispositivos e componentes).**

Para comprovar o funcionamento do acionamento, podem ser observadas as Figuras 20 e 21 que apresentam, respectivamente, o display LCD quando da leitura da velocidade instantânea do eixo do motor, operando na faixa de 500 RPM e 1000 RPM.



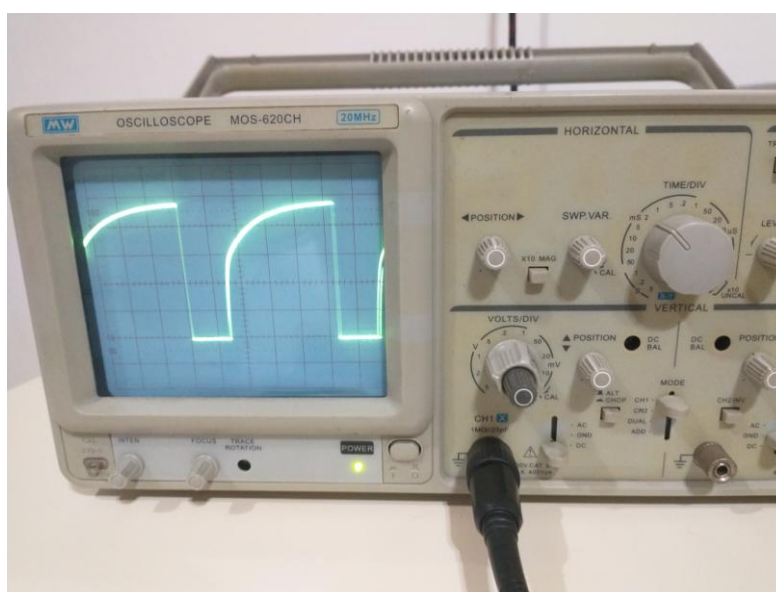
**Figura 20. Display LCD quando da leitura da velocidade do motor operando na faixa de 500 RPM.**



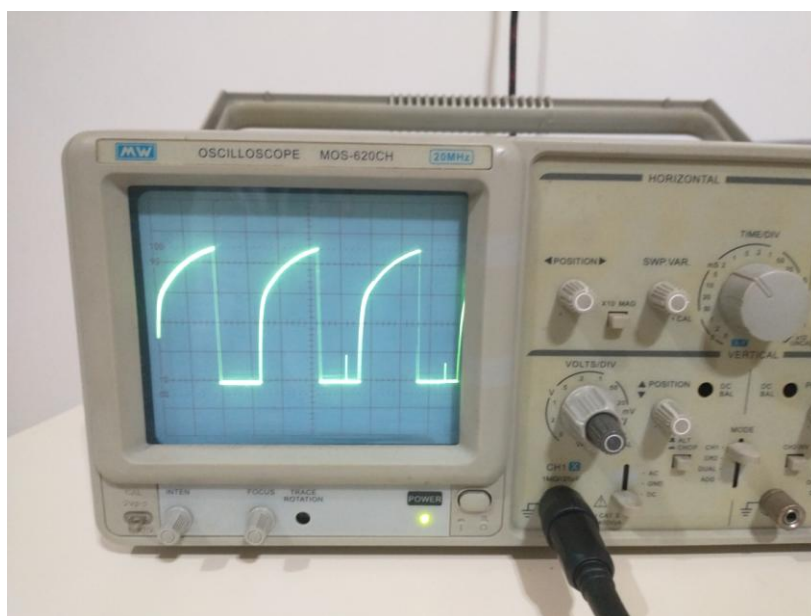


**Figura 21. Display LCD quando da leitura da velocidade do motor operando na faixa de 1000 RPM.**

De modo a validar o algoritmo que calcula a velocidade instantânea do eixo do motor, foram selecionadas inicialmente duas faixas de velocidade (500 RPM e 1000 RPM). As Figuras 22 e 23 apresentam, respectivamente, as formas de onda do pulso medido na saída digital do sensor de velocidade, obtidas com um osciloscópio, quando o motor apresentava no display a velocidade de 573 RPM e 937 RPM.



**Figura 22. Forma de onda do pulso medido na saída digital do sensor de velocidade, medido com osciloscópio, quando o motor apresentava no display a velocidade de 573 RPM.**



**Figura 23. Forma de onda do pulso medido na saída digital do sensor de velocidade, medido com osciloscópio, quando o motor apresentava no display a velocidade de 937 RPM.**

As formas de onda do pulso medido na saída digital do sensor de velocidade (Figuras 22 e 23) foram medidas com um osciloscópio analógico de 20 MHz, na escala de 1 V/div., e base de tempo de 2 ms/div., mostrando que ambos os sinais possuem um amplitude de 5 V. Em relação a largura dos pulsos, observamos que a Figura 22 apresenta um período de 10,4 ms e, a Figura 23 um período de 6,4 ms. Uma vez obtidos os períodos dos pulsos, e aplicando a equação abaixo (equação 1), é possível obter a velocidade instantânea do eixo do motor em RPM.

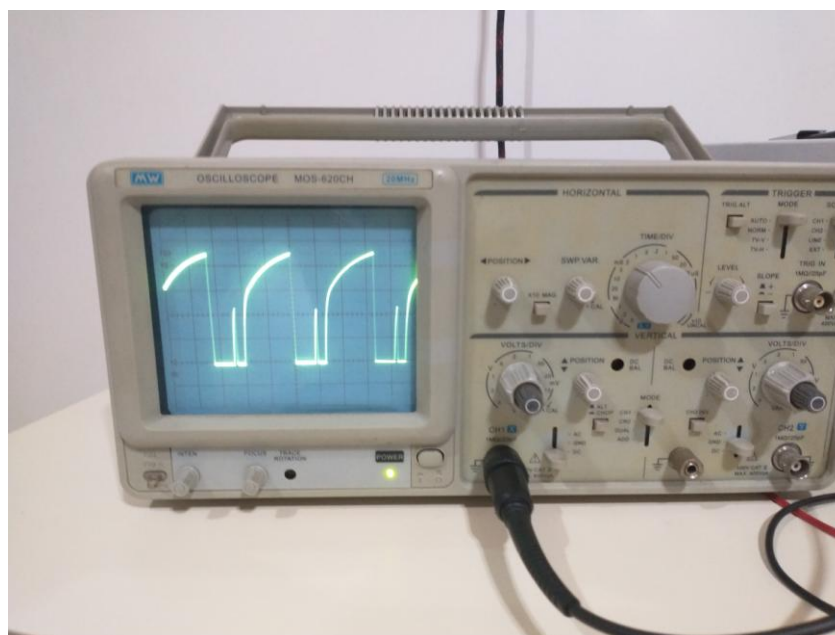
$$W \text{ RPM} = (1/(\text{período do pulso medido} \cdot 10)) \cdot 1000 \cdot 60 \quad (1)$$

Resolvendo esta equação para ambos os pulsos medidos, obtemos: 576 RPM para a Figura 22 e 937 RPM para a Figura 23. Estes valores referem-se valores medidos. Comparando os valores medidos com os calculados e apresentados no display LCD, obtemos uma diferença menor que 1%, constatando a validade dos algoritmos implementados.

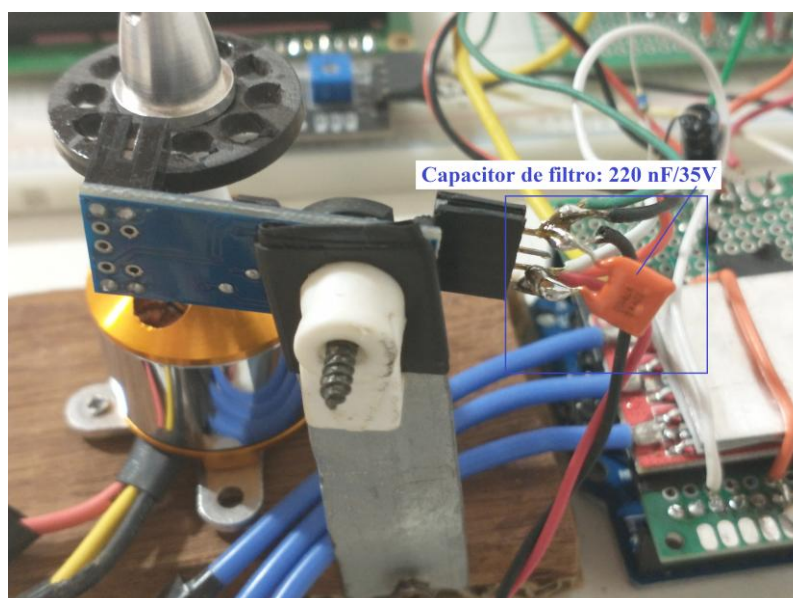
Em relação à implementação do hardware do acionamento de motor proposto inicialmente neste trabalho, foram necessárias algumas modificações em função de problemas apresentados e que são enumerados adiante.

Os principais problemas enfrentados durante a execução do projeto foram relacionados abaixo:

- I) Ruído nos pulsos do disco encoder, lidos a partir da saída do sensor de velocidade que causavam oscilação no período dos pulsos e, por conseguinte, nos valores de velocidade instantânea calculados e apresentados no display LCD. Para atenuar este problema, foi adicionado um capacitor de filtro de 220 nF / 35 V na saída do sensor de velocidade, como apresentado na Figura 24.

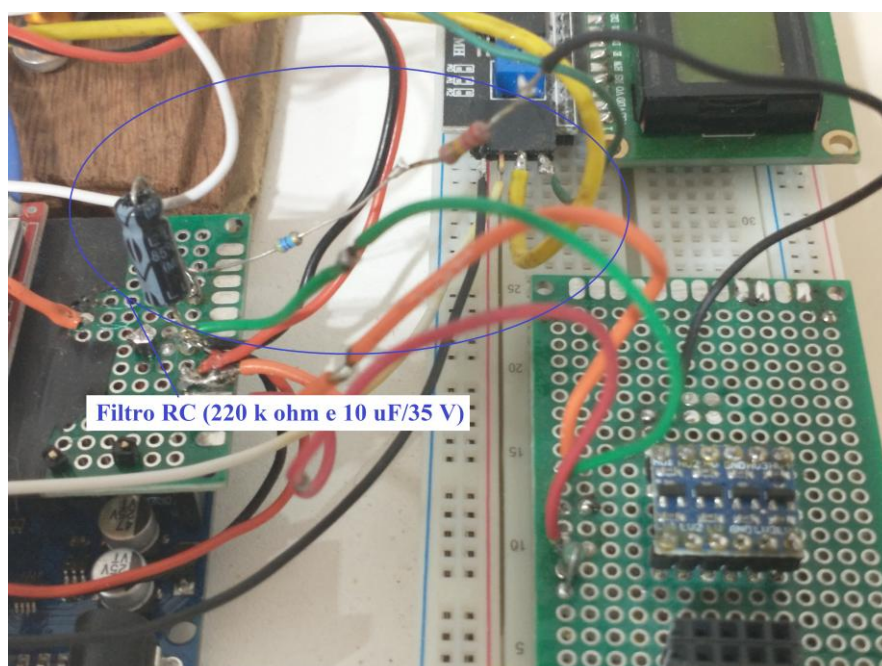


**Figura 24. Ruído nos pulsos do disco de encoder, lidos a partir da saída do sensor de velocidade.**



**Figura 25. Capacitor de filtro de 220 nF / 35 V adicionado na saída do sensor de velocidade para atenuar ruído.**

- II) Baixa integridade do sinal enviado pelo Subscriber (ESP-8266) para a placa do Arduino UNO, ou seja, o dado (tensão) enviado pelo Subscriber apresentava valor incorreto na entrada analógica A0 do Arduino, pois se tratava de um sinal pulsado. Desta forma, foi necessário adicionar um Filtro RC de valor elevado (220 k ohm e 10 uF/ 35 V) ao sinal (Figura 26), de modo a se obter um nível médio de tensão. Com a adição do circuito, o nível do sinal recebido passou a ser compatível com o esperado.



**Figura 26. Filtro RC (220 k ohm e 10 uF / 35 V) adicionado na entrada analógica A0 do Arduino de modo a compatibilizar o nível médio de tensão enviado pelo Subscriber.**

## **Links do Projeto:**

<https://github.com/LucasCorrea02/Projeto---Acionamento-do-Motor-brushless.git>

<https://www.youtube.com/watch?v=9s0NFQQ2V6Y> (video 1)

<https://www.youtube.com/watch?v=JEQdiz1ngSM> (video 2)

<https://www.youtube.com/watch?v=nQN6e2kJnZw> (video 3)

<https://www.youtube.com/watch?v=qImrF7B-YmE> (video 4)

## **4. Conclusões**

Com o desenvolvimento deste trabalho nos aprofundamos na programação do Arduino, NodeMCU, ESP8266 e protocolos MQTT de comunicação. Além disso, aprendemos a manusear diversos componentes como: display LCD, módulo I2C, motor brushless DC, drivers ESC, sensor de velocidade, etc. A proposta do trabalho foi concluída com êxito, uma vez que foi apresentado todo o hardware e software de um acionamento de motor brushless que pode ter sua velocidade controlada de maneira remota (wireless) através da utilização do protocolo MQTT. Para a comprovação dos resultados obtidos, foram apresentadas fotos do acionamento construído e formas de onda medidas que podem ser comparadas com valores calculados pelo algoritmo e que estão disponibilizados na sessão de resultados.

A utilização do conjunto de ferramentas computacionais (plataforma do Arduino / ESP-8266, associado ao protocolo de comunicação MQTT) fornece a principal vantagem associada a este projeto, uma vez que tornou possível o desenvolvimento de uma aplicação IoT razoavelmente complexa, num tempo muito menor do que se conseguiria sem as mesmas. Como desvantagem, podemos citar o fato de que, para a construção do acionamento proposto, foram utilizados três dispositivos microcontrolados (Arduino Uno, ESP-8266 e NodeMCU). Esta condição agregou não somente maior dificuldade para o desenvolvimento do trabalho, como causou também um aumento do custo do projeto. As mesmas funções poderiam ser alcançadas através da utilização de apenas uma placa Arduino com ESP-8266 integrados e um celular com aplicativo “Blynk” instalado, por exemplo.



Como melhorias em trabalhos futuros, poderíamos recomendar a utilização de uma plataforma mais econômica, como citado no parágrafo anterior. Além disso, durante as medições de velocidade do motor, muitas vezes nos deparamos com oscilações na velocidade do motor, uma vez que não existia um controle em malha fechada para a estabilização da mesma. Desta forma, outra sugestão seria a implementação de uma malha fechada de velocidade com a utilização do Arduino.

## 5. Referências

- [1] Schmitz, Claudio. (2017) “Projeto e Otimização de Motores BLDC de Imãs Permanentes Superficiais”, Dissertação de Mestrado, <https://repositorio.ufsc.br/xmlui/bitstream/handle/123456789/175323/345508.pdf?sequence=1&isAllowed=y>, Janeiro.
- [2] Control Brushless Motor Using Arduino. RZtronics, 2016. Disponível em: <<https://www.youtube.com/watch?v=DTOK6CgXRXg>>. Acesso em: 20 de set. de 2020.
- [3] BERTOLETI, Pedro. Controle e Monitoramento IoT com NodeMCU e MQTT. Filipeflop, 2016. Disponível em: <<https://www.filipeflop.com/blog/controle-monitoramento-iot-nodemcu-e-mqtt/>>. Acesso em: 07 de abr. de 2021.
- [4] OLIVEIRA, Euler. Como usar com Arduino – Módulo WiFi ESP8266 ESP-01. Blogmasterwalkershop, 2018. Disponível em: <<https://blogmasterwalkershop.com.br/arduino/como-usar-com-arduino-modulo-wifi-esp8266-esp-01>>. Acesso em: 15 de nov. de 2020.
- [5] Arduino Uno. Filipeflop, 2021. Disponível em: <<https://www.filipeflop.com/produto/placa-uno-r3-cabo-usb-para-arduino/>>. Acesso em: 07 de abr. de 2021.
- [6] Módulo ESP8266. Curto Circuito, 2021. Disponível em: <[https://www.curtocircuito.com.br/modulo-wifi-esp8266-esp-01.html?gclid=EAIAIqObChMIy47rpe7v7wIVSYCRCh3\\_iQryEAQYAYABEgLKRPD\\_BwE](https://www.curtocircuito.com.br/modulo-wifi-esp8266-esp-01.html?gclid=EAIAIqObChMIy47rpe7v7wIVSYCRCh3_iQryEAQYAYABEgLKRPD_BwE)>. Acesso em: 07 de abr. de 2021.
- [7] MESSINA, Ana Paula. Protocolo de comunicação I2C com Arduino e NodeMCU. Tecnicas, 2019. Disponível em: <<https://tecnicas.com/protocolo-de-comunicacao-i2c-com-arduino-e-nodemcu/>>. Acesso em: 17 de out. de 2020.
- [8] Módulo LCD I2C. Filipeflop, 2021. Disponível em: <<https://www.filipeflop.com/produto/modulo-serial-i2c-para-display-lcd-arduino/>>. Acesso em: 07 de abr. de 2021.
- [9] Sensor de velocidade. Filipeflop, 2021. Disponível em: <<https://www.filipeflop.com/produto/sensor-de-velocidade-encoder/>>. Acesso em: 07 de abr. de 2020.
- [10] Display LCD. Filipeflop, 2021. Disponível em: <<https://www.filipeflop.com/produto/display-lcd-16x2-backlight-verde/>>. Acesso em: 07 de abr. de 2020.

- [11] Protoboard. Filipeflop, 2021. Disponível em:<  
<https://www.filipeflop.com/produto/protoboard-400-pontos/>>. Acesso em: 07 de abr. de 2020.
- [12] Motor Brushless A2212 2200KV. Saravati, 2021. Disponível em:  
<<https://www.saravati.com.br/motor-brushless-a2212-2200kv-6t>>. Acesso em: 07 de abr. de 2020.
- [13] Potenciometro multivoltas 10kΩ. Baú da Eletrônica, 2021. Disponível em: <  
<https://www.baudaeletronica.com.br/potenciometro-multi-volta-10k.html>>. Acesso em: 07 de abr. de 2020.
- [14] Conversor de tensão bidirecional 5 V / 3,3 V. Filipeflop, 2021. Disponível em:  
<<https://www.filipeflop.com/produto/conversor-de-nivel-logico-33-5v-bidirecional/>>. Acesso em: 07 abr. 2021.
- [15] Driver ESC (Electronic Speed Control) 12 Volts. Filipeflop, 2021. Disponível em:  
<<https://www.filipeflop.com/produto/esc-30a-brushless-com-bec-interno-2a5v-aeromodelos/>>. Acesso em: 07 de abr. de 2021.
- [16] Disco Encoder. Baú da Eletrônica, 2021. Disponível em:<  
[https://www.baudaeletronica.com.br/disco-encoder.html?gclid=EAIaIQobChMI7rqMo-zv7wIVj4KRCh38jQN9EAYYASABEgJxM\\_D\\_BwE](https://www.baudaeletronica.com.br/disco-encoder.html?gclid=EAIaIQobChMI7rqMo-zv7wIVj4KRCh38jQN9EAYYASABEgJxM_D_BwE)>. Acesso em: 07 de abr. de 2021.
- [17] Adaptador USB para Módulo Wi-Fi ESP8266 ESP-01. Filipeflop, 2021. Disponível em  
< <https://www.filipeflop.com/produto/adaptador-usb-para-modulo-wifi-esp8266-esp-01/>>. Acesso em: 07 de abr. de 2021.
- [18] Módulo Wi-Fi ESP8266 NodeMcu ESP-12. Filipeflop, 2021. Disponível em <  
<https://www.filipeflop.com/produto/modulo-wifi-esp8266-nodemcu-esp-12/>>. Acesso em: 07 de abr. de 2021.