



## Desenvolvimento de um acionamento de motor de aeromodelo “brushless” utilizando a plataforma do Arduino

Lucas Corrêa, Sandra Costa, Willian França

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

lgmpcorrea@gmail.com, costa.sp1sandra@gmail.com, willian@mackenzie.br

**Abstract.** *This project deals with the development of a BLDC motor drive. For the execution of the work, the arduino platform associated with the ESP-8266 wi-fi internet module using the MQTT protocol was used.*

**Resumo.** *Esse projeto trata do desenvolvimento de um acionamento de motor de aeromodelo brushless. Para a execução do trabalho foi utilizada a plataforma do arduino associada ao modulo de internet wi-fi ESP-8266 através do protocolo MQTT.*

### 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, drones, 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. 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. 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 conectividade

através da troca de dados a partir de uma placa Wi-fi ESP8266 e protocolo MQTT de mensagens.

## 2. Materiais e Métodos

Para a construção do acionamento foram utilizados os componentes enumerados a seguir:

- 1) **Placa Arduino Uno:** O Arduino é uma plataforma eletrônica como entrada e saída, e que utiliza de uma linguagem simples para que objetos interativos possam ser manipulados. Ele pode ser utilizado no desenvolvimento de objetos interativos independentes, conectados a um computador, a uma rede, ou até mesmo diretamente com a internet, para enviar e receber dados.



Figura 1. Placa do Arduino Uno

- 2) **Módulo ESP8266:** O módulo wireless ESP8266 permite que você conecte o seu Arduino às redes wi-fi (wireless) padrão 802.11 b/g/n, que são as redes mais comuns utilizadas para acessar a internet.

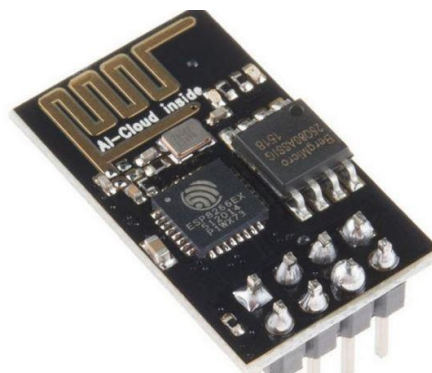


Figura 2. Módulo ESP-8266

- 3) **Módulo LCD I2C:** Muitas vezes a quantidade de pinos disponíveis no seu Arduino é limitada, pois após conectar display e sensores sobram poucos pinos para uso. Contudo com este **módulo** de interface I2C você poderá controlar o seu display lcd usando apenas 2 pinos.

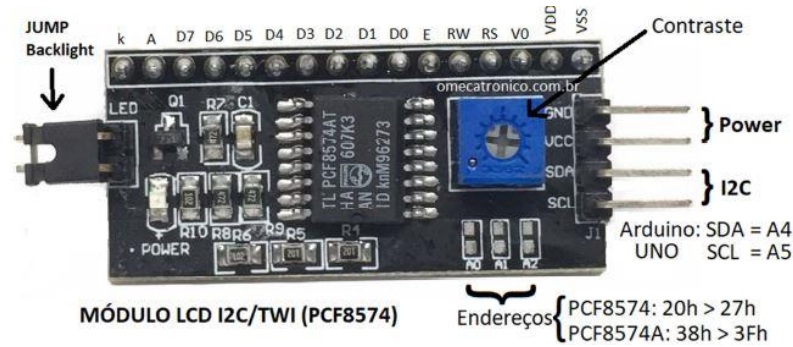


Figura 3. Módulo LCD I2C

- 4) **Sensor de velocidade:** O sensor de velocidade é utilizado para realizar medições de rotação de motores, contagem de pulsos e como controlador de posicionamento. É indicado para utilização com um disco encoder, onde as ranhuras do disco irão interromper o feixe de luz infravermelho e enviar um sinal ao microcontrolador.

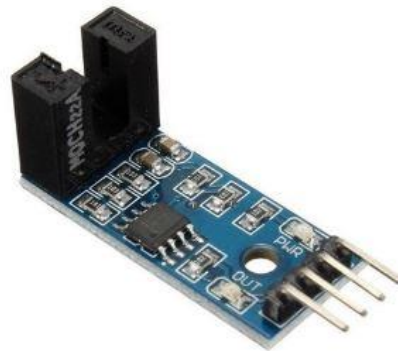


Figura 4. Sensor de Velocidade

- 5) **Display LCD 16 x 2:** Display Lcd com 16 colunas por 2 linhas, backlight verde. Possui o controlador HD44780 usado em toda indústria de LCD's como base de interface.



Figura 5. Display LCD 16 x 2

- 6) **Protoboard:** Protoboard é uma excelente ferramenta para a montagem de circuitos eletrônicos, sendo uma maneira rápida, fácil e prática para montar seus projetos. Possui 400 pontos e em sua parte inferior há um adesivo que permite colá-lo em uma superfície isolante. São 100 pontos de distribuição e 300 pontos de conexão terminal. Possui coordenadas coloridas para facilitar a visualização na hora de colocar seu componente.



Figura 6. Protoboard

**7) Motor Brushless A2212 2200KV:** O Motor Brushless A2212 2200KV é um motor de corrente contínua sem escovas, livre de manutenção, também conhecido como Motor BLDC.

Pesando apenas 40g, o Motor Brushless A2212 2200KV é pequeno e extremamente eficiente, com alta capacidade de carga, livre de manutenção, ruído reduzido, e vida útil mais longa (devido a ausência de desgaste da escova), possui alto torque, com a capacidade de girar hélices diretamente sem o uso de redução mecânica, minimizando as perdas por atrito, não utiliza partes móveis que sofrem desgaste, e não adiciona sobrepeso tornando seu aeromodelo muito mais leve, ideal para seus projetos de aeromodelismo elétrico e drones.



Figura 7. Motor Brushless A2212 2200KV

**8) Pontenciometro multivotas 10k:** é um componente de precisão, pois ele possibilita darmos um giro maior que  $360^\circ$  em seu eixo (enquanto um potenciômetro rotativo comum só chega a  $300^\circ$ ), com isso sua resistência varia lentamente possibilitando maior precisão no ajuste.



Figura 8. Potenciômetro multivotas 10k

**9) Conversor de tensão:** Módulo com função bidirecional que reduz de 5V para 3,3V e aumenta de 3,3V para 5V, possui duas entradas e duas saídas em cada lado, compatível também com dispositivos de 2,8V e 1,8V

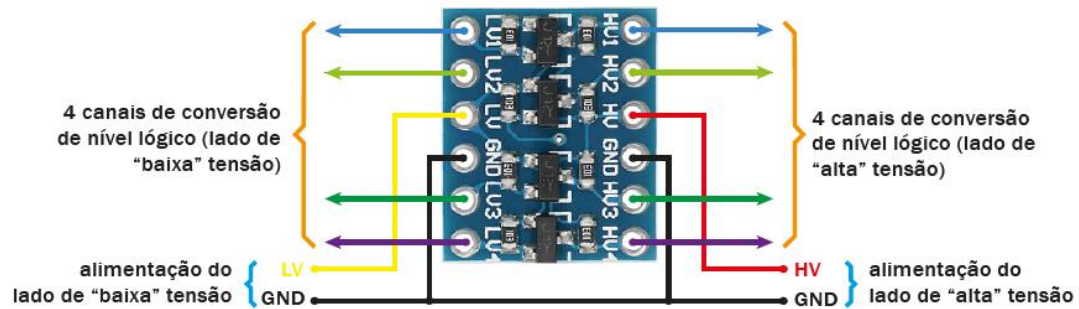


Figura 9. Conversor de tensão

**10) Driver esc 12 volts:** utilizado para controle de motores Brushless usados em Aeromodelos, Quadcopteros e projetos em geral.



Figura 10. Driver esc 12 volts

**11) Disco de Encoder para Sensor de Velocidade:** O Disco Encoder possibilita que um sensor de velocidade / chave óptica possa fazer a leitura de quantas voltas foram dadas por um motor. Este disco é utilizado nos mais diversos equipamentos para definir a velocidade do deslocamento de um objeto ou uma engrenagem.



Figura 11. Disco de Encoder para Sensor de Velocidade



### 3. Resultados

Os resultados deste trabalho são a própria construção do hardware e do software do acionamento proposto.

Hardware:

A figura abaixo apresenta todo o hardware implementado, onde foram utilizados todos os componentes já apresentados na sessão de materiais.

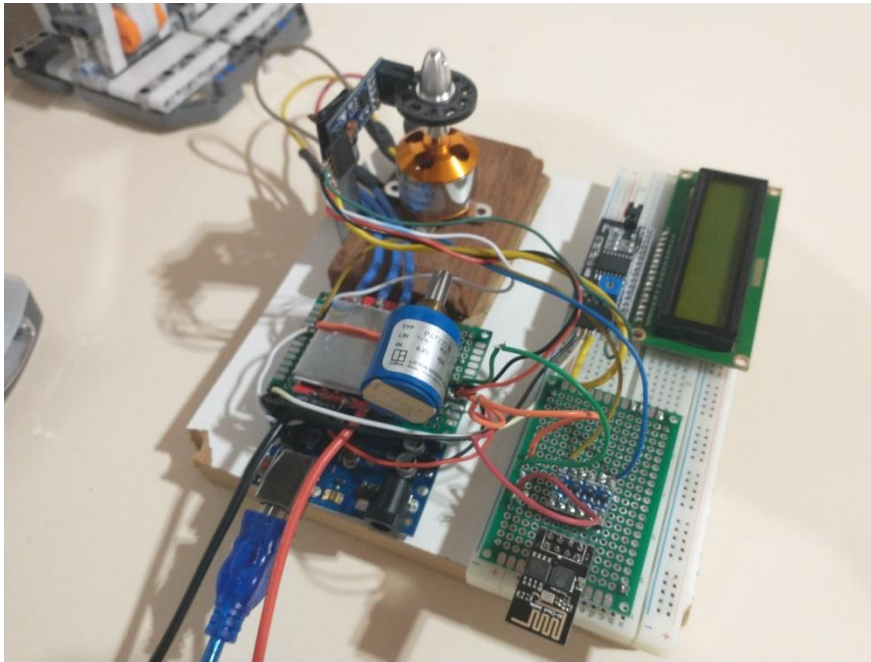


Figura 1. acionamento completo

Para comprovar o funcionamento do acionamento, pode ser observada a figura abaixo que mostra o display LCD quando da leitura da velocidade do motor.



Figura 2. Velocidade do motor mostrada no display



## Software:

A figura abaixo apresenta o código em #C++ do software implementado para o acionamento.

## Código desenvolvido:

```
// --- Mapeamento de Hardware ---
#define pino_D0 3
// --- Inclusão de Biblioteca ---
#include <Servo.h>
#include <Wire.h>
#include <LiquidCrystal_I2C.h>
// --- Criação Classe Servo ---
Servo esc;
// Set the LCD address to 0x27 for a 16 chars and 2 line display
LiquidCrystal_I2C lcd(0x27, 16, 2);
// --- Variáveis Globais ---
int      rpm;
volatile byte pulsos;
unsigned long timeold;
unsigned int pulsos_por_volta = 10;
// --- Interrupção ---
void contador()
{
    //Incrementa contador
    pulsos++;
}
// --- Configurações Iniciais ---
void setup()
{
    esc.attach(6);           //Specify the esc signal pin as D6
    esc.writeMicroseconds(1000); //Initialize signal to 1000
    lcd.begin ();
    pinMode(pino_D0, INPUT);
```

```

//Interrupcao 1 - pino digital 3
//Aciona o contador a cada pulso
attachInterrupt(1, contador, FALLING);
pulsos = 0;
rpm = 0;
timeold = 0;
}
void loop()
{
//Atualiza contador a cada segundo
if (millis() - timeold >= 1000)
{
//Desabilita interrupcao durante o calculo
detachInterrupt(1);
rpm = (60 * 1000 / pulsos_por_volta ) / (millis() - timeold) * pulsos;
timeold = millis();
pulsos = 0;
//Mostra o valor de RPM no display
lcd.setBacklight(HIGH);
lcd.clear();
lcd.setCursor(0, 0);
lcd.print("Rotacao: ");
lcd.setCursor(1, 1);
lcd.print(rpm, DEC);
lcd.setCursor(8, 1);
lcd.print("RPM");
//delay(1000);
//Varre valor do Potenciometro
int val;
val= analogRead(A0);          //Read input from analog pin A0 and store in val
val= map (val, 0,1023, 1000, 2000);    //Mapping the val to minimum and maximum
input

```

```

    esc.writeMicroseconds(val);      //using val as the signal to esc
    //Habilita interrupcao
    attachInterrupt(1, contador, FALLING);
}
}

```

O programa desenvolvido executa as seguintes tarefas:

A) rotina que lê o valor do potenciômetro, realiza o escalonamento dos valores lidos p valores aceitáveis pelo driver ESC, carrega estes valores para o driver ESC para operação do motor;

B) rotina de interrupção (int1) a cada segundo que monitora os pulsos no pino 3 do sensor de velocidade e disco do encoder. A interrupção é desligada p cálculo da velocidade angular instantânea do eixo do motor, atualização do display e leitura do potenciômetro;

#### **4. Conclusão**

Com o desenvolvimento deste trabalho nos aprofundamos na programação do arduino e do ESP8266 utilizando o protocolo MQTT. Além disso, aprendemos a manusear diversos componentes como: display LCD, módulo I2C, motores brushless DC, drivers ESC, sensor de velocidade, etc. De modo geral a proposta do trabalho foi concluída, uma vez q foi apresentado um acionamento do motor em funcionamento, o que pode ser comprovado na sessão de resultados.

#### **5. Referências:**

<https://www.youtube.com/watch?v=DTOK6CgXRXg>

<https://tecdicas.com/protocolo-de-comunicacao-i2c-com-arduino-e-nodemcu/>

<https://cadernodelaboratorio.com.br/memoria-eeeprom-externa-com-o-arduino/>

[https://www.youtube.com/watch?v=71E\\_ZrMj4r8](https://www.youtube.com/watch?v=71E_ZrMj4r8)

<https://github.com>

[https://threads1818.rssing.com/chan-58411047/all\\_p245.html](https://threads1818.rssing.com/chan-58411047/all_p245.html)

<https://github.com/search?p=1&q=drv10983&type=Code>

<https://www.youtube.com/watch?v=vTlK0frF4FI>

<https://playground.arduino.cc/Main/I2CBidirectionalLevelShifter/>

<https://blogmasterwalkershop.com.br/arduino/como-usar-com-arduino-modulo-wifi-esp8266-esp-01/>

**Links para os Vídeos do Projeto:**

**<https://www.youtube.com/watch?v=YG71-UcJtrM&feature=youtu.be>**

**<https://www.youtube.com/watch?v=BSuVwxzROdI&feature=youtu.be>**

**[https://www.youtube.com/watch?v=jaP\\_2A6Okjk&feature=youtu.be](https://www.youtube.com/watch?v=jaP_2A6Okjk&feature=youtu.be)**