

UNIVERSIDADE PRESBITERIANA MACKENZIE
CURSO TECNOLOGO EM ANÁLISE E DESENVOLVIMENTO DE SISTEMAS

GABRIEL ALMEIDA DO CARMO

MEDIDOR DE VELOCIDADE

GABRIEL ALMEIDA DO CARMO

MEDIDOR DE VELOCIDADE

Trabalho para disciplina Objetos Inteligentes Conectados apresentado a Universidade Presbiteriana Mackenzie como requisição parcial para obtenção do diploma em análise e desenvolvimento de sistemas.

Orientador: Rodrigo Cardoso Silva

RESUMO

Este projeto tem por finalidade desenvolver um sistema inteligente para monitoramento de velocidade utilizando encoder (disco de giro). Para realizar esse projeto foi utilizado Wokwi para simulação online, Proteus 8 para desenvolvimento dos circuitos e Arduino IDE para programação.

Para esclarecer melhor sobre o funcionamento e aplicação do projeto, serão apresentados, no decorrer do trabalho, informações relevantes ao tema, como informações sobre os dispositivos utilizados, diagramas de ligação, circuitos elétricos.

A intenção de se desenvolver algo do tipo é pela necessidade de monitoramento de velocidade que se aplica em vários cenários distintos.

Palavras-chave: Encoder, Wokwi, Proteus 8, Arduino, IDE.

ABSTRACT

This project aims to develop an intelligent system for speed monitoring using encoder (turning disk). To carry out this project, Wokwi was used for online simulation, Proteus 8 for circuit development and Arduino IDE for programming.

In order to better clarify the operation and application of the project, relevant information will be presented during the work, such as information about the devices used, connection diagrams, electrical circuits.

The intention of developing something like this is due to the need for speed monitoring that applies in several different scenarios.

Keywords: Encoder, Wokwi, Proteus 8, Arduino, IDE.

Lista de Ilustrações

Figura 1 – Arduino Mega	8
Figura 2 - Modulo ESP8266	9
Figura 3 - Motor DC	9
Figura 4 - Modulo Enconder	10
Figura 5 - Disco Enconder	10
Figura 6 - Modulo Farol	11
Figura 7 - Modelo de montagem.....	12
Figura 8 – Projeto em Funcionamento.....	16
Figura 9 - MQTT Explore.....	17
Figura 10 - Console Wokwi.....	17

Sumário

1. Introdução	7
2. Matérias	8
3. Métodos	11
4. Funcionamento	12
5. Resultados	16
6. Conclusão	18
7. Referências	18

1. Introdução

Este trabalho consiste em desenvolver um sistema de monitoramento de velocidade baseado em Encoder que faça a aquisição e geração de dados digitais e/ou analógicos através de um hardware, comunicando com um microcontrolador Arduino.

Atualmente existem diversas soluções de hardware e software de aquisição de dados no mercado, porém em sua grande maioria com um custo bem elevado, o que dificulta a utilização deles em trabalhos acadêmicos ou até em ambientes de ensino.

Normalmente estes sistemas são utilizados para realizarem tratamento de sinais em diversas plantas industriais e também em pesquisas, onde são necessárias capturas de valores para análises posteriores.

Encoder é um transdutor, ou seja, é um sensor que converte um movimento mecânico angular ou linear em uma série de pulsos analógicos ou digitais elétricos, muito presente em aplicações industriais. Um encoder gera pulsos que são usados, por exemplo, para determinar velocidade, taxa de aceleração, distância, rotação, posição ou direção.

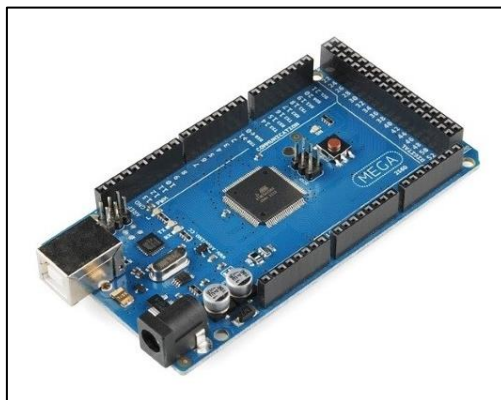
Arduino é uma plataforma de prototipagem eletrônica open-source que se baseia em hardware e software flexíveis e fáceis de usar. O Arduino pode sentir o estado do ambiente que o cerca por meio da recepção de sinais de sensores e pode interagir com os seus arredores, controlando luzes, motores e outros atuadores. O microcontrolador na placa é programado com a linguagem de programação Arduino, baseada na linguagem Wiring, e o ambiente de desenvolvimento Arduino, baseado no ambiente Processing.

2. Matérias

- *Arduino Mega*

Arduino é uma plataforma de prototipagem eletrônica de hardware livre e de placa única, projetada com um microcontrolador Atmel AVR com suporte de entrada/saída embutido, uma linguagem de programação padrão, a qual tem origem em Wiring, e é essencialmente C/C++. O objetivo do projeto é criar ferramentas que são acessíveis, com baixo custo, flexíveis e fáceis de se usar por principiantes e profissionais. Principalmente para aqueles que não teriam alcance aos controladores mais sofisticados e ferramentas mais complicadas.

Figura 1 – Arduino Mega



Fonte: <https://www.makerhero.com/produto/placa-mega-2560-r3-cabo-usb-para-arduino/>

- *Módulo ESP8266*

Módulo NodeMCU V3 - ESP8266 - CH340 ou Nodemcu Lolin é uma placa controladora capaz de colocar seu projeto conectado na internet, assim fazendo com que você seja capaz de acessá-lo, controlá-lo, adquirir informações de forma remota. Em outras palavras essa é a porta de acesso para a Internet das Coisas (Internet of Things - IoT).

Figura 2 - Modulo ESP8266



Fonte: <https://curtocircuito.com.br/nodemcu-v3-esp8266-esp-12e-ch340.html>

- *Motor Dc com caixa de redução*

O motor redutor de velocidade é um dispositivo mecânico cujo principal objetivo é diminuir a rotação de um acionador. Seus componentes básicos geralmente são rolamentos, carcaças, engrenagens e eixos de entrada e saída.

Quanto menor é a rotação do motor, quando comparada à rotação dos eixos, menor será a velocidade. A operação inversa (quando a rotação do motor é maior que a do eixo) causa um aumento na força. Resumindo, quando a rotação diminui, o torque aumenta.

Figura 3 - Motor DC



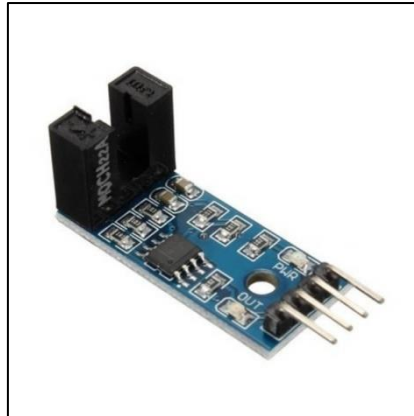
Fonte: <https://www.robocore.net/motor-motoredutor/motor-dc-3-6v-com-caixa-de-reducao-e-eixo-duplo>

- *Sensor Encoder*

Módulo Sensor de Velocidade Encoder é utilizado para realizar medições de rotação de motores, contagem de pulsos e como controlador de posicionamento. Pode ser utilizado com os mais diversos controladores e placas, como Arduino, Raspberry Pi e PIC. 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 - Modulo Encoder



Fonte: <https://www.institutodigital.com.br/produto/modulo-sensor-de-velocidade-encoder/>

- *Disco Encoder*

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, por exemplo.

Figura 5 - Disco Encoder



Fonte: <https://www.eletronicaomega.com/disco-encoder-para-sensor-de-velocidade>

- *Modulo Farol de led*

O Módulo LED 8mm Tipo Semáforo é composto por 3 LEDs, sendo um na cor verde, um na cor amarela e outro na cor vermelha. A disposição dos LEDs no módulo faz com que o mesmo fique parecido com um semáforo e isto possibilita projetos bem interessantes.

Figura 6 - Modulo Farol



Fonte: <https://www.eletrogate.com/modulo-led-8mm-tipo-semaforo>

3. Métodos

- IDE Ardino

Arduino Integrated Development Environment é uma aplicação de plataforma cruzada, escrito em funções de C e C ++. É usado para escrever e fazer upload de programas em placas compatíveis com Arduino.

- Protocolo MTQQ

o MQTT (*Message Queuing Telemetry Transport* e tendo tradução para português sob o nome de *Transporte de Filas de Mensagem de Telemetria*) é um protocolo de mensagens que foi criado com o objetivo de oferecer um baixo consumo de rede, banda e também dos demais recursos de software. O formato utilizado no MQTT é de Cliente/Servidor.

Por esse motivo e também por ter fundamentos na pilha TCP/IP ou em outros protocolos de rede, o MQTT tem extrema utilidade dentro da área de desenvolvimento de projetos de comunicação entre máquinas, também conhecido pelo termo M2M (Machine to Machine). Outra área também onde se torna muito presente é para conectividade de IoT (Internet of Things).

- Simulador de Circuitos – Wokwi

O Wokwi é uma ferramenta online de simulação de circuitos elétricos analógicos e digitais. Por ser gratuito e fácil de usar, encontramos nele uma oportunidade de ensino de Programação Embarcada, visto que a primeira barreira encontrada pelos alunos é a de não possuir os componentes e o microcontrolador em mãos.

- Proteus 8

Proteus Design Suite é um software para criação de projetos eletrônicos, composto por uma suíte de ferramentas, incluindo captura esquemática, simulação e módulos de projetos de placas de

circuito impresso (PCB, na sigla em inglês), usadas principalmente para o projeto de circuitos integrados. O software é usado principalmente por engenheiros de projeto eletrônico e técnicos para criar esquemáticos e impressões eletrônicas para a manufatura de PCBs.

4. Funcionamento

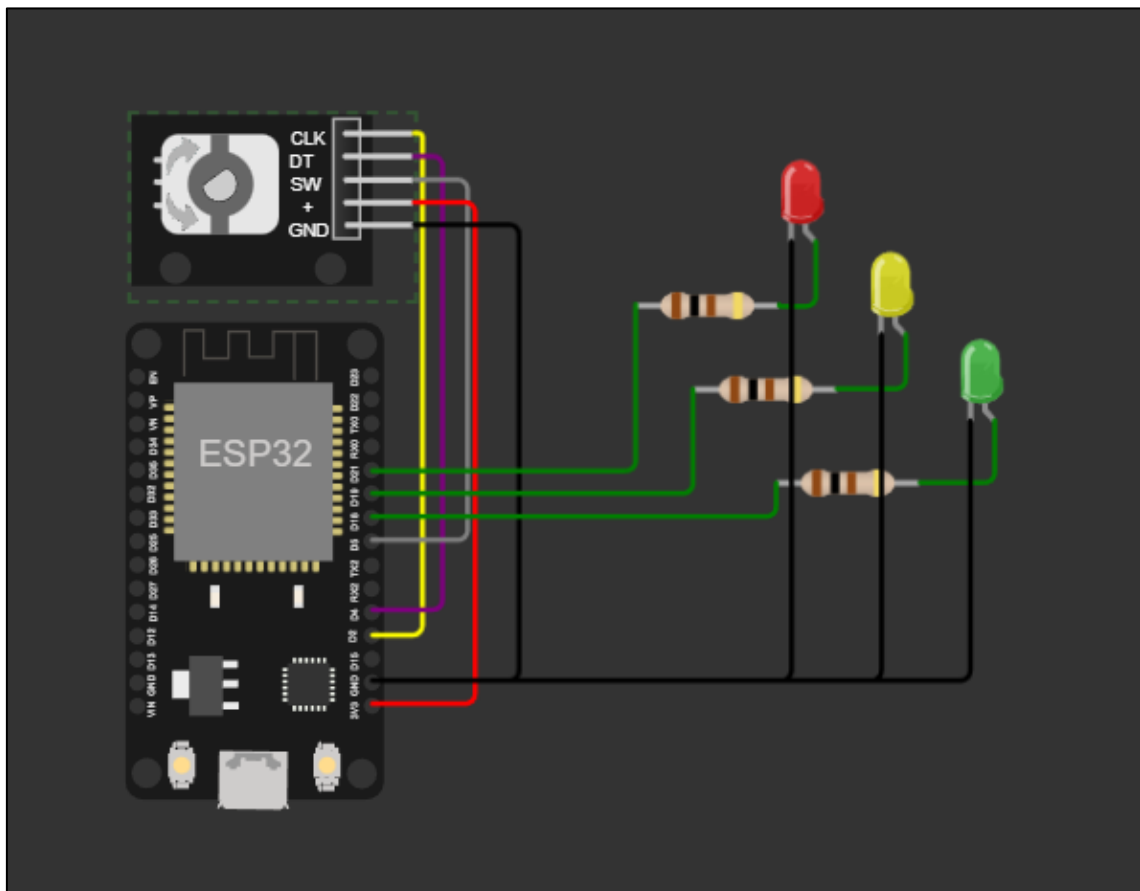
Esse projeto tem como objetivo o monitoramento de um motor DC, será medido a velocidade do motor em RPM (Rotação por minuto) e enviado uma mensagem do seu estado via internet pelo ESP 8266.

Utilizamos também 3 leds para simular um farol que será a indicação visual da velocidade do motor com os 3 estágios:

- Verde – Motor apresentar a velocidade dentro do esperado
- Amarelo – Motor com velocidade acima do esperado
- Vermelho – Motor com velocidade acima do limite esperado

Com a utilização do ESP 8266 será aplicado o protocolo MQTT para efetuar comunicação via internet para qualquer dispositivo conectado a ele com uma mensagem indicando o funcionamento em tempo real do motor.

Figura 7 - Modelo de montagem



Fonte: Do auto

- *Código Esp32/Arduino*

```

/ KY-040 ... ESP32
// CLK ... PIN 4
// DT ... PIN 2
// SW ... PIN 5
// + ... 3.3V
// GND ... GND

#include <Arduino.h>
#include "EspMQTTClient.h"

EspMQTTClient client(
  "Wokwi-GUEST",      // SSID WiFi
  "",                 // Password WiFi
  "test.mosquitto.org", // MQTT Broker
  "mqtt-wokwi",       // Client
  1883                 // MQTT port
);

long int rotValue=0, swValue=0;
uint8_t state=0;

#define ROTARY_PINA 2
#define ROTARY_PINB 4
#define ROTARY_PINSW 5
#define LED_VERMELHO 21
#define LED_AMARELO 19
#define LED_VERDE 18

portMUX_TYPE gpioMux = portMUX_INITIALIZER_UNLOCKED;

void IRAM_ATTR isrAB() {
  uint8_t s = state & 3;

  portENTER_CRITICAL_ISR(&gpioMux);
  if (digitalRead(ROTARY_PINA)) s |= 4;
  if (digitalRead(ROTARY_PINB)) s |= 8;
  switch (s) {
    case 0: case 5: case 10: case 15:
      break;
    case 1: case 7: case 8: case 14:
      rotValue++; break;
    case 2: case 4: case 11: case 13:
      rotValue--; break;
    case 3: case 12:
      rotValue += 2; break;
    default:

```

```

        rotValue -= 2; break;
    }
    state = (s >> 2);
    portEXIT_CRITICAL_ISR(&gpioMux);
}

void IRAM_ATTR isrSWall() {

    portENTER_CRITICAL_ISR(&gpioMux);
    swValue++;
    portEXIT_CRITICAL_ISR(&gpioMux);

}

void semaforo (int velocidade){
    if (velocidade < 50) {
        digitalWrite(LED_VERMELHO, LOW);
        digitalWrite(LED_AMARELO, LOW);
        digitalWrite(LED_VERDE, HIGH);
    }
    else if (velocidade >= 50 & velocidade < 100){
        digitalWrite(LED_VERDE, LOW);
        digitalWrite(LED_VERMELHO, LOW);
        digitalWrite(LED_AMARELO, HIGH);
    }
    else if (velocidade >= 100){
        digitalWrite(LED_VERDE, LOW);
        digitalWrite(LED_VERMELHO, HIGH);
        digitalWrite(LED_AMARELO, LOW);
    }
}

void lerEnviarDados() {
    Serial.println(rotValue);
    Serial.println(swValue);
    Serial.println("---");
    client.publish("topicowokwi/VelocidadeMackenzie", String(rotValue, 2));
}

void onConnectionEstablished()
{
    // Subscribe no "topicowokwi/msgRecebida/#" e mostra a mensagem
    recebida na Serial
    client.subscribe("topicowokwi/msgRecebidaMackenzie/#", [](const
String & topic, const String & payload) {
        Serial.println("Mensagem recebida no topic: " + topic + ", payload: "
+ payload);
    });
}

```

```

});
lerEnviarDados();
}

void setup(){
    pinMode(ROTARY_PINA, INPUT_PULLUP);
    pinMode(ROTARY_PINB, INPUT_PULLUP);
    pinMode(ROTARY_PINSW, INPUT_PULLUP);
    pinMode(LED_VERMELHO, OUTPUT);
    pinMode(LED_AMARELO, OUTPUT);
    pinMode(LED_VERDE, OUTPUT);

    attachInterrupt(ROTARY_PINA, isrAB, CHANGE);
    attachInterrupt(ROTARY_PINB, isrAB, CHANGE);
    attachInterrupt(ROTARY_PINSW, isrSWAll, CHANGE);
    Serial.begin(115200);

    client.enableDebuggingMessages(); // Enable debugging messages sent
to serial output
    client.enableHTTPWebUpdater(); // Enable the web updater. User and
password default to values of MQTTUsername and MQTTPassword. These can be
overridden with enableHTTPWebUpdater("user", "password").
    client.enableOTA(); // Enable OTA (Over The Air) updates. Password
defaults to MQTTPassword. Port is the default OTA port. Can be overridden
with enableOTA("password", port).
    client.enableLastWillMessage("TestClient/lastwill", "Vou ficar
offline");
}

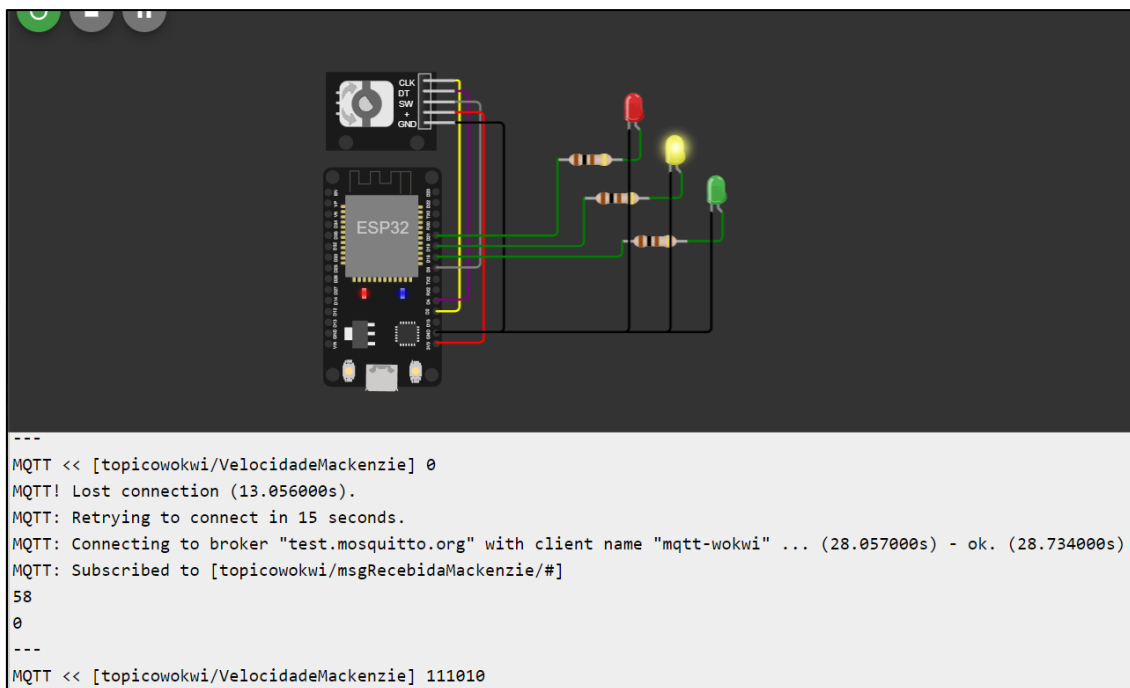
void loop()
{
    client.loop();
    semaforo(rotValue);
}

```

5. Resultados

- *Projeto*

Figura 8 – Projeto em Funcionamento



Fonte: Do autor

- *Vídeo*

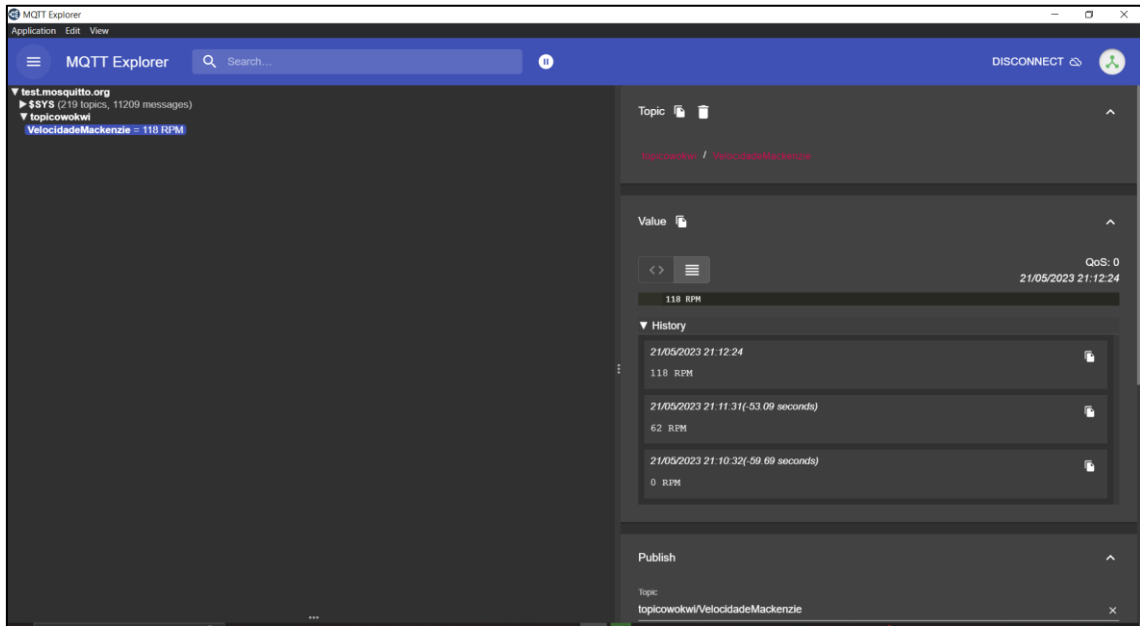
<https://youtu.be/Srin4JZIEAg>

- *Métricas*

Número Medida	Sensor	Tempo de resposta
0	0	23,35 Segundos
62	62	30,30 Segundos
118	188	45,38 Segundos

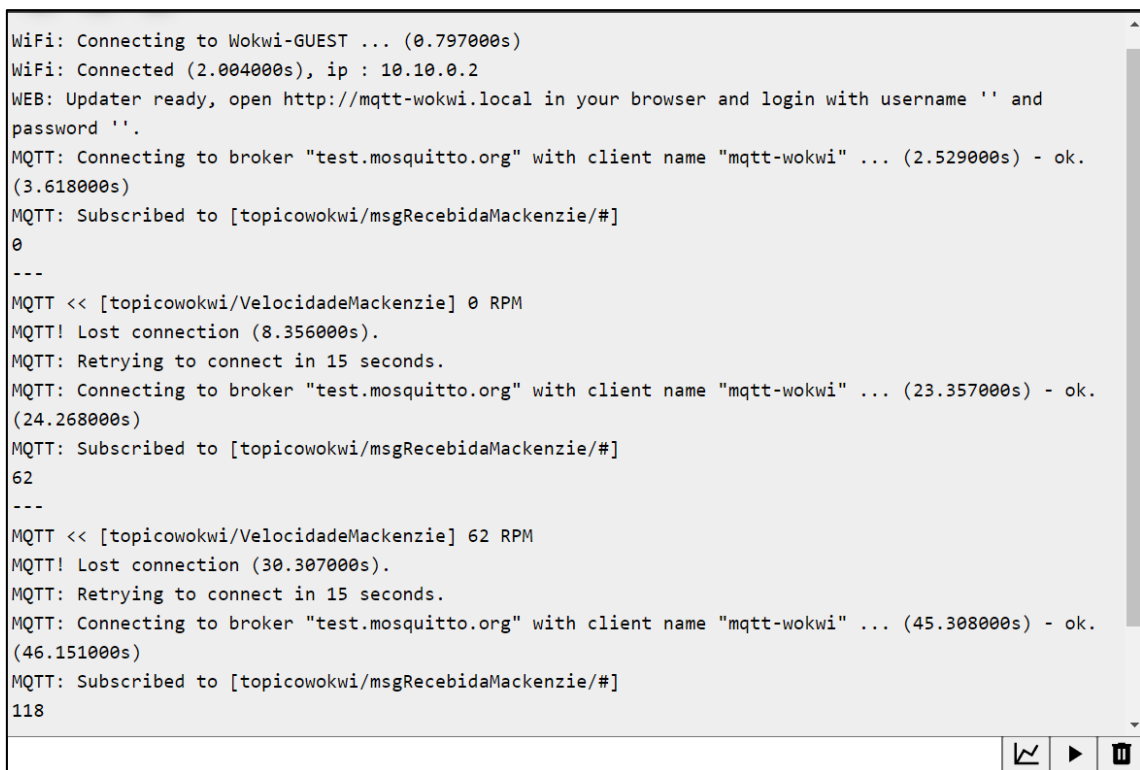
- *Telas MQTT*

Figura 9 - MQTT Explore



Fonte: Do autor

Figura 10 - Console Wokwi



Fonte: Do autor

- *GitHub*

<https://github.com/Gacarmo97/SensorVelocidade.git>

6. Conclusão

- Os objetivos propostos foram alcançados?

Sim, a única parte não alcançada foi a montagem física por dificuldade na montagem.

- Quais são os principais problemas enfrentados e como foram resolvidos?

Maior dificuldade foi a configuração do protocolo MQTT, com auxílio de pesquisas na internet consegui configurar.

- Quais são as vantagens e desvantagens do projeto?

A praticidade para montar por ter poucas peças e a facilidade para teste no Wokwi

O custo dos materiais para montar fisicamente.

- O que deveria/poderia ser feito para melhorar o projeto?

Utilizar sensores com maior precisão para melhores resultados.

7. Referências

COMO USAR COM ARDUINO – SENSOR DE VELOCIDADE (ENCODER). São Paulo: Novembro, 2019. Disponível em:

<<https://blogmasterwalkershop.com.br/arduino/como-usar-com-arduino-sensor-de-velocidade-encoder-chave-optica-5mm>>

O QUE É ENCODER? São Paulo: Disponível em:

<<https://www.dynaparencoders.com.br/blog/o-que-e-encoder/>>

ARDUINO: O QUE É? PRA QUE SERVE? QUAIS AS POSSIBILIDADES? São Paulo:

22 Setembro, 2016. Disponível em: <https://medium.com/nossa-coletividade/arduino-o-que-%C3%A9-pra-que-serve-quais-as-possibilidades-efbd59d33491>

MOTOR REDUTOR DE VELOCIDADE; São Paulo: Disponível em:
<https://www.renewreducores.com.br/motor-redutor-velocidade>