

CENTRO PAULA SOUZA FACULDADE DE TECNOLOGIA DE SANTO ANDRÉ
Tecnologia em Mecatrônica Industrial

ARIEL DANIEL RODRIGUEZ CHICO
GABRIEL SOUSA OLIVEIRA

ALIMENTADOR AUTOMATIZADO PARA ANIMAIS DOMÉSTICOS

SANTO ANDRÉ
2020

Ariel Daniel Rodriguez Chico
Gabriel Sousa Oliveira

ALIMENTADOR AUTOMATIZADO PARA ANIMAIS DOMÉSTICOS

Monografia apresentada ao Curso de
Tecnologia em Mecatrônica Industrial da
FATEC Santo André como requisito parcial
para obtenção do título de Tecnólogo em
Mecatrônica Industrial

Orientador: Professor Luiz Vasco Puglia

Santo André - SP

2020

FICHA CATALOGRÁFICA

C533a

Chico, Ariel Daniel Rodriguez
Alimentador automatizado para animais domésticos / Ariel Daniel Rodriguez Chico, Gabriel Sousa Oliveira. - Santo André, 2021. – 73f: il.

Trabalho de Conclusão de Curso – FATEC Santo André.
Curso de Tecnologia em Mecatrônica Industrial, 2021.

Orientador: Prof. Luiz Vasco Puglia

1. Mecatrônica. 2. Automação. 3. Alimentador. 4. Software. 5. Arduino. 6. Tecnologia. 7. Microcontroladores. 8. Animais domésticos. I. Oliveira, Gabriel Sousa. II. Alimentador automatizado para animais domésticos.

629.892

Ariel Daniel Rodriguez Chico
Gabriel Sousa Oliveira

ALIMENTADOR AUTOMATIZADO PARA ANIMAIS DOMÉSTICOS

Trabalho de Conclusão de Curso
apresentado a FATEC SANTO ANDRÉ
como requisito parcial à obtenção de título
de Tecnólogo em Mecatrônica Industrial.

Professor Orientador
Prof. Luiz Vasco Puglia

MEMBROS DA BANCA EXAMINADORA

Presidente da Banca
Prof. Luiz Vasco Puglia.
Fatec Santo André

Primeiro membro da Banca
Prof. Dr. Flávio Augusto Barrela.
Fatec Santo André

Segundo Membro da Banca
Profa. Suely Midori Aoki.
Fatec Santo André

Local: Fatec Santo André
Horário: 09:17
Data: 12/07/2021

SANTO ANDRÉ
2021

DEDICATÓRIA

Dedicamos este trabalho aos nossos familiares e a todas as pessoas que, de alguma maneira nos ajudaram no desenvolvimento deste projeto.

AGRADECIMENTOS

Agradecemos primeiramente a Deus por nos conceder a oportunidade de realizar o desenvolvimento deste projeto, pois sem ele nada é possível.

Agradeço a nossa família que nos incentivou em todas as nossas decisões, ajudando-nos e sempre estando ao nosso lado.

Agradecemos ao nosso orientador, Luiz Vasco Puglia, por estar ao nosso lado no desenvolvimento deste projeto, dedicando tempo hábil e nos repassando diversas informações.

Agradecemos também a todos que fazem parte da Fatec Santo André desde docentes até os alunos.

RESUMO

Com a crescente dinamização do trabalho, a vida das pessoas nas grandes cidades tem se tornado cada vez mais ocupada. Suas atividades cotidianas as obrigam a destinar menos tempo às suas relações sociais. Para suprir a necessidade de um companheiro, tanto os que moram sozinhos quanto as famílias acabam adotando um animal de estimação. Esses animais podem passar a viver durante o dia sem a presença de seu dono, e, embora isso possa causar estresse nos animais, o efeito primordial está em promover a alimentação a eles durante a ausência de seu dono. O projeto “Alimentador Automatizado para Animais Domésticos” tem como proposta desenvolver um protótipo para promover a alimentação diária e segura ao animal de estimação do usuário, mesmo durante a ausência do dono do animal. Em horário pré agendado, uma dose determinada de ração será disponibilizada ao animal. Além de garantir a alimentação diária, outras funções estão inclusas no projeto. Como, um relatório diário que indica o consumo da ração oferecida, medindo as sobras ou falta de ração necessária, ajustando, portanto, uma dieta balanceada por meio do acúmulo de dados. Um programa de celular permite, ainda, interagir com o sistema de forma remota, ou seja, mesmo distante de casa.

Palavras Chave: Arduino. Protocolo MQTT. Alimentador automático. Plataforma Blynk

ABSTRACT

With the growing dynamism of work, people's lives in cities have become increasingly busy. Their daily activities force them to devote less time to their social relationships. To overcome the need of a partner, those who live alone and families end up adopting a pet. These animals can live during the day without the presence of their owner, and, although this can cause stress in the animals, the primary effect is to provide food for them during the absence of their owner. The project "Automated Feeder for Domestic Animals" aims to develop a prototype to promote a daily and safe feeding to user's pet, even during the absence of the animal's owner. At a pre-scheduled time, a certain dose of feed will be made available to the animal. In addition to ensuring the daily feeding, other functions are included in the project. Like, a daily report that indicates the consumption of the offered ration, measuring the leftovers or lack of necessary ration, adjusting, therefore, a balanced diet through the accumulation of data. A smartphone software also allows you to interact with the system remotely, that is, even away from home.

Keywords: Arduino. MQTT Protocol. Automatic Feeder. Blynk Platform.

LISTA DE ILUSTRAÇÕES

Figura 1 NodeMcu	19
Figura 2 exemplo de protocolo MQTT	20
Figura 3 Arduino Mega.....	21
Figura 4 Exemplo de projeto no Blynk.....	22
Figura 5 Organização do Blynk: Blynk App, Server e Libraries	23
Figura 6 Célula de carga.....	24
Figura 7 Módulo HX711 SparkFun.....	25
Figura 8 Diagrama de célula de carga extensômetro	26
Figura 9 Motor de passo por dentro	28
Figura 10 disposição dos circuitos drivers no chip ULN2003APG	29
Figura 11 diagrama da placa de circuito do Módulo ULN2003APG.....	29
Figura 12 Tabela da carta de tempo modo full step do motor.....	31
Figura 13 Formato e Pinagem do ESP-01.....	31
Figura 14 Exemplo de tabela de comandos AT.....	32
Figura 15 ESP-01 Conectado ao Módulo UART	33
Figura 16 Módulo fonte 3.3-5V.....	34
Figura 17 Exemplo de rosca helicoidal.....	36
Figura 18 Exemplo de um Bocal com a Rosca Helicoidal instalada	36
Figura 19 Exemplo da visão completa do projeto.....	37
Figura 20 Exemplo de balança feita com a célula de carga.....	38
Figura 21 Esquema elétrico da balança	38
Figura 22 Código de calibração da balança	39
Figura 23 Exemplo de código final da balança.....	40
Figura 24 Circuito de atualização de firmware do esp-01 usando o arduino.....	41
Figura 25 Adaptrador USB para Esp com Botão Switch.....	41

Figura 26 Adaptador USB serial com o Botão DIP	42
Figura 27 Exemplo do Termite exibindo a versão do firmware	42
Figura 28 Tela principal e tela de log	43
Figura 29 Gerenciador de dispositivos	44
Figura 30 Exemplo da tela do ESP Flash Download Tool	45
Figura 31 Tela final do Esp Flash Download Tool	46
Figura 32 tela final do log no console	47
Figura 33 Firmware atualizado no Termite	47
Figura 34 Página do Blynk no Google Play	48
Figura 35 Tela de login do Blynk	49
Figura 36: Menu de criação de projetos Blynk	50
Figura 37 Tela de seleção de microcontrolador	51
Figura 38 Botão para adição de gadgets	52
Figura 39 Exemplo de evento	52
Figura 40 Tela final do aplicativo	53
Figura 41 Diagrama de Blocos	53

LISTA DE TABELAS

Tabela 1 Quantidade diaria de ração para cães.....	18
Tabela 2. Quantidade de ração diaria para gatos.....	18
Tabela 3 tabela de preços do projeto	35

LISTA DE ABREVIATURAS E SIGLAS

Abinpet Associação Brasileira Da Industria De Produtos Para Animais De Estimação

GPIO General Purpose Input/Output

MQTT Message Queue Telemetry Transport

IBGE Instituto Brasileiro de Geografia e Estatística

DIY Do It Yourself

SUMÁRIO

1	INTRODUÇÃO.....	12
1.1	OBJETIVO GERAL	14
1.2	OBJETIVOS ESPECÍFICOS	14
1.3	PROBLEMA	14
1.4	JUSTIFICATIVA	14
2	FUNDAMENTAÇÃO TEÓRICA.....	16
2.1	ALIMENTADORES AUTOMÁTICOS	16
2.2	ALIMENTAÇÃO PET	17
2.3	MICROCONTROLADORES	19
2.4	PROTOCOLO MQTT	20
2.5	ARDUINO.....	21
2.6	BLYNK	22
2.6.1	<i>Blynk app</i>	23
2.6.2	<i>Blynk server</i>	23
2.6.3	<i>Blynk libraries</i>	23
2.7	CÉLULA DE CARGA STRAIN GAUGE E MÓDULO HX711	24
2.8	MOTOR DE PASSO 28BYJ-48 E DRIVER ULN2003	27
2.9	MÓDULO WIFI ESP-01	31
2.10	ADAPTADOR UART PARA MÓDULO ESP-01	33
2.11	MÓDULO FONTE MULTI-SAÍDAS 3.3V-5V	34
3	DESENVOLVIMENTO.....	35
3.1	METODOLOGIA A MATERIAIS UTILIZADOS	35
3.2	PROJETO	35
3.3	TESTES REALIZADOS	38
3.4	RESULTADOS OBTIDOS	53
4	CONSIDERAÇÕES FINAIS.....	54
4.1	CONCLUSÃO	54
4.2	TRABALHOS FUTUROS	54

REFERÊNCIAS BIBLIOGRÁFICAS	56
APÊNDICES	64
APÊNDICE A – SIMULAÇÃO DO CIRCUITO NO PROGRAMA FRITZING.....	64
APÊNDICE B – ESQUEMA ELÉTRICO DO CIRCUITO	65
APÊNDICE C – PROGRAMAÇÃO EM ARDUINO DO PROJETO	66

1 INTRODUÇÃO

Desde que os animais começaram a ser domesticados sua interação com o ser humano foi mudando e, naturalmente, a relação da família com seu animal também tem mudado. Atualmente, eles passaram a ser considerados pelas pessoas como um integrante da família, buscando, assim, no animal uma companhia, um afeto, uma forma de segurança e de diversão. No Brasil, segundo o Instituto Brasileiro de Geografia e Estatística (IBGE, 2019) “46,1% dos domicílios tinham pelo menos um cachorro em 2019. Já os gatos eram parte de 19,3% dos lares brasileiros. no total são 47,9 milhões de domicílios com ao menos um cão ou gato”.

“O mercado *pet* no Brasil já é o segundo maior do mundo, com 6,4% de participação global superando o reino unido (6,1%) e ficando só atrás dos Estados Unidos, que tem a maior participação com 50% do mercado.” (Forbes, 2020). O resultado disso é que nunca o setor viveu tamanha concorrência, o que obriga as clínicas veterinárias, *pet shops* e prestadores de serviço a inovar para aparecer e crescer.

Levando em consideração que no mundo de hoje as pessoas estão mais comprometidas com o trabalho e viagens, além do ritmo cada vez mais puxado do dia-a-dia e outros contratempos, os quais, às vezes, obrigam os donos a passarem algum tempo longe dos seus animais de estimação. Com o avanço da tecnologia, a qual permitiu automatizar processos que demoravam mais tempo, usando praticas como IoT (Internet of things), a ideia do alimentador automatizado surge para suprir a obrigatoriedade da presença de uma pessoa para alimentar o animal, com o intuito de cuidar da alimentação do animal de forma automática na ausência do dono, proporcionando, assim, uma maior praticidade no cotidiano.

O alimentador fará o papel de alimentar o animal, o prototótipo consiste em despejar a ração no recipiente e deixa-la armazenada até que o comando de alimentação seja dado, uma vez ativado, o sistema irá ligar o motor que por sua vez irá girar o fuso (rosca sem fim) acoplado a ele, com o giro do fuso, a ração será tracionada até o pote do animal, e só desativará

quando o peso determinado for atingido, todo o processo será automatizado através de um aplicativo de celular que também contará com a função de acionamento manual ao pressionar de um botão dentro do aplicativo.

1.1 Objetivo geral

O objetivo principal é criar um protótipo, usando o microcontrolador arduino, de um alimentador que fará o reabastecimento da ração do animal, o protótipo poderá ser controlado a distancia com a utilização da plataforma blynk que irá usufruir de uma série de comandos pré-programados disponíveis em um código feito na IDE do próprio microcontrolador, fazendo que com mesmo a distância a alimentação do animal seja controlada e supervisionada.

1.2 Objetivos específicos

- Aplicar conhecimento em eletrônica desenvolvendo um sistema microcontrolado.
- Aplicar conhecimentos mecânicos na elaboração de uma estrutura estável e resistente.
- Implementar um aplicativo de celular para o controle a distância do dispositivo.

1.3 Problema

Não é recomendado que um animal doméstico tenha acesso à uma grande quantidade de alimento em um espaço muito curto de tempo. Não sendo possível realizar esse controle sem um sistema automatizado, foi idealizado então um protótipo que visa sanar o problema em fornecer toda a ração em uma única vez pois isso dá liberdade ao animal em decidir o quanto e quando comer. A ausência de seu dono para regularizar isso pode, também, acabar dificultando a alimentação correta e nutrição do animal, podendo acarretar problemas futuros.

1.4 Justificativa

Os conhecimentos adquiridos sobre como utilizar um microcontrolador arduino e uma interface do aplicativo blink também permitiram o desenvolvimento do trabalho, O trabalho "Alimentador automatizado para animais domésticos" tem como idéia, usar os conceitos

de IoT (Internet of Things) e como eles podem ser aplicados para automatizar tarefas básicas como por exemplo alimentar um animal.

2 FUNDAMENTAÇÃO TEÓRICA

Nesse capítulo serão descritos alguns conceitos relacionados ao presente trabalho. Os tópicos abordados nele serão: Alimentadores automáticos e seus tipos, Alimentação pet, Microcontroladores, protocolo MQTT, Arduino, Blynk, Célula de carga strain gauge e o Módulo HX711, Motor de passo 28BYJ-48 e o Driver ULN2003, Módulo Esp-01, Adaptador UART do módulo Esp-01 e por fim o Módulo fonte de multisaídas.

2.1 Alimentadores automáticos

De acordo com Andreia Sales (2019) Presidente de uma ONG de resgate a animais a 6 anos e escritora de artigos para o site Reviewbox, alimentadores automáticos são, em resumo, dispositivos utilizados para tornar a tarefa de alimentação dos animais de estimação mais simples de forma a não ocupar tempo no dia a dia. Atualmente podem ser encontrados modelos com as mais variadas funções, que permitem o controle do horário e da quantidade de alimento. No total são 3 as categorias que os diferenciam, sendo elas:

- **Alimentador automático comum:** Modelo encontrado no mercado com maior facilidade, e por ser mais simples em comparação aos outros, geralmente é feito em materiais duráveis e fáceis de serem higienizados, como acrílico, plástico ou polipropileno.
- **Alimentador automático digital:** Uma opção mais moderna que permite programar a quantidade exata de ração e a frequência das refeições. Os modelos podem oferecer outras funções, como relógio e gravador de voz por um custo mais elevado.
- **Alimentador automático controlado por aplicativo:** Uma opção para os donos de *pet*, cujo o tempo em sua residência é limitado e desejam não só alimentar seu animal como poder controlar tal tarefa, recebendo informações sobre o mesmo à distância.

Os modelos mais avançados possuem até mesmo uma câmera para monitoramento, e o usuário pode controlar a frequência e a quantidade de cada refeição mesmo se estiver viajando ou no trabalho.

2.2 Alimentação pet

Assim como com os humanos a alimentação é algo crucial para a saúde de um animal, e, assim como os pais para com as crianças, o dono de *pet* deve desde sempre ficar atento com a dieta dele para não gerar problemas futuros.

Conforme José Edson Galvão de França, (s.d) Presidente da Abinpet:

“Uma alimentação correta garante vitalidade e energia, além de um sistema imunológico mais bem preparado e ossos e dentes fortes. Muitos donos optam por restos de comida, o que diminui os gastos. Mas a alimentação humana, com seus temperos, condimentos e outros ingredientes, não é indicada para os animais de estimação.

Por serem formulados especificamente para cada espécie, os alimentos industrializados disponíveis em supermercados, pet shops e afins garantem o desenvolvimento completo de filhotes, longevidade e uma vida saudável e ativa para animais adultos, sejam eles cães, gatos, aves, peixes ou pequenos roedores. Esses insumos contêm ingredientes específicos em sua formulação que combatem doenças, além de estarem prontos para consumo. Alimentos secos têm cerca de um ano de validade, desde que bem armazenados, garantindo também segurança.

O caso dos cães é um bom exemplo de como a alimentação desequilibrada pode prejudicar o animal. O excesso de vitamina D pode deformar dentes e mandíbulas e levar ao enrijecimento de órgãos como pulmão, rins e estômago. Além de alterações físicas, males comportamentais também podem surgir de uma nutrição falha. A falta de sais minerais, como iodo, bromo e manganês, pode fazer do animal irritadiço, ao passo que a alimentação com doses altas de nutrientes pode provocar malformação nos cães mais jovens”. (Abinpet, s.d.)

As tabelas 1 e 2 indicam a quantidade de ração, diária, necessária para cães e gatos conforme o peso.

Tabela 1 Quantidade diaria de ração para cães

Porte do cachorro	Quantidade diária de ração
Porte miniatura (até 5 kg)	6% do peso do cachorro
Porte pequeno (de 5 a 10 kg)	5% do peso do cachorro
Porte médio (de 10 a 22 kg)	5% do peso do cachorro
Porte grande (de 22 a 40 kg)	4,5% do peso do cachorro
Porte gigante (mais de 40 kg)	4% do peso do cachorro

Fonte: <https://www.reviewbox.com.br/alimentador-automatico>

Tabela 2. Quantidade de ração diaria para gatos

Porte do gato	Quantidade diária de ração
Até 2 kg	25-40 g
2-3 kg	35-50 g
4-5 kg	40-60 g
6 kg	55-85 g
7 kg	60-90 g
8 kg	70-100 g
9 kg	75-110 g
10 kg	80-120g

Fonte: <https://www.reviewbox.com.br/alimentador-automatico/>

Usar esse tipo de referência no que diz respeito a alimentação é, uma das maneiras que alguns donos usam para tentar regravar a dieta de seu pet e evitar problemas futuros ou até prolongar a vida dele e que várias empresas que criam alimentadores automáticos podem usar na hora de, na descrição de seu produto, indicar o tipo de usuário mais indicado.

2.3 Microcontroladores

No Microcontrolador, os elementos internos e os periféricos externos são programados utilizando registradores. Microcontroladores em geral são um conjunto de periféricos que, conhecendo seu conjunto de registradores, podem ser programados (informação verbal).¹

As informações sobre a forma como cada registrador deve ser programado, variam de acordo com o modelo e o fabricante utilizado

Figura 1 NodeMcu



Fonte: https://i0.wp.com/porta.vidadesilicio.com.br/wp-content/uploads/2017/06/20170610_160626.jpg?resize=768%2C576&ssl=1

Um exemplo de um microcontrolador (Figura 1) é o NodeMCU (Esp-12) que é basicamente um SoC (System on a chip), ele possui uma

¹ Informação referente ao slide da disciplina de microcontroladores e microprocessadores, ministrada pelo professor Murilo Zanini De Carvalho; FATEC Santo André, em set. 2019.

alimentação de 3.3v e pode ser programado em python através de sua própria IDE, ou através da IDE do Arduino

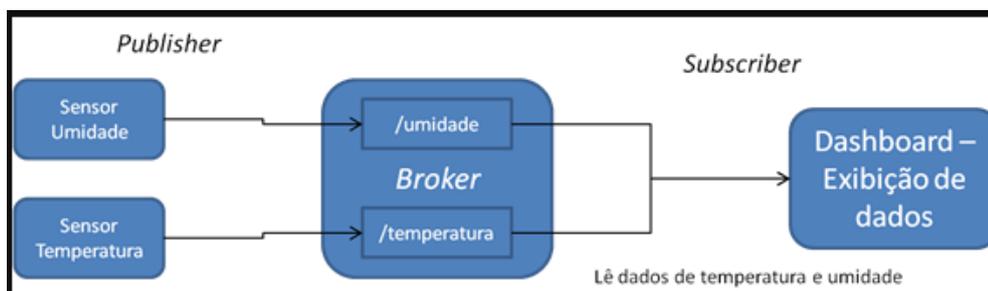
2.4 Protocolo MQTT

MQTT é uma camada de aplicação cujo objetivo é ter baixo consumo de banda de rede e de recursos de hardware, características que o fazem ser um dos protocolos mais utilizados para projetos IoT.

“Desenvolvido pela IBM no final dos anos 1990 com base em TCP/IP, o MQTT (*Message Queue Telemetry Transport*). Como seu nome sugere, ele é um protocolo de mensagens com suporte para a comunicação assíncrona entre as partes, ou seja, desacopla o emissor e o receptor da mensagem tanto no espaço quanto no tempo e, portanto, é escalável em ambientes de rede que não são confiáveis. Surgido como uma alternativa para vincular sensores em pipelines de petróleo a satélites, se tornou oficialmente um padrão aberto OASIS em 2014, quando passou a contar com suporte às linguagens de programação mais populares”.(Altus. 2018)

O Protocolo MQTT (Figura 2) possui como princípio de funcionamento um servidor, chamado de Broker, a função desse broker é direcionar as informações aos locais certos. Por exemplo quando um sensor transmite um dado ele é chamado de publisher, o broker guarda a informação do publisher e as redireciona para o subscriber, que é o dispositivo no qual será utilizada tal informação.

Figura 2 exemplo de protocolo MQTT



Fonte: https://miro.medium.com/max/567/1*ibfX1UaLp0mfetaxgqWj8Q.png

“Por ter uma estrutura leve e flexível, o MQTT se tornou um dos melhores protocolos para desenvolvimento de aplicações com alto grau de conexão. Enquanto sua leveza permite que seja utilizado em hardwares de dispositivos altamente restringidos ou em redes com largura de banda limitada e alta latência, a flexibilidade encontrada no padrão possibilita suporte a diversos cenários de aplicativos para dispositivos e serviços de IoT. Estas características diferenciam o MQTT dos demais protocolos Ethernet que, por

serem baseados em HTTP, apresentam importantes problemas de velocidade, confiabilidade e desempenho”. (Altus, 2018.)

2.5 Arduino

Criado em 2005 por um grupo de 5 pesquisadores, Massimo Banzi, David Cuartielles, Tom Igoe, Gianluca Martino e David Mellis, Arduino é uma plataforma muito utilizada no ramo de automação, desde iniciantes até os mais experientes entusiasmados da eletrônica e seu uso é cada vez mais explorado até em cursos de graduação. Projetada para ser um dispositivo de fácil programação e com um preço acessível e que pudesse ser utilizada várias formas (Minakawa et al, 2012).

A placa pode ser hibridamente programada em C e C++, sua conexão é feita via uma IDE (Integrated Development Environment, ou Ambiente de Desenvolvimento Integrado) que é disponibilizada no site oficial da *arduino* e ou pelo celular através de aplicativos que simulam esta mesma IDE, permitindo praticidade e portabilidade de projetos dos mais simples aos mais complexos.

Dentre as inúmeras versões do arduino já lançadas oficialmente, a utilizada para o projeto em questão, é a plataforma MEGA (Figura 3) cujo número de entradas e saídas é proporcional ao necessário para o projeto.

Figura 3 Arduino Mega



Fonte: <http://www.robotpark.com/image/cache/data/PRO/91068/Arduino-Mega-2560-R3-Pic01-700x700.png>

2.6 Blynk

A plataforma Blynk (Figura 4) é um aplicativo personalizável cujo o intuito é dar ao usuário a oportunidade de controlar remotamente uma quantidade considerável de hardwares (a maioria baseada em arduino), e ainda receber e enviar dados aos mesmos ao passo que é possível construir de forma rápida, uma interface gráfica para diversas situações.

Figura 4 Exemplo de projeto no Blynk

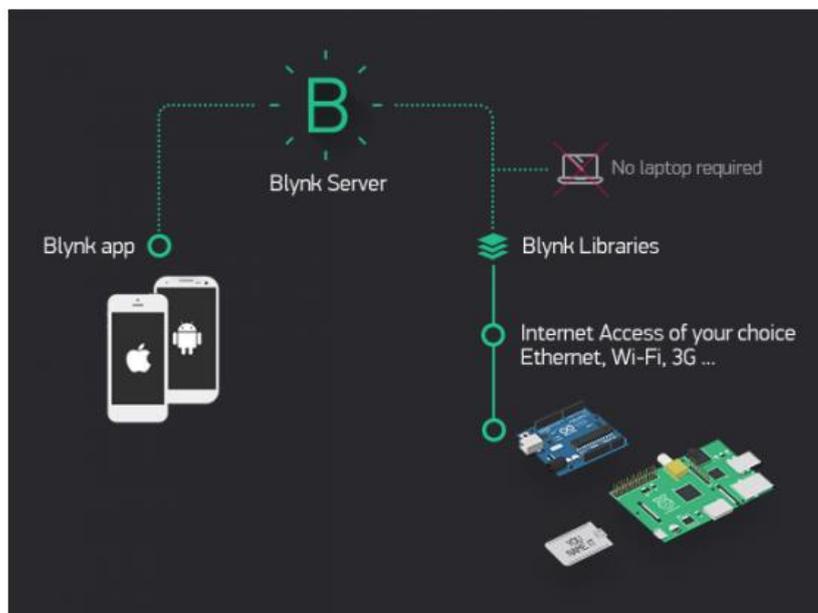


Fonte: <https://www.embarcados.com.br/wp-content/uploads/2018/05/image1-235x418.png>

Na figura 5 temos um exemplo de como seria a ideia do aplicativo blynk, em diagrama de blocos, considerando que ele é geralmente utilizado para substituir a função do uso contínuo de um computador em relação ao envio de comandos para a plataforma, uma vez que, depois de ser efetuada a programação, tudo que precisará ser utilizado para realizar ações com o projeto é um celular.

A plataforma se resume em 3 partes (figura 5): Blynk App, Blynk Server e o Blynk Libraries

Figura 5 Organização do Blynk: Blynk App, Server e Libraries



Fonte : <https://www.embarcados.com.br/wp-content/uploads/2018/05/image3.png>

2.6.1 Blynk app

De maneira resumida, sobre o aplicativo blynk, tem-se:

“O App Blynk é um aplicativo disponível para Android e iOS que permite ao usuário criar aplicações que interagem com o hardware. Através de um espaço próprio para cada projeto, o usuário pode inserir Widgets que implementam funções de controle (como botões, sliders e chaves), notificação e leitura de dados do hardware.”(Serrano, 2018).

2.6.2 Blynk server

Sobre o Blynk server,

“A Aplicação da comunicação entre o aplicativo e o hardware do usuario é responsabilidade da Cloud Blynk que é o servidor responsável por transmitir os dados do hardware, além de armazenar os dados dos sensores lidos pelo hardware o ainda que o aplicativo esteja fechado.” Vale ressaltar que os dados armazenados no server Blynk podem ser acessados externamente através de uma API HTTP, o que abre a possibilidade de utilizar o Blynk para armazenar dados gerados periodicamente como dados de sensores de temperatura, por exemplo”.(Serrano, 2018).

2.6.3 Blynk libraries

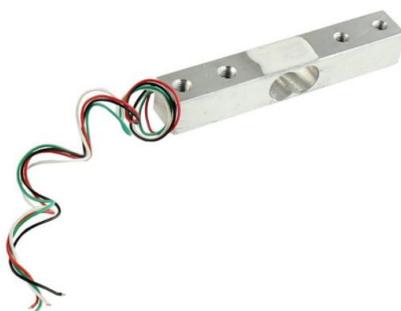
Sobre as Libraries do Blynk, é dito que:

“(…) do lado do hardware temos as bibliotecas Blynk para diversas plataformas de desenvolvimento. Essa biblioteca é responsável por gerir toda a conexão do hardware com o servidor Blynk e gerir as requisições de entrada e saída de dados e comandos. A forma mais fácil e rápida é utilizá-la como bibliotecas Arduino, no entanto, é possível obter versões da biblioteca para Linux (e Raspberry Pi!), Python, Lua, entre outras.” (Serrano, 2018).

2.7 Célula de carga strain gauge e módulo hx711

Uma célula de carga (Figura 6) é composta por uma ponte resistiva que varia em função do peso da carga aplicada. Quando a célula de carga entra em operação, ou seja, é aplicado determinado peso, ele envia uma tensão ao microcontrolador, Que vai efetuar as leituras de acordo com o peso da carga sobre a célula. Como o sinal (tensão) enviado pela célula é baixo, é necessário o uso do amplificador e conversor de sinal, de forma que a porta analógica da plataforma microcontrolada possa realizar a leitura (Santana, 2019).

Figura 6 Célula de carga

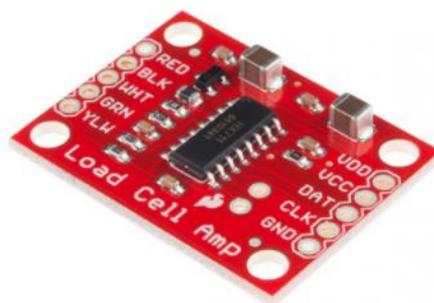


Fonte: <https://www.usinainfo.com.br/blog/wp-content/uploads/2019/09/celula-de-carga-arduino-768x580.jpg>

Segundo Straub (2019), trabalhando como interface entre a célula de carga e o Arduino, o Módulo Conversor permite fazer uma leitura de peso facilmente, convertendo também leituras analógicas em digitais com sua função de conversão analógico/digital de 24 bits. Abaixo podemos observar sua estrutura física, lembrando eu os pinos precisam ser soldados para sua utilização.

A figura 7 mostra o modelo da SparFun do módulo HX711 usado em conjunto com a célula de carga.

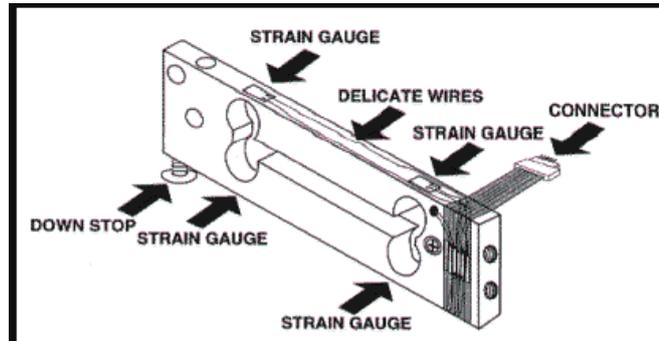
Figura 7 Módulo HX711 SparkFun



Fonte: https://www.sparkfun.com/products/13879?_ga=2.9666684.834950726.1615557496-738860267.1614338428

A célula de carga utilizada no projeto é do tipo extensômetro de barra, como mostra o diagrama na figura 8. nesse tipo a célula é configurada em formações "Z" para que o torque seja aplicado à barra e os quatro medidores de deformação na célula medirão a distorção de flexão, dois medindo compressão e dois tensão. Quando posicionados no formato de uma ponte de wheatstone, esses medidores podem facilmente medir com precisão as mudanças nos medidores de tensão por menores que sejam.

Figura 8 Diagrama de célula de carga extensômetro



Fonte: https://cdn.sparkfun.com/assets/learn_tutorials/3/8/2/loadcell.gif

2.8 Motor de passo 28BYJ-48 e driver uln2003

Um motor de passo é um dispositivo eletromecânico que converte os impulsos elétricos em movimentos. O eixo do mesmo gira conforme são energizadas as bobinas que ele possui.

A sequência dos impulsos aplicados está diretamente relacionada com a direção de rotação do eixo do motor, a velocidade de rotação do motor com a frequência dos impulsos de entrada e o comprimento de rotação com o número de impulsos de entrada. Ou seja, para controlar o motor de passo precisa-se aplicar tensão a cada uma das bobinas em uma sequência específica (Brites, 2018).

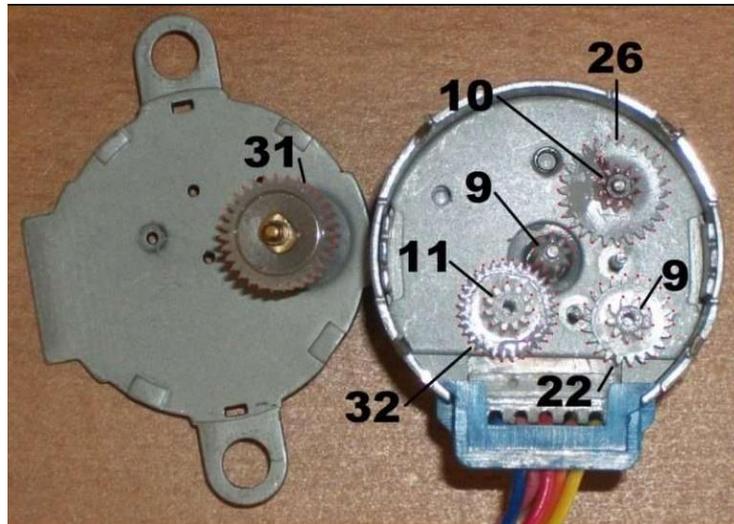
Uma das vantagens mais significativas de um mecanismo como esse é a sua capacidade de ser controlado com precisão num sistema de circuito aberto. Controle de circuito aberto significa que nenhuma informação de feedback sobre a posição é necessária. Este tipo de controle elimina a necessidade de caros sistemas de detecção. Sua posição é conhecida simplesmente através do controle dos impulsos de entrada (Brites, 2018).

As especificações do Motor são:

- Tensão de operação: 5V CC
- Número de fases: 4
- Razão da variação de velocidade: 1/64 (mecanismo de redução)
- Ângulo do passo: 5,625 graus => 64 passos/volta (360/64=5,625)
- Resistência CC: 50 ohms
- Frequência: 100 Hz
- Torque de tração: > 34,3 mN.m

A imagem a seguir (Figura 9) mostra como é o motor de passo aberto e o número de engrenagens que ele possui.

Figura 9 Motor de passo por dentro



Fonte: <https://blog.eletragate.com/guia-completo-do-motor-de-passo-28byj-48-driver-uln2003/>

Foi feito em um tópico no fórum oficial da arduino (<https://forum.arduino.cc/t/geared-stepper-motor/71308>) uma postagem com o cálculo para saber qual seria o valor da redução, como mostrado na figura 11, quando aberto pode-se observar que a caixa de redução tem várias engrenagens (31 dentes, 32 dentes, 26 dentes, 22 dentes, 11 dentes, 10 dentes, e mais duas com 9 dentes).

Sendo assim, a equação para determinar o valor da redução é:

$$\frac{31 \times 32 \times 26 \times 22}{11 \times 10 \times 9 \times 9} = \frac{283712}{4455} = \frac{25792}{405} = 63,68395$$

Na especificação do fabricante, o valor está arredondado para 64, e, para se calcular o número de passos do motor interno dá para girar uma volta do eixo externo, considerando a redução, temos:

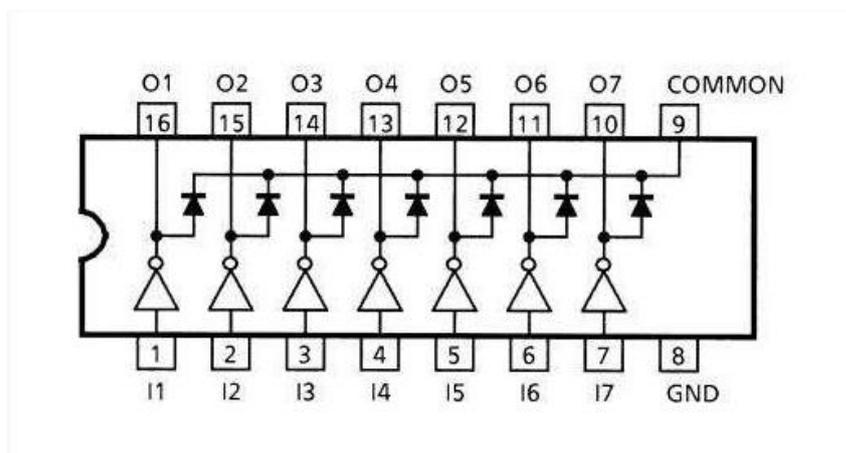
$$\frac{(64 \times 25792)}{405} = 4075,7728395$$

o que significa que para uma volta inteira do eixo externo são necessários 4075 passos.

Sobre o módulo que acompanha o motor, ele possui um chip (ULN2003) que possui 7 drivers de transistores darlington, usados para o acionamento das cargas indutivas (Figura 10).

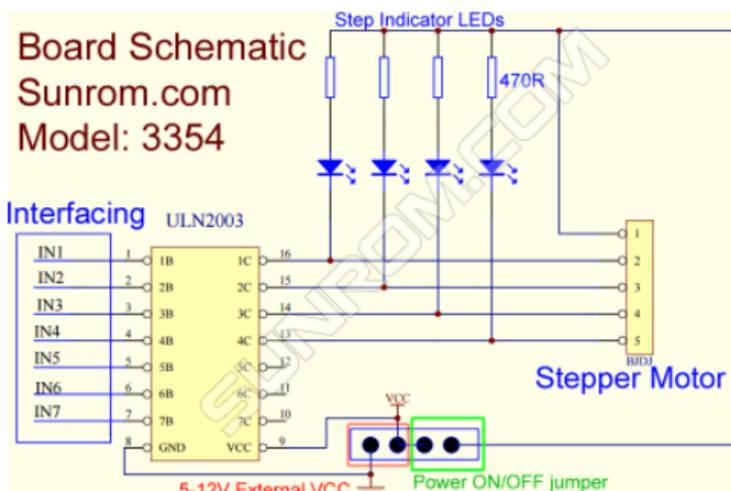
“Todas as saídas têm o coletor aberto e diodos de supressão (Clamp). Os transistores suportam tensões de até 50V e correntes de até 500 mA. Todas as entradas IN1, IN2, IN3 e IN4 são compatíveis com sinais TTL e CMOS, com limite de 5V. O pino comum tem que ser conectado na tensão de alimentação do motor. Nesse caso é conectado no 5V”. (Eletrogate,2018).

Figura 10 disposição dos circuitos drivers no chip ULN2003APG



Fonte: <https://blog.eletrogate.com/wp-content/uploads/2018/07/ULN2003pinout.jpg>

Figura 11 diagrama da placa de circuito do Módulo ULN2003APG



Fonte: https://blog.eletrogate.com/wp-content/uploads/2018/07/Stepper_Motor_28BYJ48_ULN2.gif

A imagem anterior (Figura 11) é uma representação do diagrama do módulo ULN2003.

Os quatro leds vermelhos (A,B,C e D) são usados para indicar o acionamento de cada um dos drivers (fases do motor). Mesmo que o diagrama mostre sete entradas, somente quatro podem ser usadas e somente essas tem pinos no conector.

Uma outra observação a se considerar no uso do motor de passo com esse driver, é a necessidade de se usar uma fonte externa para fornecer os 5V necessários para o seu funcionamento, uma vez que, caso venha a se tentar os usar os 5V fornecidos pelo arduino, isso poderia gerar uma sobrecarga sobre o mesmo.

o Motor de passo é unipolar, isto é, ele possui quatro enrolamentos que são denominados como Fases.

Cada circuito driver do chip ULN2003A aciona uma das fases. E a ativação de cada driver é realizada pelas portas digitais do Arduino (portas D08, D09, D10 e D11 por exemplo).

Para um motor de Passo Unipolar, existem quatro modos de operação. O modo "Passo completo" com alto torque (Full step), o modo "Passo completo" com baixo torque (Wave step), o modo "Meio Passo" (half step) e Micro-passo (Micro stepping). Dentre eles o mais adequado a se utilizar neste projeto é o full step, pois ao se manipular carga pelo princípio da tração é necessário levar em conta o toque necessário para poder mover a mesma.

Para entender como as Fases são acionadas em cada caso, são usadas cartas de tempo, onde o valor "0" representa enrolamento desativado e "1" representa enrolamento ativo.

No caso do modo full step (Figura 12), duas Fases são acionadas ao mesmo tempo.

Figura 12 Tabela da carta de tempo modo full step do motor

Step	$\phi 4$	$\phi 3$	$\phi 2$	$\phi 1$
0	0	0	1	1
1	0	1	1	0
2	1	1	0	0
3	1	0	0	1

Fonte: <https://blog.eletrogate.com/wp-content/uploads/2018/07/Fullstep.jpg>

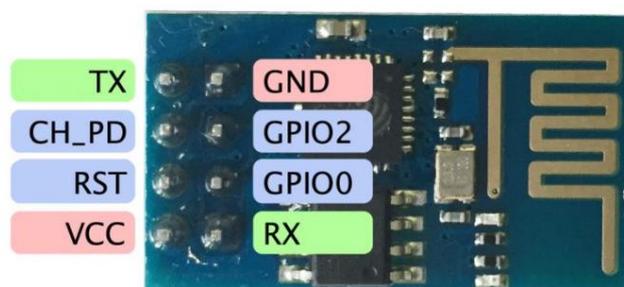
Esse modo é o mais usado, pois é, dentre todos, o que tem mais torque no entanto consumo de corrente é mais alto, devido à ativação das duas fases.

2.9 Módulo WIFI esp-01

Para Segundo e Morais (2017):

“O módulo Wi-Fi ESP8266 ESP-01 foi especialmente desenvolvido para que se possa conectar o microcontrolador a uma conexão WiFi de forma fácil e eficaz. O módulo WiFi ESP8266 suporta as redes 802.11 b/g/n, muito usadas atualmente, podendo trabalhar como um Ponto de Acesso (Acess Point) ou como uma Estação (Station), enviando e recebendo dados. A comunicação do módulo com o Arduino pode ser feita via serial utilizando os pinos RX e TX, podendo ser configurada através de comandos AT. Porém, o Adaptador WiFi ESP-01 já possui o conversor de nível e pode ser conectado diretamente a protoboard. Dessa forma, é possível enviar dados de um componente para o outro através da rede Wi-Fi”.

Figura 13 Formato e Pinagem do ESP-01



Fonte: <https://www.embarcados.com.br/wp-content/uploads/2016/11/Pinos-ESP-01-696x330.jpg>

Existem dois principais modos de operação do ESP-01.

Modo 1 – Comandos AT

“No modo Comandos AT, o módulo possui instalado um programa chamado ‘Firmware Comandos AT’. Esse programa tem uma função pré-determinada que é receber comandos de operação via UART(pinos TX e RX). Nesse modo, o ESP-01 deverá trabalhar em conjunto com um microcontrolador. O microcontrolador será o responsável por enviar os comandos AT para o ESP-01. A placa Arduino possui um microcontrolador capaz de realizar esses comandos. O Firmware Comandos AT já vem instalado de fábrica podendo estar em diferentes versões, mas pode ser atualizado caso venha com uma versão antiga”. (Bauermeister, 2018).

Existem comandos específicos para cada tarefa do ESP-01. Por exemplo, o comando AT+GMR é utilizado para verificar o a versão do firmware do modulo. Na figura 14 são mostrados alguns dos comandos AT que podem ser utilizados e sua funções.

Figura 14 Exemplo de tabela de comandos AT

Comando AT	Função
AT+GMR	verifica versão do Firmware
AT+SYSGPIOWRITE	liga/desliga um pino do módulo
AT+SYSGPIOREAD	lê o valor de um pino
AT+UART_CUR	configura comunicação UART
AT+CWMODE_CUR	configura o modo WiFi(access point, cliente)

Fonte: <https://uploads.filipeflop.com/2018/03/taela2.png>

“Existe um manual oficial de comandos AT a ser consultado e nele Comandos que não são mais usados e não funcionam estão indicados como “deprecated” informando qual o comando atual”. (Bauermeister, 2018).

Este modo é mais utilizado quando se quer aumentar a capacidade de expansão do Esp-01 utilizando um arduino, por exemplo, pois isso permite utilizar uma maior quantidade de portas digitais e ou analógicas para o projeto.

Modo 2 – Standalone

No modo Standalone, o ESP-01 trabalha como um microcontrolador por si só, sem a necessidade de um microcontrolador externo. O software pode ser totalmente customizado de acordo com as necessidades de uso. É possível utilizar a IDE Arduino, criar um programa e gravar no módulo. É possível também utilizar as bibliotecas de WiFi para ESP8266 e desenvolver o software com base nelas por exemplo.

Esse modo é muito utilizado com as placas de desenvolvimento como NodeMCU, pois possuem mais pinos tornando os projetos mais livres para customizações.

2.10 Adaptador uart para módulo ESP-01

Esse adaptador (Figura 15) facilita o uso dos pinos VCC, GND, TX e RX do módulo ESP-01. Também faz a conversão de nível lógico entre 5V e 3,3V que é a tensão de trabalho do ESP-01. Tornando assim ser totalmente seguro conectar esse adaptador diretamente ao Arduino que trabalha em 5V por exemplo.

É um adaptador ideal para usar o ESP-01 em modo Comandos AT juntamente com o Arduino, pois para este modo necessitam apenas dos pinos, TX e RX.

Figura 15 ESP-01 Conectado ao Módulo UART



Fonte: <https://uploads.filipeflop.com/2018/03/foto-adaptador.png>

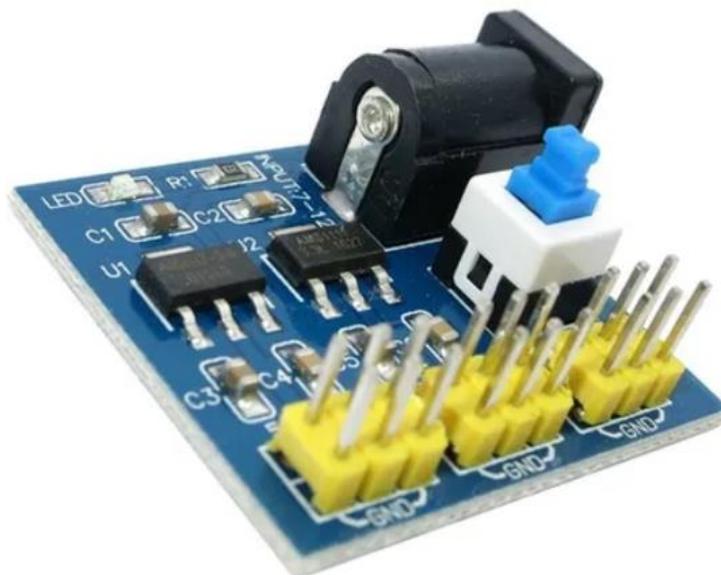
2.11 Módulo fonte multi-saídas 3.3V-5V

Para não sobrecarregar o módulo arduino, foi feito o uso de um módulo fonte de multi-saídas (Figura 16), ele possui 3 pinos que correspondem a saída de 3.3V, outros 3 pinos com a saída de 5v, e em cada um desses pinos existe um pino correspondente ao GND.

Seu funcionamento é facilitado pois possui entrada alimentação por plug P4, bastando conectar o cabo com tensão de 6,5 a 12VDC

Outro diferencial do módulo é a presença de um interruptor liga/desliga, que durante a montagem torna-se muito útil, fazendo ser desnecessário o desligamento da fonte de alimentação, evitando choques elétricos.

Figura 16 Módulo fonte 3.3-5V



Fonte: https://http2.mlstatic.com/D_NQ_NP_611852-CBT42174983475_062020-O.webp

3 DESENVOLVIMENTO

Nessa sessão é relatado o processo de desenvolvimento do projeto em questão, testes realizados, montagem do circuito e elaboração dos códigos utilizados na programação do mesmo, dentre outras informações acerca da parte pratica do trabalho.

3.1 Metodologia a materiais utilizados

Na tabela 3 serão apresentados os preços dos principais materiais usados no projeto.

Tabela 3 tabela de preços do projeto

Item	Valor
Arduino Mega	R\$ 77,72
Adapt. Esp p/5v	R\$ 25,90
Adapt Esp USB	R\$ 22,90
Célula de carga+ HX711	R\$ 25
Driver ULN2003+ Motor 5v 28byj-48	R\$ 18,90
Esp-01	R\$ 26,90
RTC DS3231	R\$ 16,90
Módulo Fonte de alimentação	R\$ 15
Total	R\$ 229,22

Fonte: Autoria Própria

Os valores aqui relatados (em Reais) não levam em consideração o frete, e foram os mais baixos encontrados dentre os praticados no mercado no momento da pesquisa (Fevereiro,2021).

3.2 Projeto

Para a realização da parte física do projeto, utilizou-se um sistema de tração, que consiste em ter uma rosca sem fim (ou fuso como na figura 17) que transporta uma quantidade de ração entre seus espaços conforme o giro do motor, o recipiente de armazenamento derrama a ração entre os espaços da rosca helicoidal e ela para de girar assim que o valor determinado do peso de ração estipulado para preencher o pote é atingido. Para servir de

estrutura, o fuso é colocado dentro de um bocal (figura 18), com largura proporcional ao do fuso para que não se comprometa o giro do mesmo.

Figura 17 Exemplo de rosca helicoidal



Fonte:

<https://content.instructables.com/ORIG/FTD/UYEK/I7W1S2NO/FTDUYEK/I7W1S2NO.jpg>

Figura 18 Exemplo de um Bocal com a Rosca Helicoidal instalada



Fonte:

<https://content.instructables.com/ORIG/FW1/8V0U/I7W1S2RB/FW18V0U/I7W1S2RB.jpg>

A Rosca Helicoidal (Figura 18) pode ser depositada em um bocal de cano de PVC por exemplo, ou em um outro tipo de recipiente que pode assim como a propria rosca helicoidal, ser confeccionado sob medida em uma impressora 3d.

Figura 19 Exemplo da visão completa do projeto



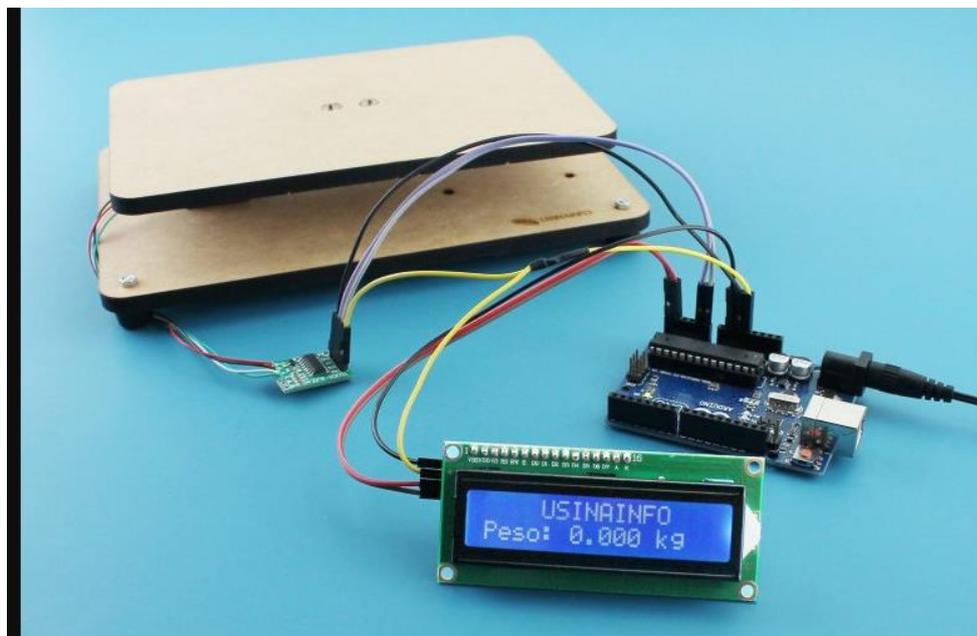
Fonte:<https://content.instructables.com/ORIG/FUA/OH0S/I88LKO3N/FUAOH0SI88LKO3N.jpg?auto=webp&frame=1&width=1024&height=1024&fit=bounds&md=89049b3af6501d3cb76d5f0a9aa7cc16>

Na figura 19 tem-se uma idéia da versão completa do projeto no quesito estrutural, onde estão demonstrados o recipiente em que é depositada a ração, e na parte de baixo uma caixa onde estaram instalados o bocal e o fuso.

3.3 Testes realizados

Para dar noção ao sistema, do peso do pote de ração, será usado uma balança feita com a célula de carga em conjunto com o HX711.

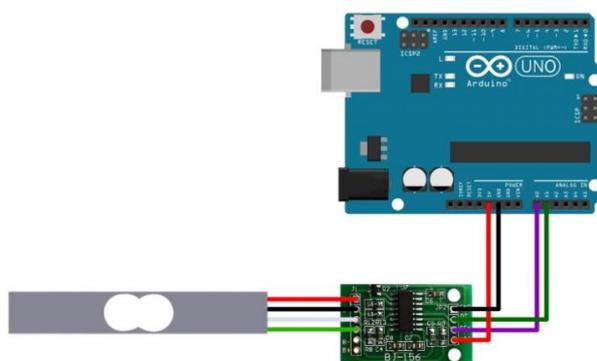
Figura 20 Exemplo de balança feita com a célula de carga



Fonte: <https://www.usinainfo.com.br/blog/wp-content/uploads/2019/09/balan%C3%A7a-768x511.jpg>

O esquema elétrico (Figura 21) para montagem da balança consiste em conectar o arduino ao HX711 o mesmo fará a comunicação com a célula de carga.

Figura 21 Esquema elétrico da balança



Fonte: <https://www.usinainfo.com.br/blog/wp-content/uploads/2019/09/esquema-ligacao-arduino-hx711-celula-de-carga-768x480.jpg>

Depois de instalada a biblioteca do HX711, deve-se implementar o código de calibração da balança (Figura 22).

Figura 22 Código de calibração da balança

```
#include "HX711.h"

#define DT A1
#define SCK A0

HX711 escala; // Relaciona a variável escala

void setup() {
  escala.begin(DT, SCK);
  Serial.begin(9600);
  Serial.print("Leitura da Tara: ");
  Serial.println(escala.read()); // Aguarda o termino de verificação do peso
  Serial.println("Aguarde!");
  Serial.println("Iniciando ...");
  escala.set_scale(); // Utiliza uma escala padrão de verificação

  escala.tare(20); // Fixa o peso como tara
  Serial.println("Insira o item para Pesar");
}

void loop() {
  Serial.print("Valor da Leitura: ");
  Serial.println(escala.get_value(10),0); // Retorna peso descontada a tara
  delay(100);
}
```

Adaptado de: <https://www.usinainfo.com.br/blog/balanca-arduino-com-celula-de-peso-e-hx711-tutorial-calibrando-e-verificando-peso/>

Os valores lidos através desse código, depois de inserir um objeto de peso conhecido e de menor valor ao do peso limite da célula de carga, irão variar entre 75000 e 75300 não extrapolando esta margem, depois será preciso escolher 8 entre eles, preferencialmente consecutivos, para se realizar uma média.

$$\text{Média} = \frac{\text{Soma dos Valores escolhidos}}{\text{Número de valores escolhidos}}$$

Com a média calculada, pode-se afeir o valor da escala, valor este que será utilizado como base para a determinação de todos os valores que iram ser posicionados sobre a balança.

$$\text{Escala} = \frac{\text{Valor da média}}{\text{Peso real do objeto de calibragem}}$$

O valor do peso utilizado para calibragem deve ser na mesma unidade à qual iremos desenvolver as pesagens da balança, e com isso pode-se adicionar os valores no código final.

A Figura 23 possui um exemplo da versão final do código da balança.

Figura 23 Exemplo de código final da balança

```

1 #include "HX711.h"
2 #include "LiquidCrystal_I2C.h";
3
4 #define DT A1
5 #define SCK A0
6
7 HX711 escala;
8
9 LiquidCrystal_I2C lcd(0x27, 16, 2);
10
11 void setup() {
12   escala.begin (DT, SCK);
13
14   lcd.init();
15   lcd.backlight();
16
17   Serial.begin(9600);
18   Serial.print("Leitura do Valor ADC: ");
19   Serial.println(escala.read()); // Aguarda até o dispositivo estar pronto
20   Serial.println("Nao coloque nada na balanca!");
21   Serial.println("Iniciando...");
22   escala.set_scale(397930.55); // Substituir o valor encontrado para escala
23   escala.tare(20); // O peso é chamado de Tare.
24   Serial.println("Insira o item para Pesar");
25 }
26
27 void loop() {
28   lcd.setCursor(4, 0);
29   lcd.print("USINAINFO");
30   lcd.setCursor(0, 1);
31   lcd.print("Peso: ");
32   lcd.print(escala.get_units(20), 3);
33   lcd.println(" kg ");
34   delay(1000);
35 }

```

Adaptado de: <https://www.usinainfo.com.br/blog/balanca-arduino-com-celula-de-peso-e-hx711-tutorial-calibrando-e-verificando-peso/>

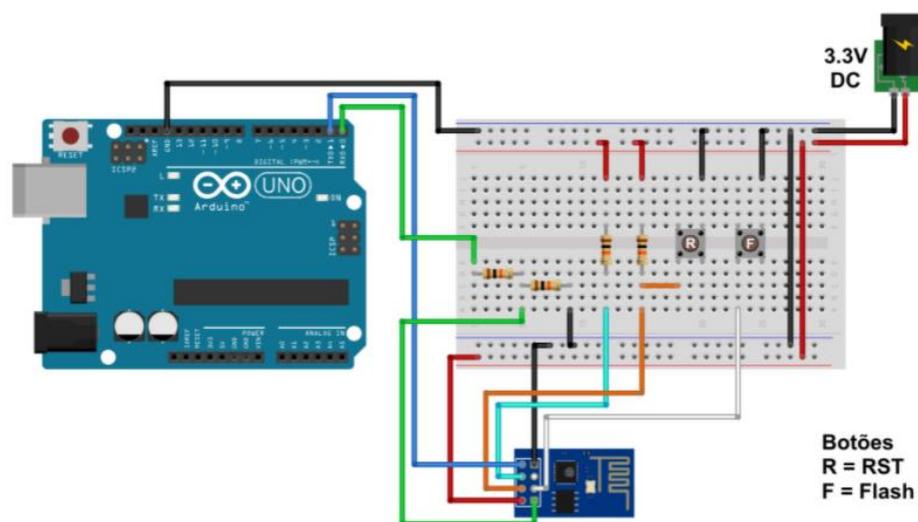
No caso do código da imagem (Figura 22), utiliza-se um LCD para mostrar o que deveria ser exibido no monitor serial, porém como o projeto usa o blynk que possui a função LCD, não será necessário usar a biblioteca *LiquidCrystal_I2C.h*

Para fazer a comunicação entre o arduino e a internet fazendo com que o Blynk possa ser utilizado a fim de controlar o projeto, é necessária a utilização do ESP-01, o mesmo se conectara ao wifi, e servirá de como um roteador para o arduino.

A primeira coisa a ser levada em consideração é a configuração do ESP, ao utiliza-lo pela primeira vez, será necessário atualizar seu *firmware*. Segundo Daniel Ribeiro (2014), em um artigo para o site do *tecmundo*: “os Firmware são um conjunto de instruções operacionais que são programadas diretamente no hardware de equipamentos eletrônicos”.

Tal atualização pode ser feita de mais de uma maneira, uma delas por exemplo, é montar um circuito (Figura 24) com dois botões em conjunto com resistores de 10k em uma protoboard e usar o arduino como interface serial.

Figura 24 Circuito de atualização de firmware do esp-01 usando o arduino



Fonte: https://uploads.filipeflop.com/2016/01/Circuito_Arduino_ESP8266.png

Ou, através de um adaptador USB serial para o ESP (Figura 25), que vem com um botão do tipo Switch, que é usado para aterrar o pino GPIO0 ao GND.

Figura 25 Adaptador USB para Esp com Botão Switch

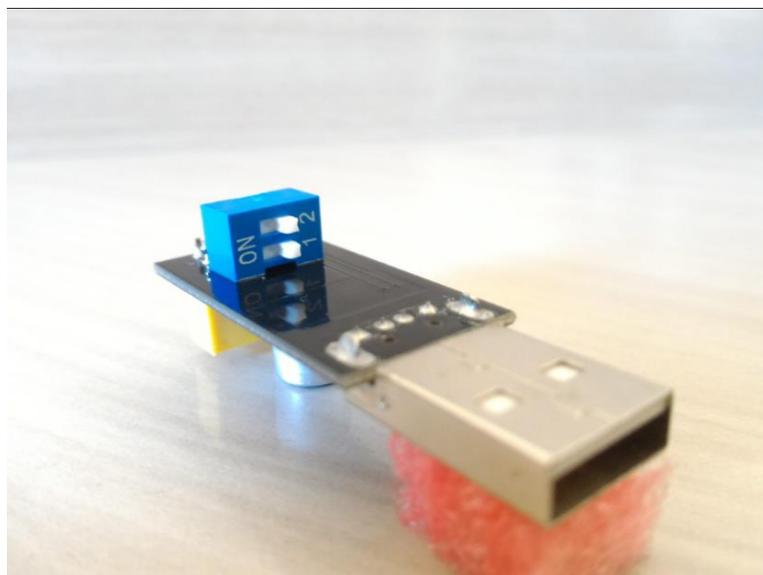


Fonte: https://http2.mlstatic.com/D_NQ_NP_973658-MLB44269256038_122020-O.webp

Existe também, a possibilidade de se fazer uma adaptação em um adaptador USB comum (sem o botão switch) que consiste em soldar um

Push Button (ou um Interruptor liga/desliga do tipo DIP) entre o GPIO0 e o GND (Figura 26).

Figura 26 Adaptador USB serial com o Botão DIP

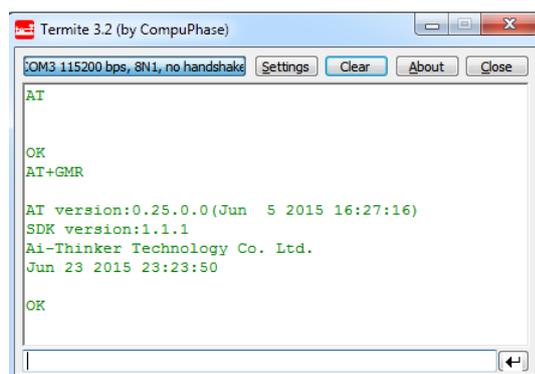


Fonte: https://blogmasterwalkershop.com.br/wp-content/uploads/2017/03/img04_modificando_o_adaptador_usb_serial_wifi_esp8266_para_upgrade_do_esp-01_arduino.jpg

Para verificar a versão do firmware antes de efetuar o upgrade, use-se o *Termite*, com a configuração de baud rate de 115200, padrão do ESP8266.

Ao digitar o comando AT na parte inferior do programa, o módulo deve responder com OK. Em seguida, digite-se AT+GMR para que seja exibida a versão atual do firmware (Figura 27).

Figura 27 Exemplo do Termite exibindo a versão do firmware

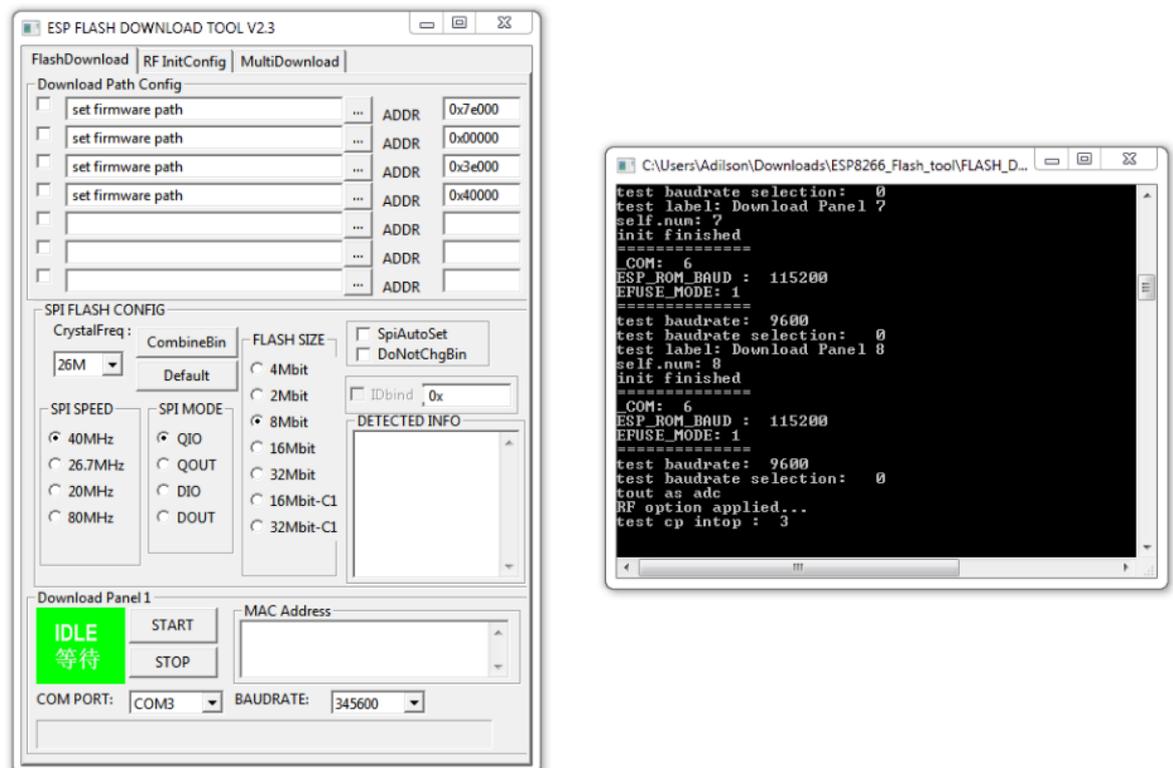


Fonte: https://uploads.filipeflop.com/2016/01/Termite_Versao_Firmware_ESP8266_old.png

A seguir é necessário fazer o download do *Esp flash download tool* e do *firmware* mais atual para o esp (ambos preferencialmente do site do espressif).

Como demonstra a imagem da Figura 28, ao executar o programa, são abertas duas telas, uma é a principal e a outra é o log.

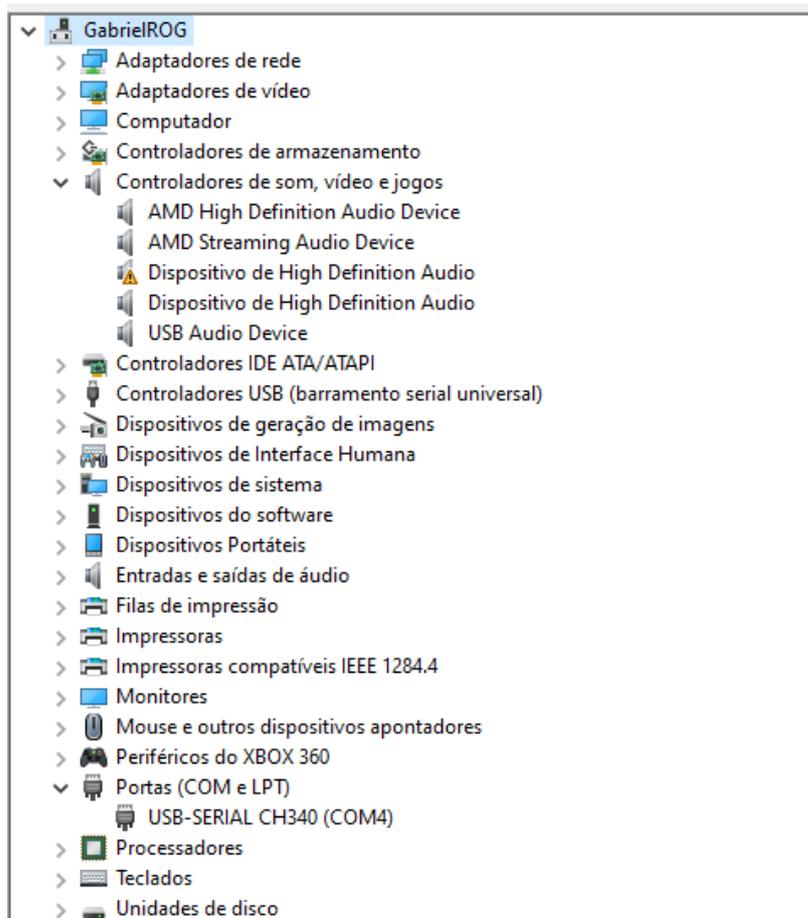
Figura 28 Tela principal e tela de log



Fonte: https://uploads.filipeflop.com/2016/01/ESP_Flash_Download_Tool_Telas.png

Deve-se selecionar a porta COM na parte inferior do programa no caso do adaptador serial, será preciso baixar um driver do CH340G para ser reconhecido), para saber qual a porta COM está sendo utilizada, basta acessar o gerenciador de dispositivos (Figura 29).

Figura 29 Gerenciador de dispositivos

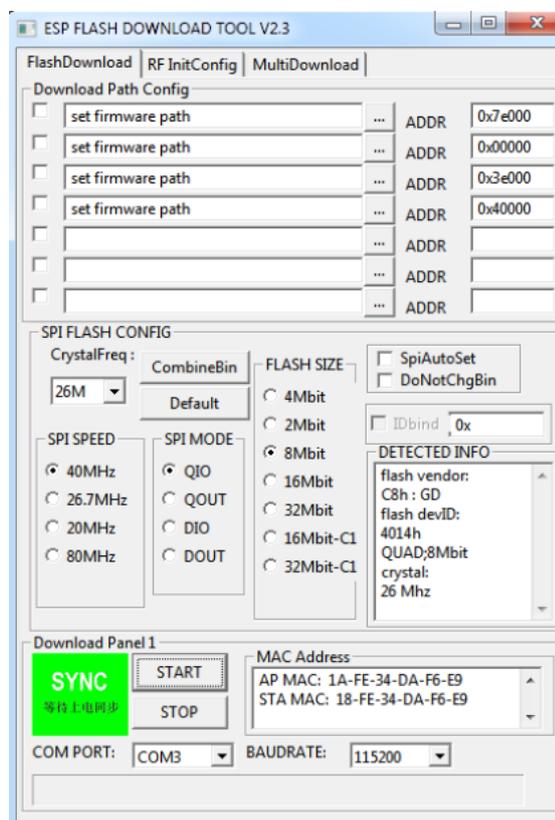


Fonte: O Grupo

No circuito, mantenha pressionado o botão “RST” e no adaptador deve-se manter pressionado o botão (ou apenas trocar ele de posição no caso do Switch), e em seguida pressione o botão Flash. Solte então o botão “RST” e solte o botão flash, para que o módulo entre em modo de programação. De volta ao programa, acione o botão START para que as

informações do módulo sejam lidas e mostradas em DETECTED INFO, no lado direito (Figura 30).

Figura 30 Exemplo da tela do ESP Flash Download Tool



Fonte:

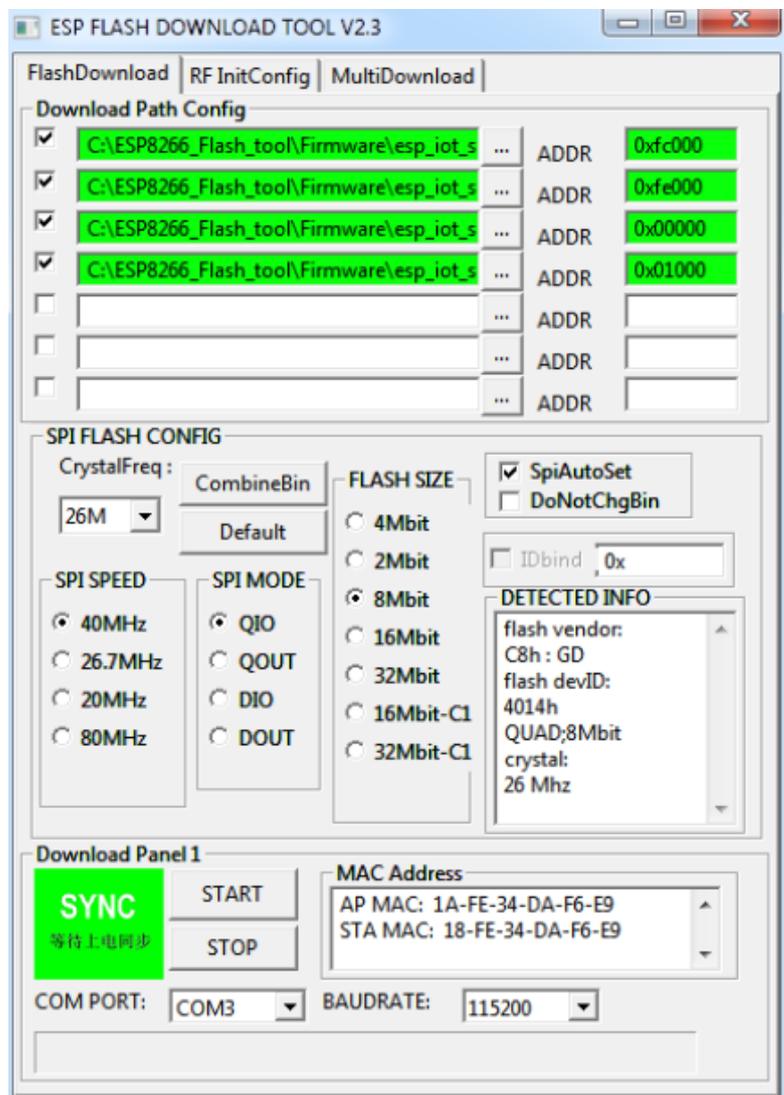
https://uploads.filipeflop.com/2016/01/ESP_Flash_Download_Tool_Lidas.png

Após marcar a opção SpiAutoSet, deve-se selecionar os 4 arquivos em “download Patch Config” que estarão no arquivo baixado anteriormente, nomeados conforme o exemplo a seguir.

- ESP8266_Flash_toolFirmwareesp_iot_sdk_v1.5.0binesp_init_data_default.bin
- ESP8266_Flash_toolFirmwareesp_iot_sdk_v1.5.0binblank.bin
- ESP8266_Flash_toolFirmwareesp_iot_sdk_v1.5.0binboot_v1.4(b1).bin
- ESP8266_Flash_toolFirmwareesp_iot_sdk_v1.5.0binat512+512user1.1024.new.2.bin

Feito isso, deve-se preencher os endereços (ADDR, Vide Figura 31).

Figura 31 Tela final do Esp Flash Download Tool

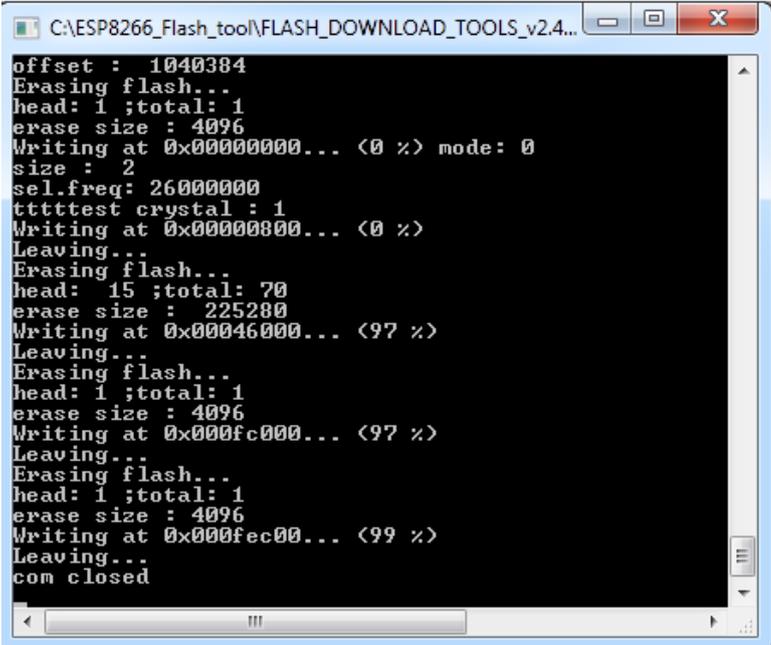


Fonte: https://uploads.filipeflop.com/2016/01/ESP_Flash_Download_Tool_Arquivos.png

Após pressionar START, o upgrade seja iniciado. A tela de console mostra o status da gravação e a porta serial sendo fechada (com closed), ao final do processo.

A Figura 32 demonstra como deve ficar a tela ao final do processo

Figura 32 tela final do log no console



```

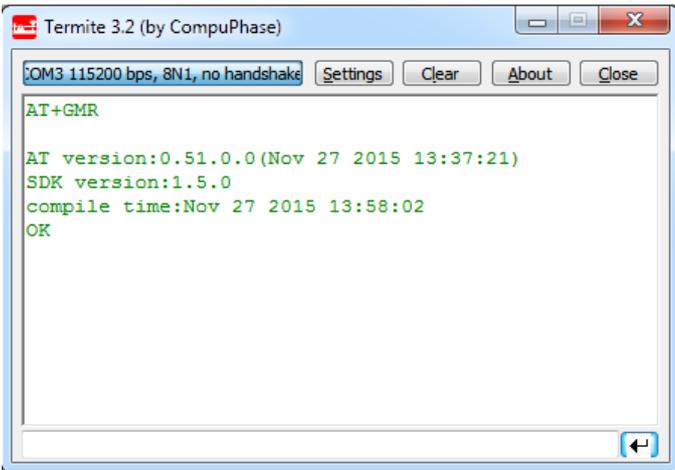
offset : 1040384
Erasing flash...
head: 1 ;total: 1
erase size : 4096
Writing at 0x00000000... <0 %> mode: 0
size : 2
sel.freq: 26000000
ttttest crystal : 1
Writing at 0x00000000... <0 %>
Leaving...
Erasing flash...
head: 15 ;total: 70
erase size : 225280
Writing at 0x00046000... <97 %>
Leaving...
Erasing flash...
head: 1 ;total: 1
erase size : 4096
Writing at 0x000fc000... <97 %>
Leaving...
Erasing flash...
head: 1 ;total: 1
erase size : 4096
Writing at 0x000fec00... <99 %>
Leaving...
com closed

```

Fonte: https://uploads.filipeflop.com/2016/01/ESP_Console_Flash.png

Após finalizado o processo, pode-se desligar módulo e liga-lo novamente sem apertar nenhum botão e verificar que o *firmware* foi atualizado com sucesso digitando o comando AT+GMR novamente no Termitte.

Figura 33 Firmware atualizado no Termitte



```

Termitte 3.2 (by CompuPhase)
COM3 115200 bps, 8N1, no handshake Settings Clear About Close
AT+GMR
AT version:0.51.0.0(Nov 27 2015 13:37:21)
SDK version:1.5.0
compile time:Nov 27 2015 13:58:02
OK

```

Fonte: https://uploads.filipeflop.com/2016/01/Termitte_Versao_Firmware_ESP8266_Versao_Nova.png

Após a atualização, pode-se mudar o *baud rate* (Taxa de comunicação) para uma mais baixa, como 9600, sendo assim possível usar comandos AT no monitor serial da IDE do arduino, para isso, basta no Termite, digitar o comando AT+UART_DEF, para fazer a alteração para 9600 de baud rate, o comando ficará:

AT+UART_DEF=9600,8,1,0,0

Para usar o aplicativo Blynk, será necessário baixá-lo na loja de aplicativos de um smartphone (Figura 34). Google Play Store no caso do Sistema Operacional Android ou na App Store no caso do IOS (Sistema Operacional da apple) Respectivamente.

Figura 34 Página do Blynk no Google Play

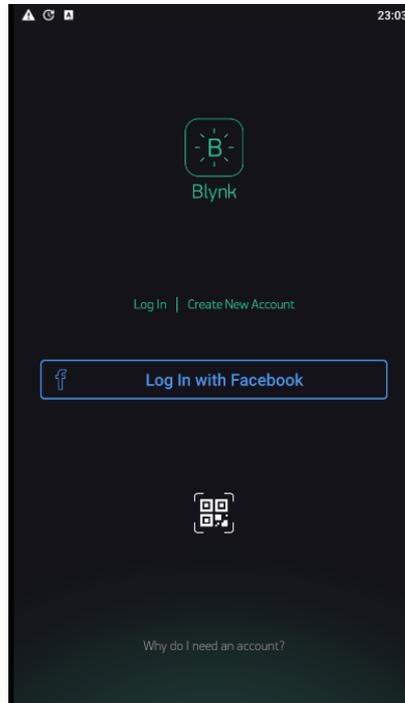


Fonte: O Grupo

Após baixado o aplicativo, ao abrir o mesmo será mostrada uma tela onde deverá ser feito o login e ou criação de uma nova conta.

A imagem da Figura 35 mostra a tela de login do aplicativo após ter sido baixado.

Figura 35 Tela de login do Blynk

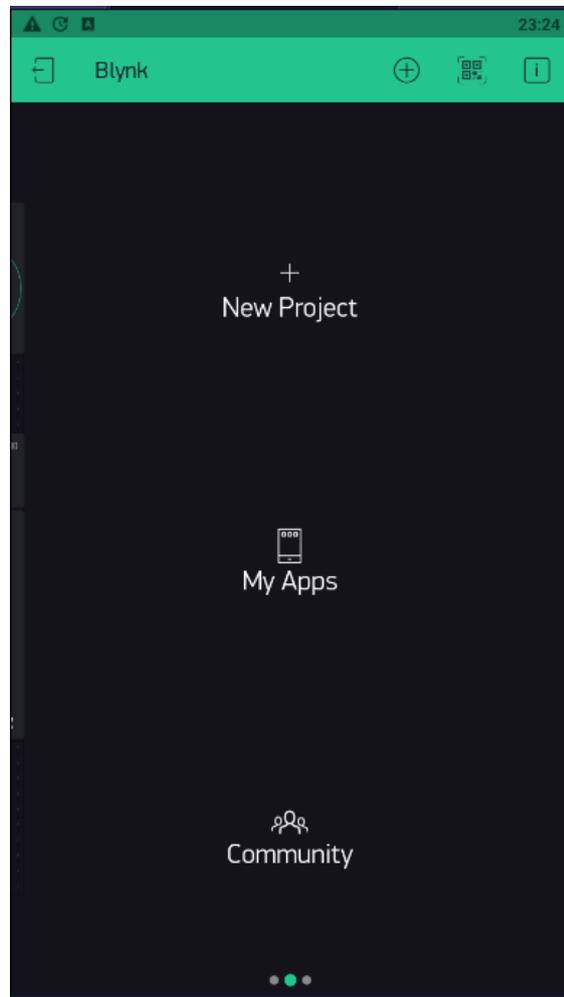


Fonte: O Grupo

Após realizada a criação da conta ou o login, o usuário será redirecionado para o menu de criação de seu próprio projeto, cada projeto possui um código de identificação próprio que será enviado ao email da conta ao qual o usuário esta logado, este código deve ser colocado em um campo específico indicado nos próprios exemplos da biblioteca do blynk.

A Figura 36 demonstra como é a tela inicial do Blynk para criação de novos projetos

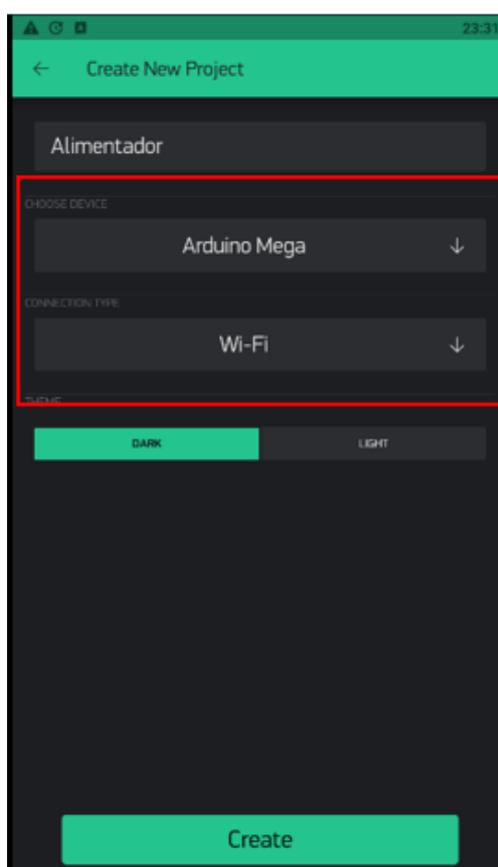
Figura 36: Menu de criação de projetos Blynk



Fonte: O Grupo

Ao clicar em “New Project”, o usuário deverá selecionar a plataforma embarcada que pretende utilizar em conjunto com o aplicativo e o tipo de conexão que realizará a comunicação entre os dois e em seguida pressionar o botão “Create” (Figura 37).

Figura 37 Tela de seleção de microcontrolador



Fonte: O Grupo

Depois disso, será exibida uma tela com alguns ícones na parte superior, ao pressionar o botão com um sinal de “+” dentro de um círculo, serão apresentados os *Gadgets* em uma barra lateral a direita da tela como demonstrado na Figura 38.

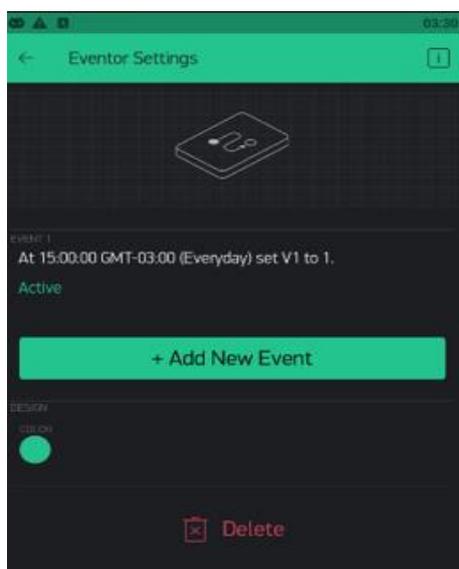
Figura 38 Botão para adição de gadgets



Fonte: O Grupo

O projeto terá dois botões, um no pino V1 representado o acionamento manual do despejo de ração, programado de 0 a 1, outro em V4 programado de 100 a 600 representando a tara da balança, um slider em V2 programado de 100 a 500, valores estes que representam os gramas de ração que irão ser despejados, e o *gadget* “Superchart” onde irá ser adicionado um gráfico nomeado de ‘peso’, programado no pino V3, e ao selecionar a opção *live*, a mesma o atualizará periodicamente com valores em gramas da quantidade de alimento no recipiente, e por último um *event*, responsável pela definição de horários em que o sistema ativará (Figura 39).

Figura 39 Exemplo de evento

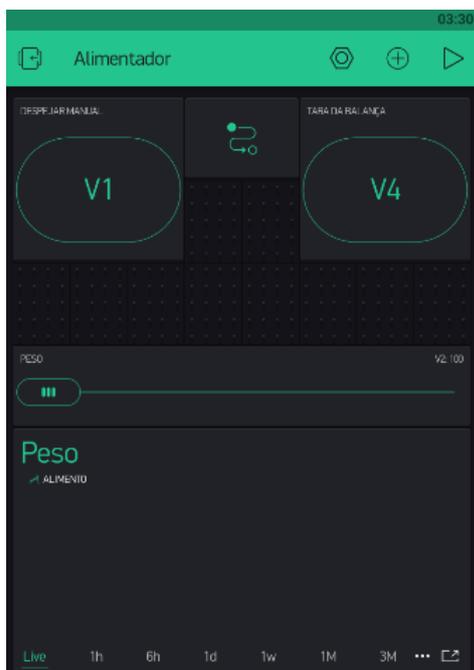


Fonte: O Grupo

3.4 Resultados obtidos

Na Figura 40, está uma representação de como ficará a tela do aplicativo em si depois de acrescentados todos os itens citados.

Figura 40 Tela final do aplicativo



Fonte: O Grupo

Na figura 41 está em forma de diagrama de blocos, a idéia de como o sistema deve operar.

Figura 41 Diagrama de blocos do funcionamento do projeto



Fonte: O Grupo

4 CONSIDERAÇÕES FINAIS

Neste capítulo, serão apresentadas as conclusões finais em relação aos resultados obtidos e sugestões para trabalhos futuros em relação ao projeto.

4.1 Conclusão

A implementação do projeto no âmbito de fazer estes reais que implicam a parte física do projeto, não pode ser realizada devido a pandemia do covid 19 que impediu a reunião do grupo com o orientador e a utilização dos recursos que estão localizados dentro das dependências da faculdade.

Não foi possível, no entanto, efetuar a construção de uma estrutura externa para o alimentador ou até mesmo a confecção da rosca sem fim, sem a utilização de uma impressora 3D e pelo fato do grupo não poder se reunir.

Ainda assim, mesmo que com algumas limitações, foi possível realizar a programação e, por exemplo, efetuar a conexão do Esp-01 com a internet; a comunicação do aplicativo com o restante do circuito e a ativação do motor por meio manual (ao pressionar o botão virtual) e automaticamente (o determinar um horário com a função *event* do Blynk) no intuito de testar o projeto, provando-o assim, que ele é funcional.

4.2 Trabalhos futuros

Com a análise dos resultados obtidos e visando aprimoramentos, foi feita uma lista com as propostas futuras quanto ao projeto:

- Adaptar o sistema para trabalhar *offline*

Desenvolver um método e incluir na programação para que o sistema trabalhe mesmo que sem internet.

- Adicionar métodos que possam manipular água e remédios

Algo que agregaria muito ao sistema seria o desenvolvimento de uma funcionalidade para manipular além da ração de maneira controlada,

também, água e remédios (para o caso do animal ter algum enfermo que deva ser tratado constantemente).

- Adição de uma câmera para monitoramento do animal

Acoplar no circuito uma câmera para que o usuário tenha acesso a imagens em tempo real do ambiente onde o sistema está instalado, passando ao usuário também, mais segurança por ter consciência do que ocorre em sua residência quanto está ausente.

- Sistema de energia baseado em energia solar

Desenvolver e adicionar ao projeto, uma maneira de usar energia solar como forma alternativa de energia, para o de ausência de eletricidade.

REFERÊNCIAS BIBLIOGRÁFICAS

ALIMENTO completo é fundamental para o bem-estar dos animais de estimação. **ABINPET**. Disponível em: <<http://abinpet.org.br/alimento-completo-e-fundamental-para-o-bem-estar-dos-animais-de-estimacao/>>. Acesso em: 21 nov. 2020.

ALVES, Gabriel Lemos. **AUTOFEEDER: ALIMENTADOR AUTOMÁTICO PARA ANIMAIS DOMÉSTICOS DE PEQUENO PORTE**. Orientador: Igor Muzeka. 2020. 49 f. TCC (graduação) - Centro Universitário Unifacvest, Lages. Disponível em: <<https://www.unifacvest.edu.br/assets/uploads/files/arquivos/db467-alves,-gabriel-lemos.-autofeeder-alimentador-automatico-para-animais-domesticos-de-pequeno-porte.-tcc-defendido-em-julho-de-2020.pdf>>. Acesso em: 06 mar. 2021.

AUTOMATIC Arduino Powered Pet Feeder. **Instructables**. Disponível em: <<https://www.instructables.com/Automatic-Arduino-Powered-Pet-Feeder/>>. Acesso em: 20 mar. 2021.

BATISTA, Daniel Lima de Melo. et al. **ALIMENTADOR AUTOMÁTICO PARA CÃES**. Orientador: Jefferson Doolan Fernandes. 2015. 4 f. TCC – Curso de Ensino Médio, Instituto Federal de Educação, Ciência e Tecnologia do Rio Grande do Norte, Parnamirim, 2015. Disponível em: <<http://sistemaolimpico.org/midias/uploads/c2c424637ae10c01d4eb153d020a4e70.pdf>>. Acesso em: 23 nov. 2020.

BAUERMEISTER, Giovanni. Guia do Usuário do ESP8266. **FilipeFlop**, 2018. Disponível em: <<https://www.filpeflop.com/blog/guia-do-usuario-do-esp8266/>>. Acesso em: 10 abr. 2021.

BLYNK. **Blynk**. Disponível em: <<http://docs.blynk.cc/>>. Acesso em: 14 mar. 2021.

BRAGA, Newton de Carvalho. A onda do Arduíno. **Instituto Newton C.Braga**, 2010. Disponível

em:<<https://www.newtoncbraga.com.br/index.php/microcontroladores/138-atmel/986-a-onda-do-arduino-col001.html>>. Acesso em: 09 mar. 2021.

BRITES, Felipe Gonçalves; SANTOS, Vinicius Puga de Almeida. **MOTOR DE PASSO**. 2008. 15 f. Curso de Engenharia de Telecomunicações (Programa de Educação Tutorial) - Universidade Federal Fluminense, Niterói, 2008. Disponível em: <<http://www.telecom.uff.br/pet/petws/downloads/tutoriais/stepmotor/stepmotor2k81119.pdf>>. Acesso em: 12 jul. 2021.

BUOGO, Douglas Rossi; JUNIOR, Joselino Xavier. **PROTÓTIPO DE ALIMENTADOR AUTOMÁTICO PARA ANIMAIS DOMÉSTICOS – GINGAPETS**. Orientador: Michael Klug. 2017. 86 f. TCC (graduação) - Curso de Tecnologia em Mecatrônica Industrial, Instituto Federal de Educação, Ciência e Tecnologia de Santa Catarina, Joinville, 2017. Disponível em: <<http://joinville.ifsc.edu.br/~bibliotecajoi/arquivos/tcc/mecind/180273.pdf>>. Acesso em: 23 nov. 2020.

COCHEÇA o MQTT, protocolo mais utilizado em aplicações IoT. **Altus**, 2018. Disponível em: <<https://www.altus.com.br/post/194/conheca-o-mqtt-2C-protocolo-mais-utilizado-em-aplicacoes-iot>>. Acesso em: 25 nov. 2020.

CONTROL 28BYJ-48 Stepper Motor with ULN2003 Driver & Arduino. **Last Minute Engineers**. Disponível em: <<https://lastminuteengineers.com/28byj48-stepper-motor-arduino-tutorial/>>. Acesso em: 12 abr. 2021.

CRUZ, Gabriela Costa da. **PROTÓTIPO DE ALIMENTADOR AUTOMÁTICO DE PETS**. Orientador: Otávio de Souza Martins Gomes. 2018. 56 f. TCC – Curso de Ensino Técnico, Instituto Federal De Educação, Ciência e Tecnologia de Minas Gerais, Formiga, 2018. Disponível em: <https://www.formiga.ifmg.edu.br/documents/2018/Biblioteca/POC_Informativa/POC-2018---Gabriela-Costa-da-Cruz.pdf>. Acesso em: 06 mar. 2021.

CUNHA, Camila Oliveira da. et al. **ALIMENTADOR AUTOMÁTICO**. Orientador: Renê Graminhani. 2017. 35 f. TCC – Curso de Ensino Técnico, Escola Técnica Estadual Jorge Street, São Caetano do Sul, 2017. Disponível em: <<http://www.jorgestreet.com.br/wp-content/uploads/2020/03/TCC-ALIMENTADOR-AUTOM%C3%81TICO.pdf>>. Acesso em: 06 mar. 2021.

DANDO uma breve análise no protocolo MQTT. **Medium**, 2017. Disponível em: <<https://medium.com/internet-das-coisas/iot-05-dando-uma-breve-an%C3%A1lise-no-protocolo-mqtt-e404e977fbb6>>. Acesso em: 24 nov. 2020.

ESP8266 – Referências de Comando AT. **Sua dica**. Disponível em: <<http://suadica.com/dica.php?d=388&t=esp8266-referencias-de-comandos-at>>. Acesso em: 25 abr. 2021.

ESP8266 AT Instrution Set. **Espressif**, 2020. Disponível em: <https://www.espressif.com/sites/default/files/documentation/4a-esp8266_at_instruction_set_en.pdf>. Acesso em: 10 abr. 2021.

FARRIS, Besty. Automatic or Remote Dog Feeder. **Hackster.io**, 2017. Disponível em: <<https://www.hackster.io/bmfarris/automatic-or-remote-dog-feeder-7d973f>>. Acesso em: 18 abr. 2021.

FIRMINO, Marcos de Souza; MATEUS, Willian da Rocha Firmino. **THYMOS PET – DOSADOR ALIMENTAR AUTOMÁTICO PARA ANIMAIS DOMÉSTICOS**. Orientador: Jorge Alberto Lewis Esswein Junior. 2020. 97 f. TCC – (graduação) – Curso de Engenharia Elétrica, Universidade do Sul de Santa Catarina, Tubarão, 2020. Disponível em : <<https://www.riuni.unisul.br/bitstream/handle/12345/9936/TCC-%20THYMOS%20PET%20%e2%80%93%20DOSADOR%20ALIMENTAR%20AUTOM%c3%81TICO%20PARA%20ANIMAIS%20DOM%c3%89STICOS.pdf?sequence=4&isAllowed=y>>. Acesso em: 28 nov. 2020.

INFORMAÇÕES gerais do setor Pet. **ABINPET**. Disponível em: <http://abinpet.org.br/infos_gerais/>. Acesso em: 21 nov. 2020.

IOT – MQTT e REST, os protocolos utilizados no mundo IoT. **Deal**, 2019. Disponível em: <<https://www.deal.com.br/blog/iot-mqtt-e-rest-os-protocolos-utilizados-no-mundo-iot/>>. Acesso em: 25 nov. 2020.

JESUS, Eliezer Silva Bonfim de. et al. **SISTEMA AUTOMATIZADO DESTINADO À ALIMENTAÇÃO DE ANIMAIS DE PEQUENO PORTE**. Orientador: Rui Carlos de Sousa Mota. 2018. 4 f. TCC – Curso de Ensino Médio, Instituto Federal de Educação, Ciência e Tecnologia da Bahia, Simões Filho, 2015. Disponível em: <<http://sistemaolimpo.org/midias/uploads/2583c53703ed3b3a74c0a548ed688e55.pdf>>. Acesso em: 20 nov. 2020.

KHATRI, Pankai. Automatic Pet Feeder Using Arduino. **CircuitDigest**, 2018. Disponível em: <<https://circuitdigest.com/microcontroller-projects/automatic-pet-feeder-using-arduino>>. Acesso em: 19 mar. 2021.

KOKSAL, Ilker. Como a tecnologia está mudando a indústria pet. **Forbes**, 2019. Disponível em: <<https://forbes.com.br/principal/2019/08/como-a-tecnologia-esta-mudando-a-industria-pet/>>. Acesso em: 09 mar. 2021.

MCCAULEY, Mike. AccelStepper library for Arduino. **AccelStepper**. Disponível em: <<http://www.airspayce.com/mikem/arduino/AccelStepper/index.html>>. Acesso: 02 mai. 2021.

MINAKAWA, Riccieli. et al. **INTRODUÇÃO AO ARDUINO**. 2012. 25 f. Grupo de Robótica – Fundação Universidade Federal de Mato Grosso do Sul, Pioneiros, 2012. Disponível em: <http://profsergiogodoy.com.br/apostilas/arduino_basico_UFMS.pdf>. Acesso em: 12 jul. 2021.

MIRANDA, Luciene. Brasil torna-se o segundo maior mercado de produtos pet. **Forbes**, 2020. Disponível em: <<https://forbes.com.br/principal/2020/08/brasil-torna-se-o-segundo-maior-mercado-de-produtos-pet/>>. Acesso em: 12 jul. 2021.

MORAIS, José. O que é ESP8266 – A família ESP e o NodeMCU. **Portal vida de silício**, 2017. Disponível em:

<<https://portal.vidadesilicio.com.br/o-que-esp8266-nodemcu/>>. Acesso em: 25 nov. 2020.

MURTA, José Gustavo Abre. Guia completo do Motor de Passo 28BYJ-48 + Driver ULN20013. **Blog Eletrogate**, 2018. Disponível em: <<https://blog.eletrogate.com/guia-completo-do-motor-de-passo-28byj-48-driver-uln2003/>>. Acesso em: 04 abr. 2021.

OCHAKOWSKI, Nádia. **PROTÓTIPO DE UM ALIMENTADOR AUTOMÁTICO PARA ANIMAIS DE ESTIMAÇÃO**. Orientador: Miguel Alexandre Wisintainer. 2007. 51 f. TCC (graduação) – Curso de Ciências da Computação, Centro de Ciências Exatas e Naturais, Universidade Regional de Blumenau, Blumenau, 2007. Disponível em: <<http://campeche.inf.furb.br/tccs/2007-l/2007-1nadiaochakowskivf.pdf>>. Acesso em: 23 nov. 2020.

OLIVEIRA, Euler. Conhecendo o Blynk. **MasterWalker**, 2017. Disponível em: <<https://blogmasterwalkershop.com.br/blynk/conhecendo-o-blynk>>. Acesso em: 14 mar. 2021.

OLIVEIRA, Euler. Modificando o Adaptador USB Serial WiFi ESP8266 para Upgrade do ESP-01. **MasterWalker**, 2017. Disponível em: <<https://blogmasterwalkershop.com.br/embarcados/esp8266/modificando-o-adaptador-usb-serial-wifi-esp8266-para-upgrade-do-esp-01>>. Acesso em: 25 abr. 2021.

OLIVEIRA, Euler. Upgrade de Firmware do WiFi ESP8266 ESP-01 através do Arduino e Conversor USB Serial. **MasterWalker**. Disponível em: <<https://blogmasterwalkershop.com.br/embarcados/esp8266/upgrade-de-firmware-do-wifi-esp8266-esp-01-atraves-do-arduino-e-conversor-usb-serial>>. Acesso em: 10 abr. 2021.

OLIVEIRA, Euler. Upgrade de Firmware no ESP8266 ESP-01 e envio de Comandos AT. **MasterWalker**. Disponível em: <<https://blogmasterwalkershop.com.br/embarcados/esp8266/upgrade-de-firmware-no-esp8266-esp-01-e-envio-de-comandos-at>>. Acesso em: 01 mai. 2021.

OLIVEIRA, Greici. NodeMCU – Uma plataforma com características singulares para o seu projeto iot. **MasterWalker**, 2016. Disponível em: <<https://blogmasterwalkershop.com.br/embarcados/nodemcu/nodemcu-uma-plataforma-com-caracteristicas-singulares-para-o-seu-projeto-iot>>. Acesso em: 22 nov. 2020.

P, Pieter. A Beginner's Guide to the ESP8266. **Github**, 2017. Disponível em: <<https://tttpa.github.io/ESP8266/Chap01%20-%20ESP8266.html>>. Acesso em: 22 mar. 2021.

PILOTTO, Danilo. **CRIAÇÃO E IMPLEMENTAÇÃO DO FIRMWARE DA PLACA PRINCIPAL PARA FUTEBOL DE ROBOS**. Orientador: Tonidandel, Flávio. 2019. 2 f. Iniciação Científica - Departamentos de Ciência da Computação, Centro Universitário FEI, São Bernardo do Campo, 2019. Disponível em: <https://fei.edu.br/sites/sicfei/2019/cc/SICFEI_2019_paper_96.pdf>. Acesso em: 12 jul. 2021.

RELATÓRIO de inteligência. **SEBRAE**, 2018. Disponível em: <<https://www.sebrae.com.br/Sebrae/Portal%20Sebrae/UFs/RJ/Anexos/Inovacao.pdf>>. Acesso em: 13 mar. 2021.

RODRIGUES, João Marcelo de Lima. **PROJETO IOT APLICADO À CONSTRUÇÃO DE UM ALIMENTADOR AUTOMÁTICO PARA ANIMAIS DOMÉSTICOS**. Orientador: Idalmir de Souza Queiroz Júnior. 2019. 12 f. TCC (graduação) – Curso de Ciência e Tecnologia, Universidade Federal Rural do Semiárido, Mossoró. Disponível em: <https://repositorio.ufersa.edu.br/bitstream/prefix/4654/1/Jo%c3%a3oMLR_ART.pdf>. Acesso em: 20 nov. 2020.

SALES, Andreia. Alimentador automático: Como escolher o melhor em 2021?. **REVIEWBOX**, 2020. Disponível em: <<https://www.reviewbox.com.br/alimentador-automatico/>>. Acesso em: 20 fev. 2021.

SANTANA, Samuel Nunes de. **PROTÓTIPO PARA MONITORAR NIVEL DE RESERVATORIO COM CELULA DE CARGA**. Orientadores: Oliveira, Ana Claudia de Melo; Gonçalves, Denilson Pereira. 2019. 12 f. TCC

(graduação) – Curso de Tecnologia da Automação Industrial, Instituto Federal de Ciência e Tecnologia de Sergipe, Lagarto, 2019. Disponível em: <<https://repositorio.ifs.edu.br/biblioteca/bitstream/123456789/1502/1/Samuel%20Nunes%20de%20Santana.pdf>>. Acesso em: 12 jul. 2021.

SANTOS, Bruno Feu de Brito. **ALIMENTADOR AUTOMÁTICO PARA ANIMAIS UTILIZANDO ARDUINO**. Orientador: Daniel Julien Barros da Silva Sampaio. 2015. 53 f. Trabalho de conclusão de curso (Bacharelado - Engenharia Elétrica) - Universidade Estadual Paulista, Faculdade de Engenharia de Guaratinguetá, 2015. Disponível em: <<https://repositorio.unesp.br/bitstream/handle/11449/139072/000864824.pdf?sequence=1&isAllowed=y>>. Acesso em: 06 mar. 2021.

SEGUNDO, Ramon Temporim de Lacerda; MORAIS, Rodrigo da Silva. **ARDUINO E MÓDULO ESP8266**. 2018. 9 f. Universidade Centro de Ensino Unificado de Brasília, Brasília, 2017. Disponível em: <<https://repositorio.uniceub.br/jspui/bitstream/235/11361/1/trabsiscom.pdf>>. Acesso em: 12 jul. 2021.

SERRANO, Tiago Medici. Introdução ao Blynk App. **Enbarcados**, 2018. Disponível em: <<https://www.enbarcados.com.br/introducao-ao-blynk-app/>>. Acesso em: 14 mar. 2021.

SOARES, Paulo. Dog Feeder Station. **Arduino Project Hub**, 2019. Disponível em: <<https://create.arduino.cc/projecthub/paulo-soares/dog-feeder-station-62ced3>>. Acesso em: 19 mar. 2021.

STRAUB, Matheus Gebert. Balança Arduino com Célula de Carga, HX711 e Display. **USINAINFO**, 2019. Disponível em: <<https://www.usinainfo.com.br/blog/balanca-arduino-com-celula-de-peso-e-hx711-tutorial-calibrando-e-verificando-peso/>>. Acesso em: 03 abr. 2021.

STRAUB, Matheus Gebert. Comedouro Automático com Arduino – Tratador para Pets. **USINAINFO**, 2020. Disponível em: <<https://www.usinainfo.com.br/blog/comedouro-automatico-com-arduino-tratador-para-pets/>>. Acesso em: 13 mar. 2021.

THOMSEN, Adilson. Controlando um Motor de Passo 5v com Arduino. **FilipeFlop**, 2013. Disponível em: <<https://www.filipeflop.com/blog/controlando-um-motor-de-passo-5v-com-arduino/>>. Acesso em: 04 abr. 2021.

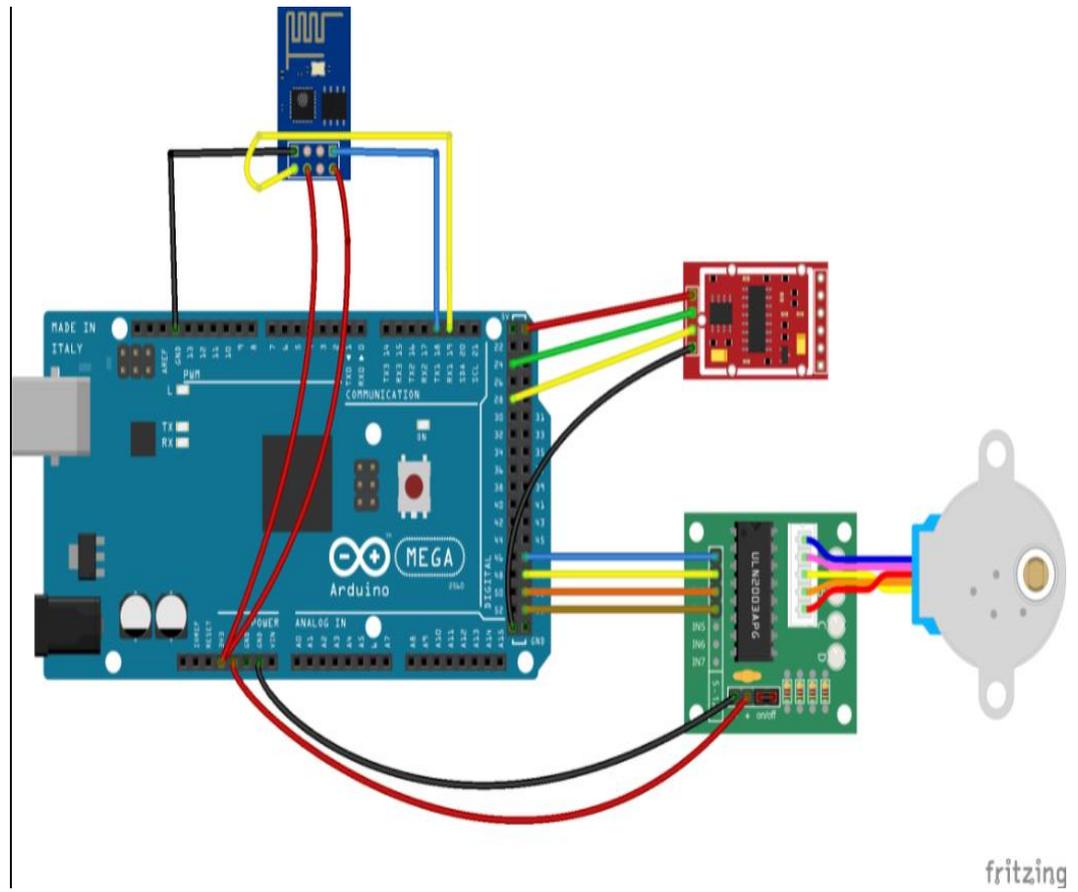
THOMSEN, Adilson. Upgrade de firmware do módulo ESP8266. **FilipeFlop**, 2016. Disponível em: <<https://www.filipeflop.com/blog/upgrade-de-firmware-do-modulo-esp8266/>>. Acesso em: 01 mai. 2021.

VARANDAS, Felipe Figueiredo. **DESENVOLVIMENTO DE UM PROTÓTIPO DE ALIMENTADOR AUTOMÁTICO PARA ANIMAIS DOMÉSTICOS - AUTOMATIC FEEDER**. Orientador: Juliano de Barros Veloso e Lima. 2019. 82 f. TCC (graduação) – Curso de Engenharia Mecatrônica, Centro Federal de Educação Tecnológica de Minas Gerais, Divinópolis, 2019. Disponível em: <<https://www.eng-mecatronica.divinopolis.cefetmg.br/wp-content/uploads/sites/195/2019/12/Felipe-Figueiredo-Varandas-2019.pdf>>. Acesso em: 27 mar. 2021.

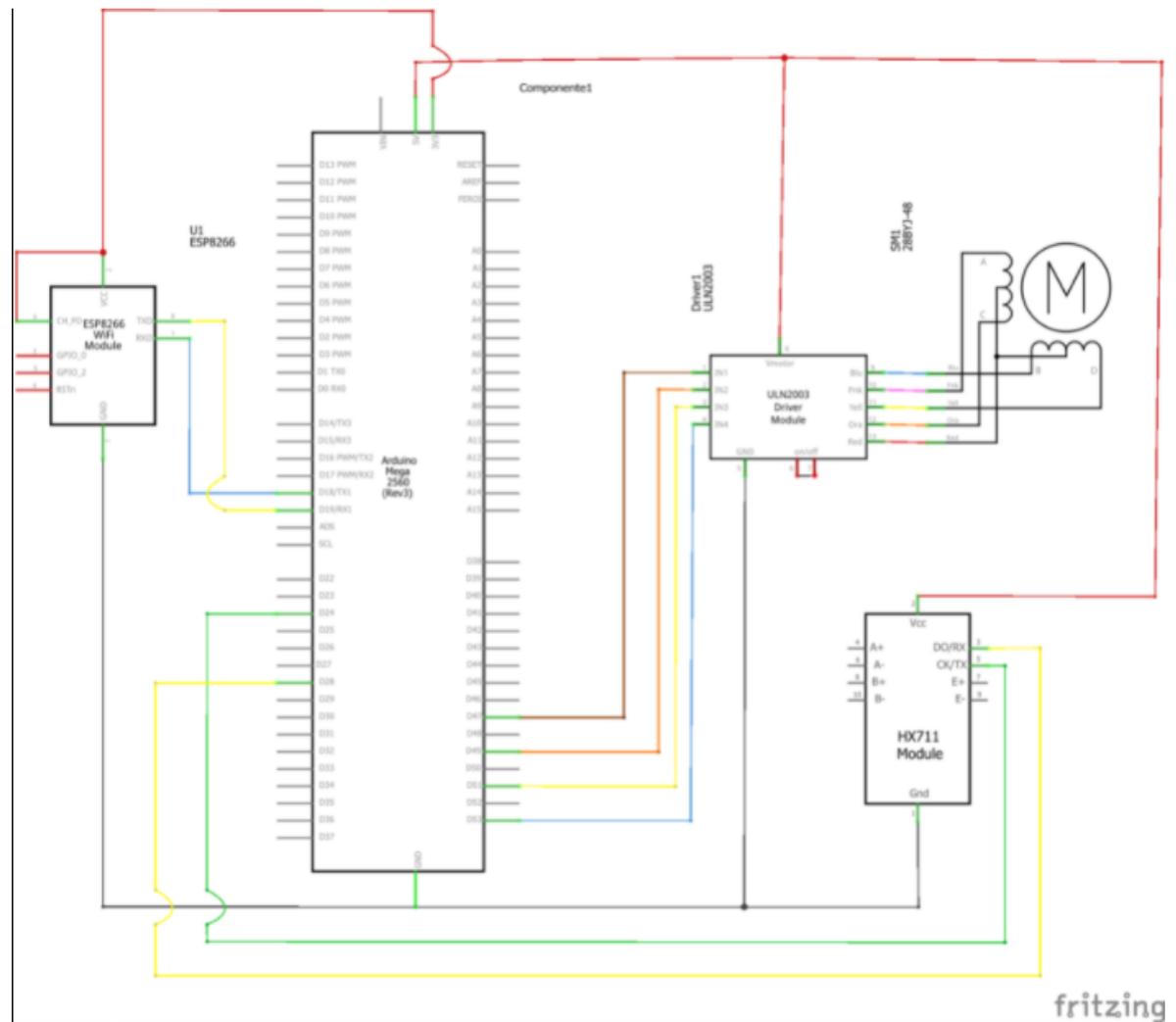
VIDAL, Vitor. Balança digital com Arduino e célula Strain Gauge. **Blog Eletrogate**, 2017. Disponível em: <<https://blog.eletrogate.com/balanca-digital-com-arduino-aprenda-a-usar-a-celula-de-carga/>>. Acesso em: 20 mar. 2021.

APÊNDICES

APÊNDICE A – Simulação do circuito no programa fritzing



APÊNDICE B – Esquema Elétrico do circuito



APÊNDICE C – Programação em arduino do projeto

```
#define BLYNK_PRINT Serial

//Definicao dos pinos do motor

#define pinSteper1 47

#define pinSteper2 49

#define pinSteper3 51

#define pinSteper4 53

//definicao dos pinos do sensor de carga

#define pinDT 28

#define pinSCK 24

//definindo escala do sensor

#define escala 0.0f

#include <ESP8266_Lib.h>

#include <BlynkSimpleShieldEsp8266.h>

#include <Stepper.h>

#include <HX711.h>

char auth[] = "lrLgOinYSmr0kmZ4Pr1M7vsNIbCT9WrO";

char ssid[] = "Gabriel";
```

```

char pass[] = "10081998";

#include <SoftwareSerial.h>

//SoftwareSerial EspSerial(2, 3); // RX, TX

// Your ESP8266 baud rate:

#define ESP8266_BAUD 115200

ESP8266 wifi(&Serial1);

//Blink timer

BlynkTimer timer;

//Objeto para controle do motor

const int stepsPerRevolution = 65;

Stepper                               motor(stepsPerRevolution
,pinSteper1,pinSteper2,pinSteper3,pinSteper4);

//objeto para o leitor de carga

HX711 scale;

//Variavel que vai abribuir um peso pra bacia de comida

int peso =0;

BLYNK_WRITE(V1)//Pino virtual acionado por evento

{

  if(param.asInt()==1){

    despejarAlimento();

  }
}

```

```
}
```

```
BLYNK_WRITE(V2) // Slider para definir o peso da bacia
```

```
{
```

```
  peso = param.asInt();
```

```
  Serial.print("Peso configurado para: ");
```

```
  Serial.print(peso);
```

```
  Serial.println("g");
```

```
}
```

```
BLYNK_WRITE(V4){
```

```
  if(param.asInt()==1){
```

```
    scale.power_up(); // LIGANDO O SENSOR
```

```
    delay(2000);
```

```
    scale.tare();//Zerando a balanca desconsiderando estruturas
```

```
    scale.power_down();
```

```
  }
```

```
}
```

```
void enviaPeso(){
```

```
  Blynk.virtualWrite(V3, getPeso()); //sending to Blynk
```

```
}
```

```
void setup()
```

```
{  
  
  //definindo velocidade do motor  
  
  motor.setSpeed(2);//rpm  
  
  
  //iniciando sensor de carga  
  
  scale.begin(pinDT, pinSCK); // CONFIGURANDO OS PINOS DA BALANCA  
  
  scale.set_scale(escala); // ENVIANDO O VALOR DA ESCALA CALIBRADO  
  
  
  // Debug console  
  
  Serial.begin(9600);  
  
  delay(10);  
  
  // Set ESP8266 baud rate  
  
  Serial1.begin(ESP8266_BAUD);  
  
  delay(10);  
  
  Blynk.begin(auth, wifi, ssid, pass);  
  
  timer.setInterval(20000L,enviaPeso);  
}  
  
void loop()  
  
{  
  
  Blynk.run();  
  
  timer.run();  
  
}  
  
  
void despejarAlimento(){  
  
  while(getPeso() < peso){
```

```
    motor.step(10);  
    delay(500);  
}  
}  
  
int getPeso(){  
    scale.power_up(); // LIGANDO O SENSOR  
    int result = scale.get_units(5);  
    scale.power_down();  
    return result;  
}
```