

**UNIVERSIDADE TECNOLÓGICA FEDERAL DO PARANÁ
DEPARTAMENTO ACADÊMICO DE ELETRÔNICA
CURSO DE TECNOLOGIA EM AUTOMAÇÃO INDUSTRIAL**

**ANTONIO CARLOS DO NASCIMENTO
BRUNO HENRIQUE GOEBEL**

DESENVOLVIMENTO DE UM CLAVICULÁRIO ELETRÔNICO

TRABALHO DE CONCLUSÃO DE CURSO

PONTA GROSSA

2018

ANTONIO CARLOS DO NASCIMENTO

BRUNO HENRIQUE GOEBEL

DESENVOLVIMENTO DE UM CLAVICULÁRIO ELETRÔNICO

Trabalho de Conclusão de Curso apresentado como requisito parcial à obtenção do título de Tecnólogo em Automação Industrial, do Departamento Acadêmico de Eletrônica, da Universidade Tecnológica Federal do Paraná.

Orientador: Prof. Dr. Josmar Ivanqui

PONTA GROSSA

2018



FOLHA DE APROVAÇÃO

DESENVOLVIMENTO DE UM CLAVICULÁRIO ELETRÔNICO

por

ANTONIO CARLOS DO NASCIMENTO

e

BRUNO HENRIQUE GOEBEL

Este Trabalho de Conclusão de Curso foi apresentado em 04 de julho de 2018 como requisito parcial para a obtenção do título de Tecnólogo em Automação Industrial. Os candidatos foram arguidos pela Banca Examinadora composta pelos professores abaixo assinados. Após deliberação, a Banca Examinadora considerou o trabalho aprovado.

Prof. Dr. Josmar Ivanqui

Orientador

Prof. Dr. Felipe Mezzadri

Membro titular

Prof. Dr^a. Virginia Helena V. Baroncini

Membro titular

Prof. Dr. Josmar Ivanqui

Responsável pelos TCC

Prof. Dr. Felipe Mezzadri

Coordenador de Curso

- A Folha de Aprovação assinada encontra-se arquivada na Secretaria Acadêmica -

Dedicamos este trabalho primeiramente a Deus, por ser essencial em nossas vidas, autor de nosso destino. Aos nossos pais, irmãos, companheiras e amigos, pelas alegrias, tristezas e dores compartilhadas.

Com vocês, as pausas entre um parágrafo e outro de produção melhora tudo o que temos produzido na vida.

AGRADECIMENTOS

Agradecemos primeiramente a Deus por ter nos concedido saúde e força para superar as dificuldades e alcançar nossos objetivos.

A oportunidade a nós oferecida para a realização deste curso.

Aos nossos pais, que são nossos exemplos de vida.

Gostaríamos de deixar registrado também, nosso reconhecimento aos familiares, esposa, namorada e filho por nos apoiar e incentivar os nossos estudos, e entender a dedicação e tempo a eles oferecido.

Aos colegas de sala que, ao final dessa jornada, tornaram grandes amigos.

Agradecemos também ao Professor Dr. Josmar Ivanqui por nos orientar, e pela sabedoria com que nos guiou nesta trajetória. Que possamos continuar com nossos estudos e encontrar ótimos profissionais de ensino como encontramos aqui nessa Instituição.

Enfim, a todos os que estiveram ao nosso lado nas horas boas e ruins, e participaram de alguma forma desta caminhada, contribuindo para um bom desenvolvimento e a conclusão deste trabalho. A todos nosso muito obrigado.

“Talvez não tenha conseguido fazer o melhor, mas lutei para que o melhor fosse feito. Não sou o que deveria ser, mas Graças a Deus, não sou o que era antes”.

Marthin Luther King

RESUMO

NASCIMENTO, Antonio Carlos Do; GOEBEL, Bruno Henrique. **Desenvolvimento de um Claviculario Eletrônico.** 2018. 55 f. Trabalho de Conclusão de Curso (Tecnologia em Automação Industrial) - Universidade Tecnológica Federal do Paraná. Ponta Grossa, 2018.

O cuidado e organização diante a bens materiais faz parte do dia a dia de todos, chaves geralmente protegem estes bens, porém nem sempre há um cuidado específico com as mesmas. O presente trabalho tem como objetivo o desenvolvimento de um claviculario eletrônico com disposição para vinte chaves, as quais ficarão armazenadas em compartimentos de um disco rotativo. Para o acesso as chaves, o usuário insere uma senha, previamente cadastrada, e o equipamento disponibilizará a chave desejada. Foi aplicada uma restrição onde cada usuário terá direito a somente uma chave por vez e somente poderá requerer outra após a devolução da requerida anteriormente. Os dados referentes a retirada e entrega da mesma, como data e hora, número da chave, usuário ficam armazenados na memória EEPROM do arduino para possíveis consultas, afim de que no caso de extravio, ou para outros fins de consulta, a chave seja identificada com facilidade.

Palavras-chave: Claviculario Eletrônico. Senha. Chave.

ABSTRACT

NASCIMENTO, Antonio Carlos Do; GOEBEL, Bruno Henrique. **Development of an Electronic Claviculary**. 2018. 55 p. Work of Conclusion Course (Technology in Industrial Automation) - Federal Technology University - Paraná. Ponta Grossa, 2018.

The care and organization on the material goods is part of the everyday life of all, keys usually protect these assets, but there is not always a specific care with the same. The present work aims at the development of an electronic claviculary in the mood for twenty keys, which will be stored in compartments of a rotating disk. To access the keys, the user enters a password, previously registered, and the equipment will provide the desired key. It was applied a restriction where each user will be entitled to only one key at a time and can only request another after the return of the previously required. The pickup and delivery data, such as date and time, number of key, user are stored in EEPROM memory of the Arduino for possible queries, so that in the event of loss, or for other purposes, the key is identified with ease.

Keywords: Electronic claviculary. Password. Key.

LISTA DE ILUSTRAÇÕES

Figura 1 - Claviculário eletrônico (Feerica).....	12
Figura 2 - iFob.....	13
Figura 3 - Desenho do disco de acrílico.....	19
Figura 4 - Disco Confeccionado em acrílico 6 mm.....	20
Figura 5 - Recipientes em Tubos de PVC 40 mm.....	20
Figura 6 - Motor de Passo Nema.....	21
Figura 7 - Base de apoio do protótipo.....	21
Figura 8 - Suporte para o motor.....	22
Figura 9 - Drive para motor de passo Allegro A4988.....	23
Figura 10 - Ligação do Drive Allegro A4988.....	23
Figura 11 - Motor de passo Kiatronics 28BYJ-48.....	25
Figura 12 - Teclado matricial de membrana 4x3.....	25
Figura 13 - Display LCD 16x2.....	26
Figura 14 - Arduíno Mega 2560.....	27
Figura 15 - Fluxograma de Programação.....	28
Figura 16 - Sensor indutivo.....	29
Figura 17 - Sensor de velocidade encoder.....	30
Figura 18 - Protótipo do Claviculário Eletrônico em testes.....	31
Figura 19 - Esquema de Ligação.....	31

LISTA DE TABELAS

Tabela 1 - Tabela de testes de resolução de micro passos e delay do PWM	24
Tabela 2 – Testes de Monitoramento de Erros de Posição.....	32
Tabela 3 – Testes Referentes a Senha.....	33

LISTA DE ABREVIATURAS

V	Volts
K Ω	Quilo ohm
mm	Milímetro
Kgf.cm	Quilograma-força por Centímetro
Kb	Quilo byte
ID	Identidade

LISTA DE SIGLAS

LCD	Visor de cristal líquido
MDF	Painel de fibra de média densidade
CNC	Controle Numérico Computadorizado
PWM	Modulação de largura de pulso
PVC	Cloreto de polivinila
EEPROM	Memória somente leitura apagável eletricamente
GND	Filtro de densidade neutra graduada
LED	Diodo emissor de luz

LISTA DE ACRÔNIMOS

STEP	Passo
DIR	Direção

SUMÁRIO

1 INTRODUÇÃO	12
1.1 TEMA	13
1.1.1 Delimitação do Tema	13
1.2 PROBLEMA	14
1.3 HIPÓTESE	14
1.4 OBJETIVOS	15
1.4.1 Objetivo Geral	15
1.4.2 Objetivos Específicos	15
1.5 JUSTIFICATIVA	15
2 DESENVOLVIMENTO	16
2.1 REVISÃO DA LITERATURA	16
2.2 MÉTODO DE EXECUÇÃO	17
2.3 DESENVOLVIMENTO DO PROJETO	18
2.3.1 Disco Em Acrílico	18
2.3.2 Recipientes De Armazenamento Das Chaves	19
2.3.3 Motor De Passo Do Disco	21
2.3.4 Base Do Protótipo	21
2.3.5 Base De Fixação Do Motor	22
2.3.6 <i>Driver</i> Do Motor De Passo	22
2.3.7 Motor De Passo E Driver Porta De Acesso	24
2.3.8 Teclado De Membrana	25
2.3.9 Display LCD	26
2.3.10 Arduíno Mega	26
2.4 FLUXOGRAMA DE PROGRAMAÇÃO	27
2.4.1 Sensor Indutivo	29
2.4.2 Sensor De Velocidade	30
2.5 RESULTADOS	32
3 CONCLUSÃO	34
REFERÊNCIAS	35
APÊNDICE A – Programação Fragmentada Em Linguagem C	37
APÊNDICE B - Custos Totais Do Claviculario Eletrônico	43
ANEXO A - Datasheet - Claviculario Eletrônico (Feerica Traka)	44
ANEXO B - Datasheet - Motores de Passo	46
ANEXO C - Arduíno Mega 2560	49

1 INTRODUÇÃO

Ter uma boa organização com bens materiais faz parte do cotidiano de todos, pois, a partir disso facilita o manuseio desses quando necessário. Bens materiais, geralmente precisam de chaves para se ter acesso (casa, carro, cofre) sendo importante assegurar-las de alguma forma. Um claviculário ou, popularmente conhecido como cofre de chaves é um excelente componente para manter chaves em segurança.

Por haver uma grande quantidade de chaves importantes, alguns estabelecimentos, tais como: concessionárias, hotéis, empresas, universidades, entre outros; muitas vezes enfrentam problemas ao gerenciar as mesmas, em função da correria do dia a dia e grande circulação de pessoas, acaba dificultando o cuidado frequente por essas chaves. Em alguns casos, por falta de tecnologia, acabam usando o que está ao alcance, um exemplo seria um cofre de chaves comum juntamente com uma folha de anotações para controlar o uso de chaves, tornando inseguro e aumentando a possibilidade de extravio, dessas gerando transtorno para resgatá-las.

Um claviculário eletrônico facilita o acesso e controle de chaves em qualquer estabelecimento, pois, partindo de sua tecnologia não há necessidade de que uma pessoa cuide das chaves, dispensando a mesma para uma outra atividade. Esta tecnologia permite segurança em suas ações, como a retirada limitada de chaves e devolução dessas, além, do rastreamento quando necessário.

Um modelo comercial de claviculário eletrônico modelo Traka S da fabricante Feerica, apresentado na Figura 1, o qual possui um custo médio entre R\$ 1.200,00 e R\$ 1.350,00 reais.

Figura 1 - Claviculário eletrônico (Feerica)



Fonte: logismarket.ind.br (2018).

Um dos dispositivos mais importantes no chaveiro eletrônico da Traka é o iFob. Este dispositivo em forma de bala, contém um chip com um ID único. Uma vez ligado a um iFob, uma chave ou um conjunto de chaves são efetivamente etiquetadas eletronicamente. As chaves são permanentemente ligadas a um iFob com um selo em aço. Cada iFob é então associado a uma posição específica no chaveiro eletrônico o qual fica preso até ser libertado por um utilizador autorizado. (FEERICA, 2018).

Conforme pode ser visualizado na Figura 2 e seu catalogo está disponível no anexo A.

Figura 2 - iFob



Fonte:logismarket.ind.br (2018).

Tendo em vista os modelos disponíveis comercialmente, este trabalho propõe desenvolver um claviculário eletrônico com um diferencial em que o usuário poderá ter acesso à uma única chave, o disponível no mercado pode ser burlado com requerimento de mais de uma chave, já que, após inserir sua senha a porta do claviculário se abre totalmente, deixando todas as chaves expostas.

1.1 TEMA

Desenvolvimento de um claviculário eletrônico com capacidade de armazenamento de 20 chaves e com disposição de apenas uma chave por vez para cada pessoa, com uma memória onde serão armazenados as senhas e o registro das chaves que foram retiradas por seus respectivos usuários.

1.1.1 Delimitação do Tema

O trabalho se destina ao desenvolvimento elétrico, eletrônico, mecânico e a programação básica de um claviculário eletrônico, cujo controle é realizado por um sistema microcontrolado.

1.2 PROBLEMA

Por que ter um claviculário? No dia a dia, convive-se com grande circulação de pessoas, como colaboradores, clientes, fornecedores, pessoas sem vínculo direto, entre outras, situações que exigem um cuidado maior com objetos de valor e ambientes restritos os quais determinadas pessoas possam frequentar.

Um dos inconvenientes está relacionado ao acondicionamento e extravio das chaves dos laboratórios do Departamento Acadêmico de Eletrônica (DAELE), assim despertando interesse em desenvolver um claviculário eletrônico, bem como no caso de desaparecimento de quaisquer bens, pode-se facilitar o seu rastreamento. Muitas vezes por esquecimento ou descuido, as chaves são extraviadas, causando transtorno para encontrá-las, já que atualmente não se tem um controle de quem pega ou às devolve, e como consequência disso as atividades são canceladas nos ambientes inacessíveis.

1.3 HIPÓTESE

Pretende-se desenvolver um claviculário eletrônico para melhorar a organização e o controle de chaves que permita acomodar vinte chaves, as quais ficarão acondicionadas e só serão liberadas a quem possa ter acesso as mesmas.

O funcionamento do claviculário age basicamente por um disco de acrílico, com vinte compartimentos designados ao acondicionamento das chaves, acoplado à um motor de passo, o qual gira em ambos os sentidos e é controlado por um microcontrolador, modelo ATmega, na plataforma Arduíno.

As informações para o uso do claviculário estarão disponíveis em uma tela, já que para o acesso às chaves será solicitada uma senha, a qual poderá ser digitada por meio do teclado composto no porta chaves.

1.4 OBJETIVOS

1.4.1 Objetivo Geral

Desenvolver um claviculário eletrônico para o DAELE, através de um disco de acrílico em conjunto com motor de passo e sistema microcontrolado com Arduino.

1.4.2 Objetivos Específicos

- Definir o *layout* do claviculário;
- Levantar materiais e componentes para a montagem;
- Realizar a montagem do claviculário;
- Executar testes, definir parâmetros para configuração do *drive* do motor de passo;
- Desenvolver o programa em linguagem C;
- Testes finais para conclusão.

1.5 JUSTIFICATIVA

Um claviculário comum permite aos usuários retirarem quantas chaves quiser sem nenhum controle, e como consequência muitas vezes um usuário retire uma chave e por esquecimento não devolva. O benefício esperado, é que ocorra a diminuição do extravio de chaves do departamento, já que o claviculário eletrônico irá armazenar os dados referente ao usuário que retirou ou devolveu a chave, com data e horário da atividade.

2 DESENVOLVIMENTO

A palavra “Claviculario” vem do latim, *clavicularius*, também conhecido como porta-chaves, ou cofre de chaves. Há clavicularios simples, utilizado em qualquer ambiente somente para a organização de chaves, como também existem aqueles de maior segurança, ou seja, clavicularios eletrônicos com acesso apenas por intermédio de senhas.

Ao falar que os clavicularios são comuns, é porque são fáceis de serem encontrados em recepções e/ou portarias de prédios, estacionamentos, universidades, empresas e hotéis que detém grande número de chaves a serem controladas. Um claviculario comum dá acesso a qualquer pessoa, considerado assim com um baixo nível de segurança, sendo assim, permite o fácil extravio de chaves. Já o claviculario eletrônico envolve uma capacidade de segurança e organização muito maior, somente pessoas autorizadas têm acesso, pois geralmente é composto por fechaduras eletrônicas controladas por senhas.

Por mais que a função de um claviculario pareça simples, ele pode contribuir muito para um bom controle e uma boa organização das chaves. Esse controle elimina a necessidade de se manter uma pessoa monitorando o uso das chaves, proporcionando maior segurança para as mesmas, além de reduzir custos com colaboradores.

Nos dias atuais, existem chaves que controlam bens e ambientes que possuem valores elevados, com isso, a separação e organização das mesmas podem otimizar o tempo. Esse tipo de chave não pode simplesmente ser jogada dentro de uma gaveta qualquer, causando transtorno ao procura-la posteriormente, mas sim garanti-la em um ambiente de extrema segurança.

2.1 REVISÃO DA LITERATURA

O claviculario é um cofre desenvolvido para proteger chaves. A proteção que um claviculario oferece pode ajudar na organização de universidades, condomínios, empresas, estacionamentos, entre outros, evitando a perda de chaves importantes e mantendo o controle sob as mesmas.

A origem do cofre vem do Antigo Egito, se tornando necessário desde que surgiram os primeiros objetos de valor. O baú foi utilizado por muitos anos como modelo de cofre universal, fechado com chave. Os primeiros cofres de baú faziam parte de navios piratas, recheados de moedas de ouro e joias, entre os séculos 16 e 17. O cofre mais antigo pertenceu a Cipselus, o tirano de Corinto, há cerca de 2.700 anos. (PM COFRES, 2016).

Ao longo da história, há milhares de anos, existiram diferentes tipos de cofres, mas o tal como conhecemos atualmente, “cofre moderno”, tem pouco mais de 170 anos.

No século XIX, em 1829, Alexandre Fichet (1799 – 1862), inventou uma fechadura inviolável, de alta segurança, capaz de resistir fogo, água e, conseqüentemente, aos ladrões e em 1844, em Paris (França), nasceu o cofre moderno. Já em 1879, cria-se a primeira caixa forte para bancos. (CURIOSFERA, 2018).

Com o passar dos anos, a tecnologia evoluiu muito, porém, a insegurança também aumentou. Atualmente os cofres possuem fechaduras com combinações numéricas sofisticadas, aberturas com atraso e automatismos de alta tecnologia. Com os avanços tecnológicos, a necessidade de segurança de informações e objetos cresceu, os cofres foram se tornando cada vez mais comuns no dia a dia das pessoas, surgindo assim algumas derivações dos cofres comuns, como seria o caso do claviculário, que basicamente é um cofre comum destinado a chaves.

Os claviculários comuns possuem fechaduras mecânicas, e são encontrados em diversos lugares por ajudarem no controle e na organização das chaves, contudo, atualmente existem diversas versões de claviculários desde os mais simples até os mais sofisticados controlados eletronicamente, dependendo da segurança necessária.

2.2 MÉTODO DE EXECUÇÃO

Para execução do protótipo foi elaborado a melhor forma de dispor uma única chave por vez para o usuário, dessa forma optou-se por elaborar um disco de acrílico com vinte compartimentos de armazenamento, com uma porta de acesso para apenas

uma chave onde as outras ficam isoladas dentro do compartimento do dispositivo sem acesso.

O projeto inicial tem por objetivo principal dispor apenas uma chave por usuário, assim o mesmo terá direito de requerer outra chave só após a devolução da requerida anteriormente, para não elevar os custos e não deixar o projeto muito complexo optou-se por utilizar um motor de passo com *drive A4988* para acionamento do movimento do disco e por se tratar também de um componente de precisão, foi utilizado outro motor de passo de menor potência para acionamento da porta de acesso do compartimento das chaves, o qual oferece também uma boa precisão e um torque de 0,34 kgf.cm .

Para a programação, armazenamentos de dados (senhas, usuário, data, hora, chave) e controle do dispositivo foi utilizado uma plataforma Arduino Mega 2560 produto de excelente custo benefício, de fácil acesso, e com plataforma livre para programação. O acesso da senha e número da chave foi realizada através de um teclado numérico matricial 4x3 em conjunto com um *display* LCD 16X2 para exibir informações que interessem ao usuário, onde maiores informações ficam disponíveis no Anexo D.

Para o protótipo da base do claviculário foi utilizado uma placa de MDF com 6 mm de espessura, sendo resistente e de baixo custo. Já o disco e a tampa de fechamento superior foram feitos de acrílico 6 mm para um acabamento melhor e a visualização do funcionamento do dispositivo. Poderia ser utilizado outros materiais mais resistentes, mas definiu-se a utilização desses materiais por se tratar de um protótipo.

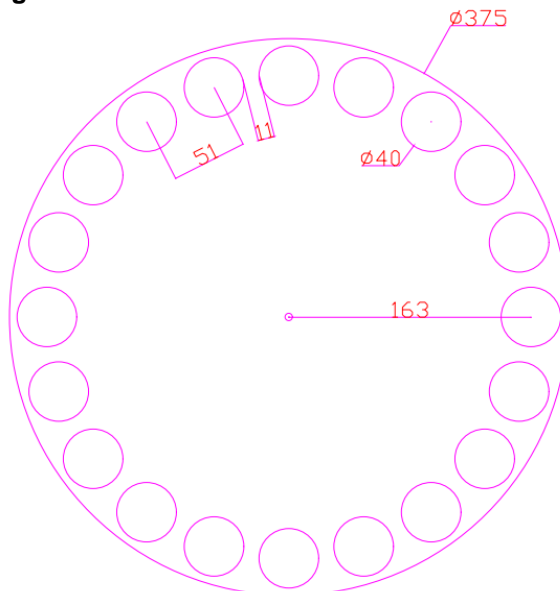
2.3 DESENVOLVIMENTO DO PROJETO

2.3.1 Disco Em Acrílico

Após desenvolver alguns modelos em madeira do claviculário para coleta de dados de qual seria a melhor opção de armazenamento das chaves optou-se por utilizar um disco de acrílico, pois este material possui resistência mecânica, é leve e além disto, fácil de cortar e furar. O disco possui 375 mm de diâmetro onde foram

dispostos 20 furos separados em ângulo de 18° com 40 mm cada um deles, com uma distância entre si de 11 mm conforme pode ser observado na Figura 3.

Figura 3 - Desenho do disco de acrílico



Fonte: Autoria Própria (2018).

Com o projeto definido, os cortes foram realizados fazendo uso de uma máquina CNC na chapa de acrílico.

2.3.2 Recipientes De Armazenamento Das Chaves

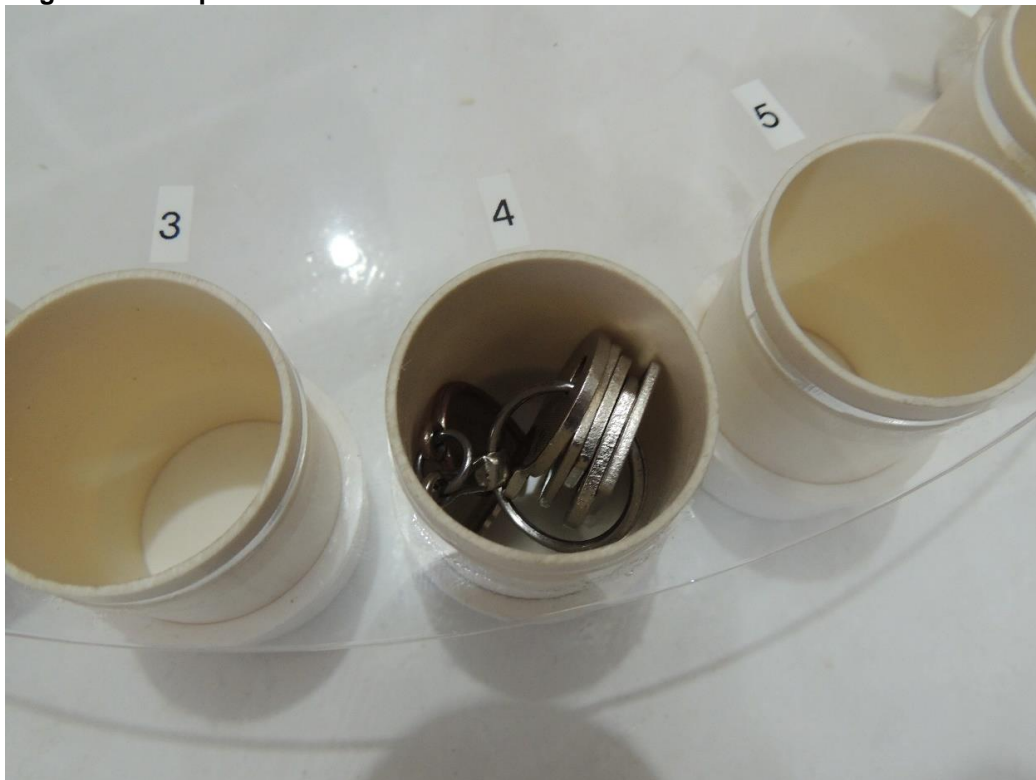
Já para armazenar as chaves foram confeccionados recipientes em tubos de PVC 40 mm branco com altura de 65 mm para melhor acomodar as chaves, como são mostrados nas Figura 4 e Figura 5.

Figura 4 - Disco Confeccionado em acrílico 6 mm



Fonte: Autoria Própria (2018).

Figura 5 - Recipientes em Tubos de PVC 40 mm



Fonte: Autoria Própria (2018).

2.3.3 Motor De Passo Do Disco

Para o movimento de rotação foi utilizado um motor de passo Nema 17-1,1 kgf.cm, apresentado na Figura 6, sendo uma solução que oferece a melhor relação custo benefício quando há necessidade de movimento com precisão de posicionamento. Dada a sua robustez possui baixíssimo índice de manutenção, otimizado para trabalhar com resoluções de micro passo que garantem uma precisão maior ao sistema de movimentação. (NEOMOTION, 2018).

Figura 6 - Motor de Passo Nema

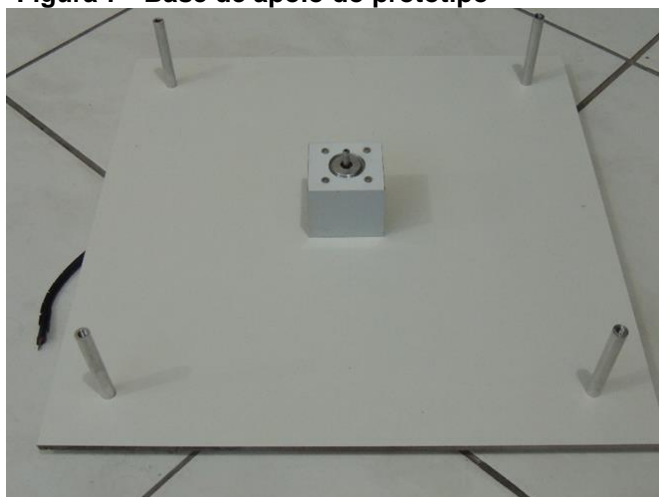


Fonte: robocore.net (2018). Adaptado.

2.3.4 Base Do Protótipo

Para acomodar o disco e outros componentes do protótipo foi utilizado uma placa de MDF com as dimensões de 400 x 400 x 6 mm para ser a base, como pode ser visto na Figura 7.

Figura 7 - Base de apoio do protótipo



Fonte: Autoria Própria (2018).

2.3.5 Base De Fixação Do Motor

Já na fixação do motor de passo, foi confeccionado um suporte em alumínio devido a este material ter uma boa condutividade térmica para ajudar no resfriamento do motor caso seja utilizado com uma frequência muito grande e também por ser leve e ter um ótimo *design*, como se observa na Figura 8.

Figura 8 - Suporte para o motor

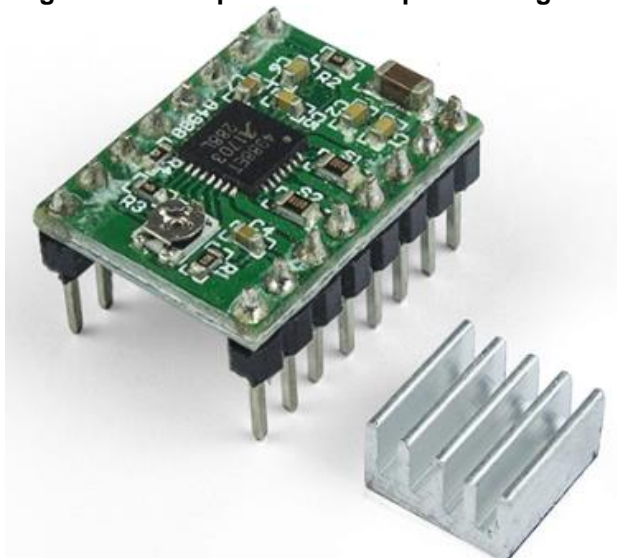


Fonte: Autoria Própria (2018).

2.3.6 Driver Do Motor De Passo

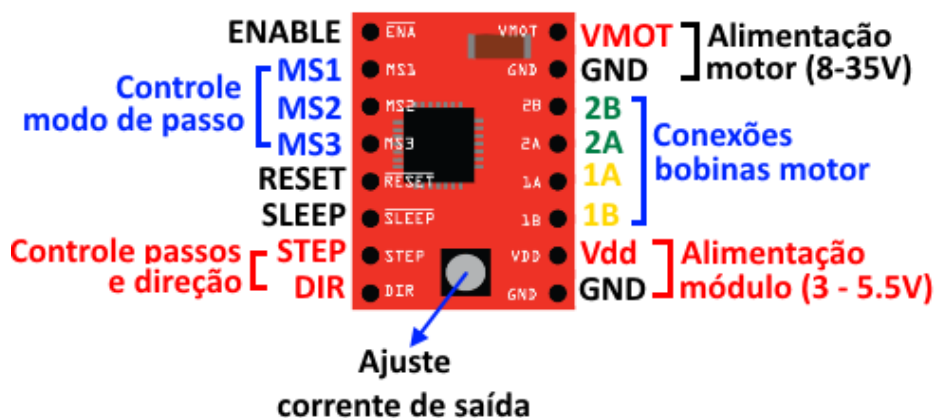
Para controlar esse motor, utilizou-se um driver para motor de passo *Allegro A4988*, conforme mostrado na Figura 9 que não é o *driver* específico, mas de qualquer forma garante um bom torque nominal. Esta placa possui um *chip* de fácil utilização, ideal para motores de passo bipolares. Ela possui proteção e controle do circuito contra sobre corrente que possam danificar o *drive*, o motor e também possuem a opção de utilizar um resistor interno de *pull down* de 100 k Ω no pino de seleção. O controle básico do motor de passo é feito por meio de dois pinos do micro controlador, ligados aos pinos *STEP* (passo) e *DIR* (direção) do módulo A4988, conforme pode se verificar na pinagem do mesmo, mostrada na Figura 10. Uma das grandes vantagens desse módulo é a possibilidade de utilizar micro passos, função configurável através dos pinos específicos.

Figura 9 - Drive para motor de passo Allegro A4988



Fonte: robocore.net (2018). Adaptado.

Figura 10 - Ligação do Drive Allegro A4988



Fonte: arduinoecia.com (2015). Adaptado.

Para o projeto foram realizados vários testes alterando as resoluções de micro passos e a frequência controlada por uma técnica conhecida como *Pulse Width Modulation (PWM)*, traduzido como modulação por largura de pulso, pois seu acionamento é feito através de pulsos elétricos que ativam sequencialmente suas bobinas, fazendo o rotor se alinhar com as mesmas e assim provocando um deslocamento do mesmo. Para dar uma volta completa são necessários 200 passos, ou seja, cada passo equivale a $1,8^\circ$ e um meio passo a $0,9^\circ$ e assim sucessivamente, o *delay* citado na tabela é o tempo entre um passo e outro, onde o mesmo foi testado e apenas o melhor resultado foi apresentado na Tabela 1 demonstrada na sequência.

Tabela 1 - Tabela de testes de resolução de micro passos e delay do PWM

Passo	Passo completo	Meio Passo	Quarto de passo	Oitavo de Passo
20 Posições				
Erro	Meio Passo	Quarto de Passo	Oitavo de Passo	Nenhum
Tempo	20:43 s	20:41 s	24:21 s	32:50 s - 21:45 s
Delay PWM	50 ms	25 ms	15 ms	10 ms - 07 ms
10 Posições				
Erro	Meio Passo	Oitavo de Passo	Nenhum	Nenhum
Tempo	09:64 s	10:20 s	12:15 s	16:09 - 11:06 s
Delay PWM	50 ms	25 ms	15 ms	10 ms - 07 ms
05 Posições				
Erro	Quarto de Passo	Quarto de Passo	Nenhum	Nenhum
Tempo	05:19 s	05:30 s	06:12 s	08:02 s - 05:92 s
Delay PWM	50 ms	25 ms	15 ms	10 ms - 07 ms
01 Posição				
Erro	Meio Passo	Oitavo de Passo	Oitavo de Passo	Nenhum
Tempo	01:67 s	01:26 s	01:38 s	01:65 s - 01:43 s
Delay PWM	50 ms	25 ms	15 ms	10 ms - 07 ms

Fonte: Autoria Própria (2018).

Este teste teve como objetivo verificar a melhor configuração para utilizar no motor de passo, levando em consideração o tempo, em segundos, que levaria para dar uma volta completa, ou de um ponto a outro, alterando a frequência do pulso *PWM*, em milissegundos. Também teve como objetivo a verificação de erros no final do ciclo, ou seja, após uma volta completa, ou entre um ponto e outro, o posicionamento inicial permaneceria o mesmo.

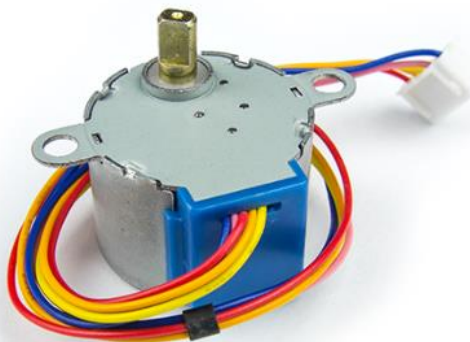
Pode-se observar a partir dos testes realizados na Tabela 1, a melhor opção para trabalhar com o motor seria com a resolução configurável no *drive*, a um oitavo de passo (*eight step*) onde não se obteve nenhum erro, mesmo com o tempo do ciclo consideravelmente acima dos demais. Porém, ao utilizar outras configurações, no final do ciclo seria necessário corrigir o erro, o que levaria mais algum tempo de reposicionamento, além de complicar a programação.

2.3.7 Motor De Passo E Driver Porta De Acesso

Na Figura 11 apresenta o *drive Allegro A4988*, em conjunto com um pequeno motor de passo de apenas 0,34 kgf.cm, para acionamento da porta de acesso do

compartimento das chaves, este motor tem seu funcionamento similar ao já apresentado.

Figura 11 - Motor de passo Kiatronics 28BYJ-48



Fonte: robocore.net (2018). Adaptado.

2.3.8 Teclado De Membrana

Para acesso as chaves depositadas no claviculário, o usuário insere uma senha, e posteriormente, se a mesma estiver correta, o usuário digita o número da chave correspondente ao laboratório desejado. Para isso instalou-se um teclado matricial membrana 4 linhas por 3 colunas Figura 12. Este teclado internamente possui uma sequência de chaves (o teclado utilizado especificamente são 12), dispostas em forma de matriz. Cada chave quando acionada faz a ligação entre uma linha e uma coluna, e essa informação é enviada ao micro controlador (nesse caso o Arduino Mega 2560), que por meio do programa interpreta qual tecla foi pressionada.

Figura 12 - Teclado matricial de membrana 4x3

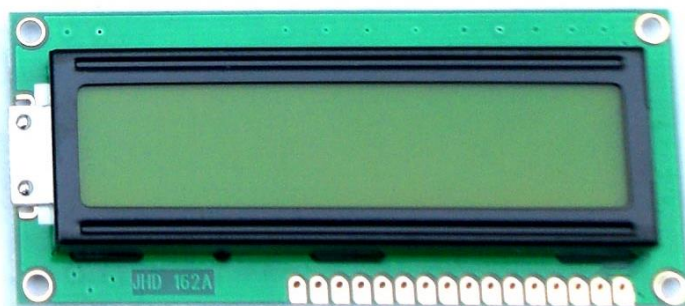


Fonte: arduinoocia.com (2015). Adaptado.

2.3.9 Display LCD

Como o usuário estará digitando em um teclado, as informações necessárias aparecerão em um *display* básico de 16 caracteres por 2 linhas (Figura 13), onde possui texto preto sobre fundo verde, o qual utiliza o extremamente comum *chipset HD44780* de interface paralela onde caso necessário o código da interface está disponível gratuitamente no site do fabricante. Para realizar a interface do *display LCD*, foi necessário utilizar alguns pinos de entrada/saída (I/O), além dos pinos necessários para a conexão do *LED backlight* que será basicamente utilizado para o contraste do *display*.

Figura 13 - Display LCD 16x2



Fonte: soldafria.com (2018). Adaptado.

2.3.10 Arduíno Mega

Para a aplicação dos recursos foi utilizado uma placa Arduíno Mega 2560, conforme pode ser visto na Figura 14, sendo uma plataforma de computação *open-source*, baseada numa placa com entradas e saídas digitais e analógicas. A plataforma Arduíno possui um próprio ambiente de desenvolvimento que implementa a linguagem de programação C, e esta pode ser usada para desenvolver objetos interativos autônomos ou pode ser conectado a um software em seu computador. (Anexo D).

Figura 14 - Arduíno Mega 2560



Fonte: robocore.net (2018). Adaptado.

Um outro recurso que far-se-á uso no Arduíno é a gravação de dados na memória *EEPROM* do micro controlador. Esse método de armazenamento foi utilizado para gravar os dados como por exemplo: senhas, nomes, chaves, data, hora e quem retirou a chave pela última vez, quando a chave foi retirada, entre outros. Esses dados são protegidos na memória, pois caso ocorra uma falta de energia elétrica ainda assim garante-se a integridade dos dados armazenados, ou mesmo em casos de travamento do Arduíno.

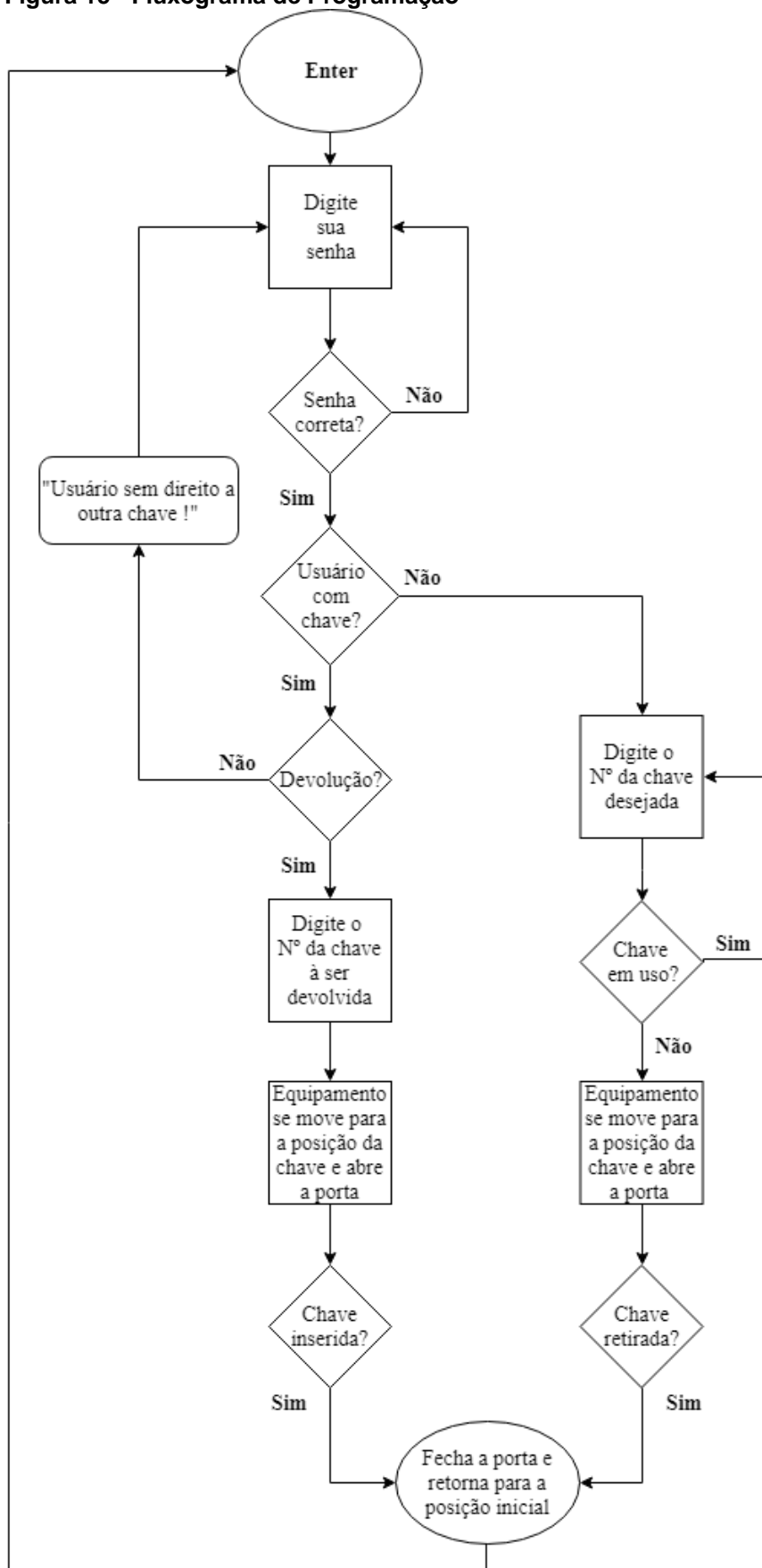
A *EEPROM* (Memória somente de leitura programável apagável eletricamente) é uma memória que pode ser apagada e reprogramada várias vezes, apesar de possuir um limite no número de ciclos de leitura/escrita. A capacidade de memória *EEPROM* varia conforme o modelo da placa, o Arduíno Mega (ATMega2560) possui uma memória de 4 KB.

A programação foi realizada utilizando a Linguagem C, que é uma linguagem de programação mais popular, em conjunto com a plataforma Arduíno e algumas de suas bibliotecas para facilitar e organizar melhor a programação.

2.4 FLUXOGRAMA DE PROGRAMAÇÃO

A metodologia da programação tem como finalidade a Figura 15 a seguir, que demonstra o fluxograma de programação.

Figura 15 - Fluxograma de Programação



Fonte: Autoria Própria (2018).

A programação a qual se encontra fragmentada no Apêndice A, seguiu a lógica do fluxograma apresentado na Figura 15 onde o usuário entra com sua senha e se a mesma estiver correta, dará sequência a execução informando o número da chave a qual se deseja. Caso a mesma estiver em uso, aparecerá a mensagem “Chave em uso”, caso digitar um valor que não corresponda a nenhuma chave aparecerá a mensagem “Chave inexistente” e retornará para digitar um novo número que corresponde a chave de determinado laboratório. Outra mensagem que aparece ao usuário é “Senha incorreta” caso tenha esquecido ou errado a mesma.

2.4.1 Sensor Indutivo

Se o número do laboratório estiver correto e a chave sem uso o motor de passo aciona automaticamente e para na posição requisitada. Após chegar na posição a porta do claviculário, irá se abrir para que o requerente tenha aceso a chave. Após a mesma ser retirada a porta se fecha automaticamente, para isso utilizou-se um sensor indutivo como demonstrado na Figura 16. Este sensor após detectar um objeto de metal (chave), gera um sinal na saída, que pode ser lido e interpretado por um micro controlador como o arduíno para ativar o motor de passo da porta de aceso. Ao fechamento da porta o dispositivo retornará à posição inicial, neste caso, determinou-se a posição inicial da chave como, a posição 01.

Figura 16 - Sensor indutivo

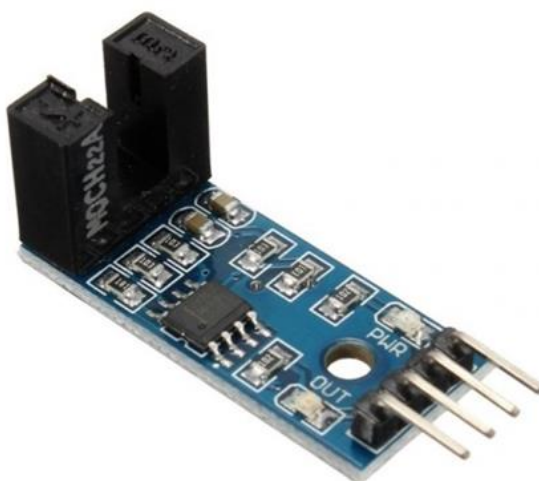


Fonte:baudaeletronica.com (2018). Adaptado.

2.4.2 Sensor De Velocidade

A necessidade de que ao retirar ou devolver uma chave o dispositivo sempre retorne para a posição inicial, é que o programa possui um ponto de referência, caso ocorra uma queda de tensão no meio do processo ou ocorra alguma interferência no sistema, assim também servirá como auto calibração para o programa e ao alimentar com uma fonte de tensão ele sempre se posicione corretamente, para tanto, instalou-se outro sensor, um sensor de velocidade que pode ser utilizado para realizar medições de rotação de motores, contagem de pulsos e como controlador de posicionamento que é o caso, este também pode ser utilizado em conjunto com o Arduíno, conforme pode ser observado na Figura 17.

Figura 17 - Sensor de velocidade encoder



Fonte:baudaeletronica.com (2018). Adaptado.

Como base superior de fechamento do protótipo, usou-se uma chapa de acrílico com dimensões de 400 x 400 x 6 mm como demonstrado na Figura 18.

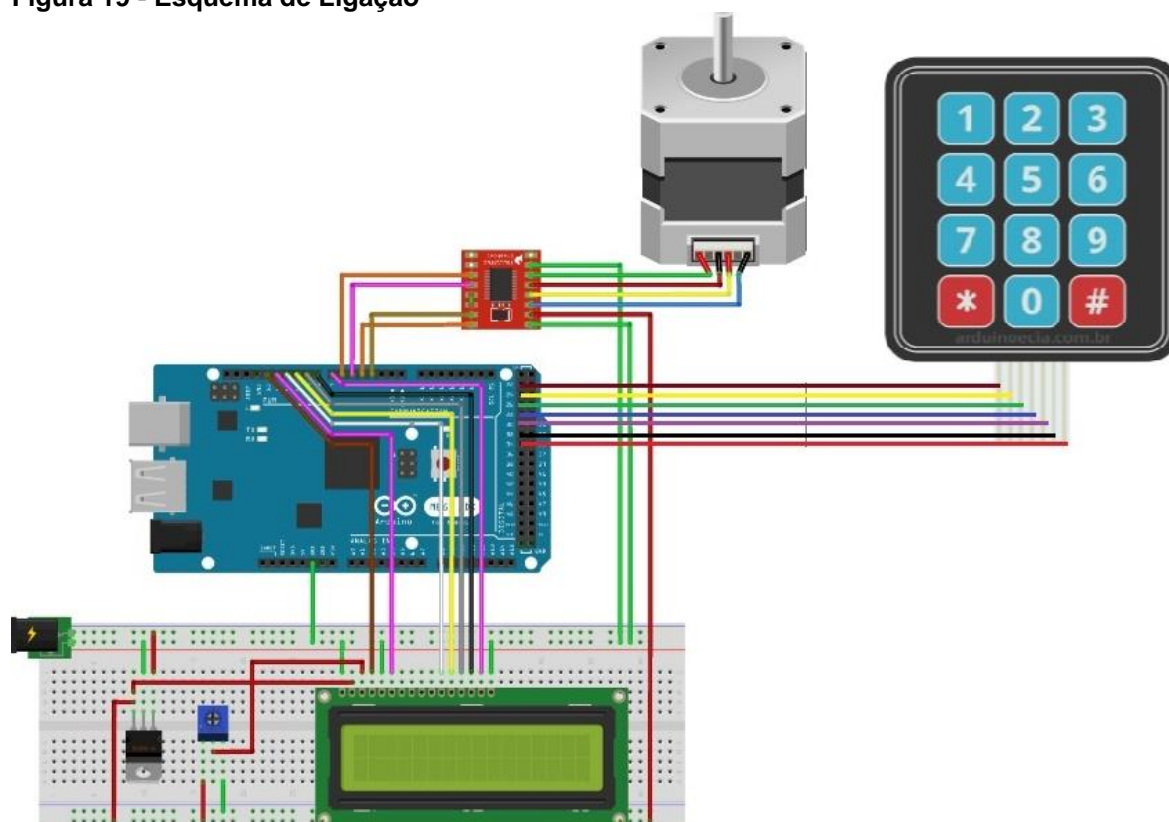
Figura 18 - Protótipo do Claviculario Eletrônico em testes



Fonte: Autoria Própria (2018).

O esquema de ligação dos componentes utilizados, pode ser visualizado na Figura 19 abaixo.

Figura 19 - Esquema de Ligação



Fonte: Autoria Própria (2018).

2.5 RESULTADOS

Com o projeto concluído foram realizados alguns testes para verificar o funcionamento do mesmo. O primeiro teste realizado foi para verificar se haveria algum erro de posicionamento nos compartimentos das chaves. Realizando os testes sem as chaves nos compartimentos obtive se com êxito o alinhamento dos compartimentos, ou seja 100% de acertos nos dois sentidos de rotação, outro teste realizado foi inserindo 05 chaves dispostas simetricamente distribuídas nos compartimentos e obtive se 93,3% de acerto no sentido anti-horário e 100% no sentido horário. Também foi realizado outro teste com 06 chaves disposta assimetricamente nos compartimentos onde obtive se 96,3% de acerto no sentido anti-horário e 100% de acerto no sentido oposto, o erro que o dispositivo apresentou foi de 0,225° fora da posição.

Outro item testado foi em relação ao digitar um número de chave não correspondente ao do claviculário, onde a mensagem “chave inexistente” aparece no display e sua resposta foi de 100% de exatidão.

Os resultados dos testes se encontram na Tabela 2.

Tabela 2 – Testes de Monitoramento de Erros de Posição

Testes de Monitoramento de Erros de Posição				
Disposição de Chaves	Sentido	Nº de testes	Nº de Erros	Nº de Chaves
Simétrica	Horário	30	0	5
	Anti-Horário	30	2	5
Assimétrica	Horário	30	0	6
	Anti-Horário	30	1	6
X	Horário	30	0	0
	Anti-Horário	30	0	0

Fonte: Autoria Própria.

Os erros de posicionamentos apresentados nos testes, foram ocasionados pela inércia do movimento de rotação, o que é corrigido pelo sensor de posicionamento inicial (encoder).

Na sequência foram realizados testes digitando senhas, os quais foram feitos inserindo senhas de um, dois, três e quatro dígitos, sendo que a senha correta é uma senha de três dígitos. O equipamento funcionou corretamente obtendo 100% de acerto, indicando as senhas incorretas e corretas, proporcionando o acesso somente

para a senha digitada corretamente. A Tabela 3 a seguir demonstra o número de testes realizados.

Tabela 3 – Testes Referentes a Senha

Testes Referentes a Senha	
Senhas	Nº de Tentativas com Senha Incorreta
1 Dígito	20
2 Dígitos	20
3 Dígitos	20
4 Dígitos	20

Fonte: Autoria Própria.

Outro resultado alcançado, foi em relação a estrutura geral do claviculário, pois obtive se um bom formato de acondicionamento das chaves, além de dispor uma estrutura arquitetônica bem agradável, e um funcionamento e operação simples, porém bem eficaz, contando com uma boa segurança no acesso as chaves.

3 CONCLUSÃO

Com base no projeto e nos resultados obtidos, conclui-se que a arquitetura do claviculário acomodou as chaves de uma forma que ocupou espaço otimizado.

Em relação aos claviculários similares encontrados no mercado, o diferencial é que este dispõe um sistema mais seguro e rápido de obter somente uma chave, já os demais possuem uma arquitetura a qual propicia o acesso a todas as chaves, por mais que obtenha senhas e trava para cada chave, fica fácil de burlar esse controle.

Devido ao seu sistema de rotação, o acesso à chave é mais rápido, pois basta digitar o número da chave desejada para que o sistema entregue automaticamente, sem a necessidade de abrir uma porta e procurar a chave em meio as outras como é o caso dos claviculários similares.

Um dos aspectos importantes a ressaltar é o custo dos equipamentos, onde o projeto descrito teve um custo de R\$ 347,81 (conforme pode ser observado no apêndice B) entre componentes eletrônicos, usinagem e montagem, porém, estes custos referem-se a um produto não finalizado para o comércio.

Uma das dificuldades encontradas quanto a execução deste projeto, foi em questão a programação em C, apesar dos autores do projeto possuírem um conhecimento obtidos durante o curso, pois, os componentes do processo devem trabalhar precisamente e em sincronismo para um bom funcionamento do claviculário o que ocasionou a não conclusão por completo do programa.

Outra questão foi a dificuldade de encontrar informações técnicas sobre o assunto, pois, se trata de um produto que não se encontra fácil no mercado e os similares não possuem muitos artigos técnicos como referência.

Este projeto propicia trabalhos futuros, como por exemplo, melhorias no desenvolvimento da programação, substituição do motor de passo por um de torque maior, melhorias nos componentes do projeto, além do desenvolvimento de acessórios e novos itens de segurança ao usuário.

Apesar das dificuldades encontradas e a escassez de tempo, a realização do trabalho foi muito significativa e produtiva, onde o aprendizado foi grande e satisfatório tanto com as pesquisas, programação, seleção de componentes, quanto com a própria elaboração do TCC que será de grande importância para o desenvolvimento profissional dos autores.

REFERÊNCIAS

ARDUINO E CIA. **Como usar o driver A4988 com motor de passo Nema 17.** 2015. Disponível em: <<https://www.arduinoecia.com.br/2015/03/driver-a4988-com-motor-de-passo-nema-17.html>>. Acesso em: 04 mai. 2018.

ARDUÍNO E CIA. **Teclado matricial membrana 4x3 com Arduíno.** 2015. Disponível em: <<https://www.arduinoecia.com.br/2015/05/teclado-matricial-membrana-4x3-arduino.html>>. Acesso em: 10 mai. 2018.

FEERICA. **Chaveiro Eletrônico Traka.** 2018. Disponível em: <<http://www.feerica.com/produtos/controlo-de-acessos/chaveiros-electronicos/chaveiro-electronico/solucao-traka/>>. Acesso em: 15 mai. 2018.

ROBOCORE TECNOLOGIA (São Paulo). **Arduíno Mega 2560 R3.** 2018. Disponível em: <<https://www.robocore.net/loja/produtos/arduino-mega-2560-r3.html>>. Acesso em: 20 mai. 2018.

BAÚ DA ELETRÔNICA (São Paulo). **Sensor de Velocidade Encoder.** 2018. Disponível em: <<http://www.baudaeletronica.com.br/sensor-de-velocidade-encoder.html>>. Acesso em: 21 mai. 2018.

ROBOCORE TECNOLOGIA (São Paulo). **Motor de Passo: NEMA 17 - 1,1 kgf.cm.** 2018. Disponível em: <https://www.robocore.net/loja/produtos/motor-de-passo-nema17-1_1kgf_cm.html>. Acesso em: 22 mai. 2018.

BAÚ DA ELETRÔNICA (São Paulo). **Sensor de Proximidade Indutivo: NPN 6~36V.** 2018. Disponível em: <<http://www.baudaeletronica.com.br/sensor-de-proximidade-indutivo-npn-6-36v.html>>. Acesso em: 22 mai. 2018.

PM COFRES (São Paulo). **Claviculários: quando as chaves valem tanto quanto o dinheiro.** 2018. Disponível em: <<http://www.pmcofres.com.br/claviculario-porta-chaves/clavicularios-quando-as-chaves-valem-tanto-quanto-o-dinheiro/>>. Acesso em: 24 mai. 2018.

PM COFRES (São Paulo). **Proteja e Organize suas Chaves com um Claviculário.** 2017. Disponível em: <<http://www.pmcofres.com.br/claviculario-porta-chaves/proteja-e-organize-suas-chaves-com-um-claviculario/>>. Acesso em: 24 mai. 2018.

PM COFRES (São Paulo). **Claviculario**: Segurança e Organização de Chaves. 2017. Disponível em: <<http://www.pmcofres.com.br/claviculario-porta-chaves/claviculario-seguranca-e-organizacao-de-chaves/>>. Acesso em: 24 mai. 2018.

PM COFRES (São Paulo). **Você Sabe Como Surgiram os Cofres?** 2016. Disponível em: <<http://www.pmcofres.com.br/seguranca/voce-sabe-como-surgiram-os-cofres/>>. Acesso em: 25 mai. 2018.

CURIO SFERA. **História do cofre**. Disponível em: <<http://www.curiosfera.com/historia-la-caja-fuerte/>>. Acesso em: 25 mai. 2018.

NEOMOTION. **Datasheet**: AK 17/ 1.10F6LN1.8. 2018. Disponível em: <<https://neomotion.com.br/motor-de-passo/>>. Acesso em: 25 jun. 2018.

LEXMANN, Robert. **Arduíno Mega 2560**. 2016. Disponível em: <<https://dspace.cvut.cz/bitstream/handle/10467/65602/F2-BP-2016-Lexmann-Robert-priloha-4-ArduinoMega2560.pdf?sequence=4&isAllowed=y>>. Acesso em: 26 jun. 2018.

APÊNDICE A – Programação Fragmentada em Linguagem C

- **SENHA**

```
// include the library code:
#include <LiquidCrystal.h>
//biblioteca responsável por capturar a tecla que foi pressionada no teclado
#include <Keypad.h>
// initialize the library with the numbers of the interface pins
LiquidCrystal lcd(13, 12, 11, 10, 9, 8);
const byte LINHAS = 4; //número de linhas do teclado
const byte COLUNAS = 3; //número de colunas do teclado
//define uma matriz com os símbolos que deseja ser lido do teclado
char SIMBOLOS[LINHAS][COLUNAS] = {
  {'1','2','3'},
  {'4','5','6'},
  {'7','8','9'},
  {'*','0','#'}
};
byte PINOS_LINHA[LINHAS] = {22, 24, 26, 28}; //pinos que indicam as linhas do teclado
byte PINOS_COLUNA[COLUNAS] = {30, 32, 34}; //pinos que indicam as colunas do teclado
//instancia de Keypad, responsável por capturar a tecla pressionada
Keypad customKeypad = Keypad( makeKeymap(SIMBOLOS), PINOS_LINHA, PINOS_COLUNA,
LINHAS, COLUNAS);
//variáveis responsáveis por armazenar as senhas
const String SENHA_ESPERADA = "123";
String SENHA_DIGITADA = "";
void setup(){
  lcd.begin(16, 2);
  inicio();
}
void loop(){
  //captura a tecla pressionada do teclado
  char customKey = customKeypad.getKey();
  //caso alguma tecla foi pressionada
  if (customKey){
    Serial.println(customKey);
    switch(customKey)
    {
      //caso alguma das teclas imprimíveis foi pressionada
      case '0':
      case '1':
      case '2':
      case '3':
      case '4':
      case '5':
      case '6':
      case '7':
```

```

case '8':
case '9':
    //concatena o novo símbolo a senha que estamos digitando
    SENHA_DIGITADA+=customKey;
    Serial.println(SENHA_DIGITADA);
    //imprime na tela o símbolo pressionado
    lcd.print(customKey);
    break;
//caso a tecla CLEAR tenha sido pressionada
case '*':
    //limpa a variável que guarda a senha que está sendo digitada
    SENHA_DIGITADA = "";
    //chama o comando para limpar a tela
    lcd.clear();
    //configura a mensagem para digitar a senha
    inicio();
    break;
//caso a tecla ENTER seja pressionada, devemos comparar as senhas
case '#':
    //limpa a tela
    lcd.clear();
    //se a senha digitada foi igual a ESPERADA
    if(SENHA_ESPERADA==SENHA_DIGITADA)
    {
        Serial.println("Senha Correta!");
        //imprime mensagem de senha correta
        lcd.print("Senha Correta!!!");
        delay(1000);
        chave();
    }
    //caso senha esteja errada
    else{
        Serial.println("Senha Incorreta!");
        //imprime mensagem de senha incorreta
        lcd.print("Senha Incorreta!");
    }
    //aguarda 2 segundos para limpar a tela novamente e esperar uma nova senha ser digitada
    delay(2000);
    lcd.clear();
    SENHA_DIGITADA = "";
    inicio();
    break;
default: break;
}
}
}
//função responsável por imprimir na tela a mensagem para digitar a senha
//é chamada toda vez q a senha foi digitada e comparada, também quando

```

```
//a tecla limpar display foi pressionada.
```

```
void inicio(){
  lcd.setCursor (5,0);
  lcd.print("UTFPR");
  lcd.setCursor (0,1);
  lcd.print("Senha:");
}
void chave(){
  lcd.clear();
  lcd.print("Laboratório:");
}
}
```

• LABORATÓRIO

```
//variáveis responsáveis por armazenar os números dos laboratórios
```

```
const String LAB01 = "1";
const String LAB02 = "2";
>>>
const String LAB20 = "20";
String LAB_DIGITADO = "";
void setup(){
  lcd.begin(16, 2);
  pinMode(MS1, OUTPUT); // Configura "MS1" como saída
  pinMode(MS2, OUTPUT); // Configura "MS2" como saída
  pinMode(dir_pin, OUTPUT); // Configura "dir_pin" como saída
  pinMode(step_pin, OUTPUT); // Configura "step_pin" como saída
  digitalWrite(MS1,HIGH); // Configura divisão de passos do motor (ver acima)
  digitalWrite(MS2, HIGH); // Configura divisão de passos do motor (ver acima)
  inicio();
  //posicao();
}
void loop(){
  //captura a tecla pressionada do teclado
  char customKey1 = customKey1pad.getKey();
  //caso alguma tecla foi pressionada
  if (customKey1){
    Serial.println(customKey1);
    switch(customKey1)
    {
      //caso alguma das teclas imprimíveis foi pressionada
      case '0':
      case '1':
      case '2':
      case '3':
      case '4':
      case '5':
```



```

case '6':
case '7':
case '8':
case '9':
    //concatena o novo símbolo a senha que estamos digitando
    LAB_DIGITADO+=customKey1;
    Serial.println(LAB_DIGITADO);
    //imprime na tela o símbolo pressionado
    lcd.print(customKey1);
    break;
//caso a tecla CLEAR tenha sido pressionada
case '*':
    //limpa a variável que guarda a senha que está sendo digitada
    LAB_DIGITADO = "";
    //chama o comando para limpar a tela
    lcd.clear();
    //configura a mensagem para digitar a senha
    inicio();
    break;
//caso a tecla ENTER seja pressionada, devemos comparar as senhas
case '#':
    //limpa a tela
    lcd.clear();
    //se o numero do laboratório digitado é igual a ESPERADA
    if(LAB01==LAB_DIGITADO)
    {
        //imprime mensagem:
        lcd.print("Laboratorio 01");
        digitalWrite(dir_pin, LOW); //seta a direção do motor
        passos=00;
        posicao(); //volta para subfunção motor de passo
    }
    if(LAB02==LAB_DIGITADO)
    {
        lcd.print("Laboratorio 02");
        digitalWrite(dir_pin, LOW);
        passos=80;
        posicao();
    }
    >>>
    lcd.print("Laboratorio 20");
    digitalWrite(dir_pin, HIGH);
    passos=80;
    posicao();
}
//caso o numero esteja errado
else{

```

```

        //imprime mensagem
        lcd.clear();
        lcd.print("Chave inexistente!");
    }
    //aguarda 2 segundos para limpar a tela
    delay(2000);
    lcd.clear();
    LAB_DIGITADO = "";
    inicio();
    break;
default: break;
    }
}
}

```

• MOTOR DE PASSO

```

/* Programa Motor de Passo
////////////////////////////////////
// MS1 MS2 //configurações possíveis:
//
// LOW LOW = Passo completo //
// HIGH LOW = Meio passo //
// LOW HIGH = Quarto de Passo //
// HIGH HIGH = Oitavo de Passo //
// //
////////////////////////////////////
*/
#define step_pin 3 // Define o pino 3 como pino dos passos
#define dir_pin 4 // Define o pino 4 como pino de direção
#define MS1 5 // Define o pino 5 como "MS1"
#define MS2 6 // Define o pino 6 como "MS2"
int direcao; // Para determinar o sentido do motor
int passos =5; // Número de passos que você deseja executar (para passos completos, 200 = 1 volta)
int volta =0;
void setup()
{
    pinMode(MS1, OUTPUT); // Configura "MS1" como saída
    pinMode(MS2, OUTPUT); // Configura "MS2" como saída
    pinMode(dir_pin, OUTPUT); // Configura "dir_pin" como saída
    pinMode(step_pin, OUTPUT); // Configura "step_pin" como saída
    digitalWrite(MS1,HIGH); // Configura divisão de passos do motor (ver acima)
    digitalWrite(MS2, HIGH); // Configura divisão de passos do motor (ver acima)
    digitalWrite(dir_pin, HIGH); // Sentido (HIGH = anti-horário / LOW = horário) - Também pode ser alterado
}
void loop(){

```

```

if(passos>=0) { // Enquanto o valor de passos for maior ou igual a zero
    digitalWrite(step_pin, HIGH); // Envia nível lógico alto para o pino de passos do motor
    delay(10); // Aguarda ms para o próximo passo
    digitalWrite(step_pin, LOW); // Envia nível lógico baixo para o pino de passos do motor
    delay(10); // Aguarda ms para o próximo passo
    passos--; // Decrementa a variável "passos"
}
else {
    if(volta==passos){// Enquanto o valor de passos for maior ou igual a zero
        digitalWrite(step_pin, HIGH); // Envia nível lógico alto para o pino de passos do motor
        delay(10); // Aguarda ms para o próximo passo
        digitalWrite(step_pin, LOW); // Envia nível lógico baixo para o pino de passos do motor
        delay(10); // Aguarda ms para o próximo passo
        volta++; // Decrementa a variável "passos"
    }
}
}
}
}

```

• TECLADO E DISPLAY LCD

```

// inclua o código da biblioteca:
#include <LiquidCrystal.h>
//biblioteca responsável por capturar a tecla que foi pressionada no teclado
#include <Keypad.h>
#define step_pin 3 // Define o pino 3 como pino dos passos
#define dir_pin 4 // Define o pino 4 como pino de direção
#define MS1 5 // Define o pino 5 como "MS1"
#define MS2 6 // Define o pino 6 como "MS2"
int passos ;
// inicializar a biblioteca com os números dos pinos da interface
LiquidCrystal lcd(13, 12, 11, 10, 9, 8);
const byte LINHAS = 4; //número de linhas do teclado
const byte COLUNAS = 3; //número de colunas do teclado
//define uma matriz com os símbolos que deseja ser lido do teclado
char SIMBOLOS[LINHAS][COLUNAS] = {
    {'1','2','3'},
    {'4','5','6'},
    {'7','8','9'},
    {'*','0','#'}
};
byte PINOS_LINHA[LINHAS] = {22, 24, 26, 28}; //pinos que indicam as linhas do teclado
byte PINOS_COLUNA[COLUNAS] = {30, 32, 34}; //pinos que indicam as colunas do teclado
//instancia de Keypad, responsável por capturar a tecla pressionada
Keypad customKey1pad = Keypad( makeKeymap(SIMBOLOS), PINOS_LINHA, PINOS_COLUNA,
LINHAS, COLUNAS);

```

APÊNDICE B - Custos Totais do Claviculário Eletrônico

Item	Descrição	Unid.	Qtde.	Preço uni.	Total
01	Motor de Passo - NEMA 17 - 1,1 kgf.cm	Pç	01	R\$ 79,00	R\$ 79,00
02	Drive Para Motor - NEMA 17	Pç	02	R\$ 16,90	R\$ 33,80
03	Acoplador Universal para Eixo de 5mm	Pç	01	R\$ 12,00	R\$ 12,00
04	Tela Display LCD 16x2 Fundo Azul para Arduino	Pç	01	R\$ 20,00	R\$ 20,00
05	Teclado Membrana Matricial 4x3 para Arduino	Pç	01	R\$ 16,99	R\$ 16,99
06	Placa de Acrílico	Pç	02	R\$ 20,00	R\$ 40,00
07	Tubo de Esgoto PVC - 40 mm	M	1,5	R\$ 3,35	R\$ 5,03
08	Tampa p/ Tubo PVC – 40 mm	Pç	20	R\$ 1,60	R\$ 32,00
09	Usinagem do Acrílico	Un	1	R\$ 15,00	R\$ 15,00
10	Placa Arduino Mega 2560	Pç	1	R\$ 40,99	R\$ 40,99
11	Motor de Passo GSP-24RW-01	Pç	1	R\$ 19,00	R\$ 19,00
12	Módulo encoder acoplador óptico Arduino	Pç	1	R\$ 14,00	R\$ 14,00
13	Sensor Indutivo	Pç	1	R\$ 20,00	R\$ 20,00
				Total	R\$ 347,81

ANEXO A - Datasheet - Clavulário Eletrónico (Feerica Traka)

www.feerica.com

feerica®

Chaveiro Electrónico



A solução da Traka é líder mundial, especialista em soluções de gestão de chaves e equipamentos. Permitindo aceder a edifícios, áreas de alta segurança, equipamentos, maquinaria, veículos, etc ..., a gestão de chaves dentro de uma empresa é fundamental e importante permitindo um maior controlo, monitorização e registo de qualquer acesso físico. Numa solução convencional, qualquer tentativa de gerir chaves, consiste normalmente em registar em papel alguns detalhes da chave que foi dispensada ou devolvida, confirmado por uma assinatura muitas vezes ilegível – uma solução que se torna ineficiente, insegura, e em que o rastreamento dessa chave é quase uma tarefa impossível.

Torne as suas chaves convencionais em chaves inteligentes

Um dos dispositivos mais importantes no chaveiro electrónico da Traka é o iFob. Este dispositivo em forma de bala, contém um chip com um ID único. Uma vez ligado a um iFob, uma chave ou um conjunto de chaves são efectivamente etiquetadas electronicamente. As chaves são permanentemente ligadas a um iFob com um selo em aço. Cada iFob é então associado a uma posição específica no chaveiro electrónico o qual fica preso até ser libertado por um utilizador autorizado.



Restringir o acesso a utilizador autorizados

Uma vez configurado o perfil apropriado do utilizador e detalhes das chaves no sistema, o chaveiro electrónico da Traka assegura que somente utilizadores autorizados tenham acesso ao chaveiro e às suas respectivas chaves. O sistema automaticamente regista quando uma chave é retirada e por quem, numa base de dados central. Esta informação é disponibilizada via *display* do chaveiro ou através do PC administrador.

O que distingue a solução Traka

- O utilizador é configurado com um determinado perfil, permitindo só retirar as chaves a que tem acesso
- As chaves podem ser configuradas com um tempo de devolução no chaveiro e gerar um alarme caso esse tempo seja ultrapassado
- O sistema permite configurar um horário de acesso ao chaveiro e às chaves
- Dependência entre chaves: uma determinada chave só pode ser retirada se a sua chave dependente estiver presente no chaveiro
- Configuração de um data de expiração, para que as chaves fiquem automaticamente sem acesso após essa data
- Determinadas informações podem ser registadas quando as chaves são devolvidas: quilometragem, falhas ou defeitos, etc... Chaves com falhas registadas podem ser automaticamente trancadas a qualquer pessoa e só serem libertadas perante o administrador.
- Notificações via e-mail podem ser automaticamente enviadas aos supervisores de sistema, alertando da ocorrência de um determinado evento: Porta do chaveiro foi deixada aberta, remoção de uma determinada chave do chaveiro, expiração do tempo de devolução de uma chave, etc...

Para obter mais informações sobre Guarda de Chaves, por favor contacte-nos.

criatividade tecnologia inovação

Feerica, SA
Av. Dr. Francisco Sá Carneiro
NEM Pav. 1 2640-486 Mafra

Tel +351 261 812 203
callcenter@feerica.com
Portugal

feerica®

Chaveiro Electrónico

Arquitectura modular para qualquer tipo de empresa

Os chaveiros electrónicos da Traka estão disponíveis em três séries permitindo uma capacidade de 10 a 540 chaves.

- A Série M permite uma gestão de 10 a 20 chaves
- A série S permite uma gestão de 10 a 60 chaves
- A série L permite uma gestão de 10 a 180 chaves

Múltiplos sistemas podem ser interligados através de uma rede informática permitindo a gestão de um número ilimitado de chaves, administrados e centralizado pelo software Traka32.

Identificação do Utilizador

O chaveiro da Traka suporta qualquer tipo de dispositivo de controlo de acessos permitindo a identificação do utilizador perante o chaveiro. Nestes dispositivos estão incluídos os leitores de proximidade RFID, de banda magnética, de código de barras e tecnologia biométrica tais como impressão digital.

Software Traka32

O chaveiro Electrónico é gerido pelo software Traka32 permitindo uma base de dados em Microsoft Access ou SQL. Todo o sistema pode comunicar com o software Traka32 utilizando diferentes opções: Ethernet Wireless, GPRS, RS485, RS232 ou modem. O software Traka32 de fácil utilização permite emitir relatórios com informação detalhada sobre todo o rastreamento da chave. Permite ainda:

- Adicionar / Apagar utilizadores, iFobs, chaves
- Configuração do perfil de utilizador e respectivas permissões de acesso
- Monitorizar on-line do sistema, informando quais as chaves que estão ou não presentes nos chaveiros
- Notificação automática via e-mail na ocorrência de eventos especificados
- Libertação de uma chave em caso de emergência
- Configuração do perfil de utilizador de software
- Numa configuração em rede, a base de dados guardada no servidor pode ser partilhada por multi-postos, permitindo a gestão e controlo do chaveiro da respectiva região
- Determinação do número de vezes que uma determinada chave foi utilizada e exactamente por quanto tempo



As especificações podem sofrer alterações

Para obter mais informações sobre Guarda de Chaves, por favor contacte-nos.

criatividade tecnologia inovação

Feerica, SA
Av. Dr. Francisco Sá Carneiro
NEM Pav. 1 2640-486 Mafra

Tel +351 261 812 203
callcenter@feerica.com
Portugal

ANEXO B - Datasheet - Motores de Passo



P/N: AK17/1.10F6LN1.8
 P/N Antigo: AK39H/12-1.8

neoyama

MOTOR DE PASSO

1. Marca do Produto

AKIYAMA MOTORS

2. Especificações Técnicas

NEMA	Conexão	Holding torque (kgf.cm)	Corrente de pico (mA/Fase)	Tensão (V/Fase)	Resistência (Ω/Fase)	Indutância (mH/Fase)
17	Bipolar - Série	1,1	70	16,8	140	148
	Unipolar	0,77	100	12	70	37

*Corrente de pico: $I_p = I_{RMS} \times \sqrt{2}$

Item	Especificação
Ângulo do passo	1,8°
N° de passos	200
Enrolamento	Espiras bifilares
Temperatura máx. de operação	80°C
Temperatura ambiente	-10°C ~ 50°C
Resistência de isolamento	100 Ω / 500 Vdc
Rigidez dielétrica	500 Vac / 1 min.
Classe de isolamento	B
Folga máx. radial	0,03 mm / Carga = 400g
Folga máx. axial	0,03 mm / Carga = 500g
Detent torque	0,06 gf.cm
Inércia do rotor	48 g.cm ²
Quantidade de fios	6
Peso	0,22 Kg
Sugestão de driver compatível	AKDMP5-1.7A



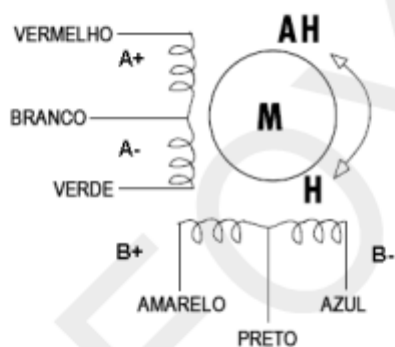
3. Sequência de Fases

	Passo	A+	B-	A-	B+
Sentido: horário	0	+	+		
	1		+	+	
	2			+	+
	3	+			+

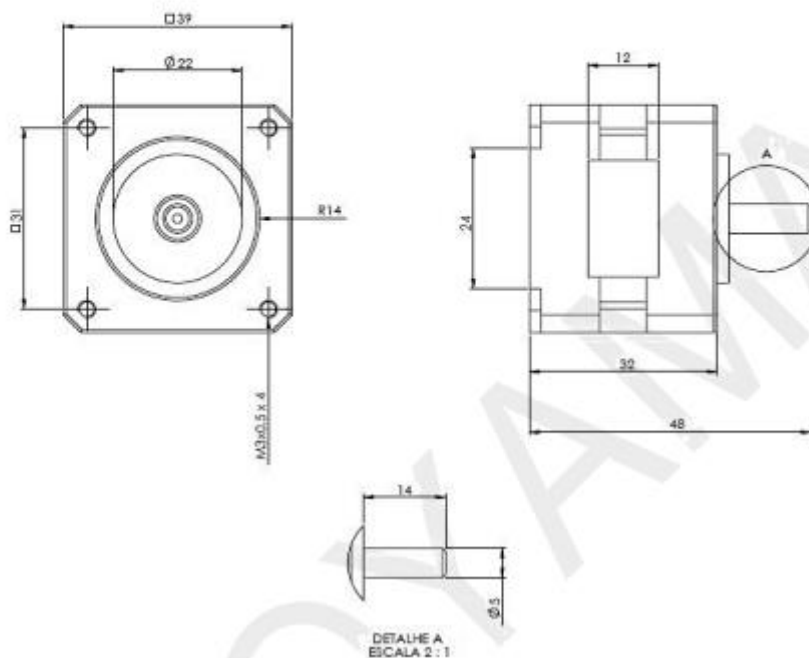
Sentido: anti-horário

Iniciando do Passo 0 - Sentido horário
 Iniciando do Passo 3 - Sentido anti-horário

4. Esquema de Ligação



5. Desenho Técnico 2D



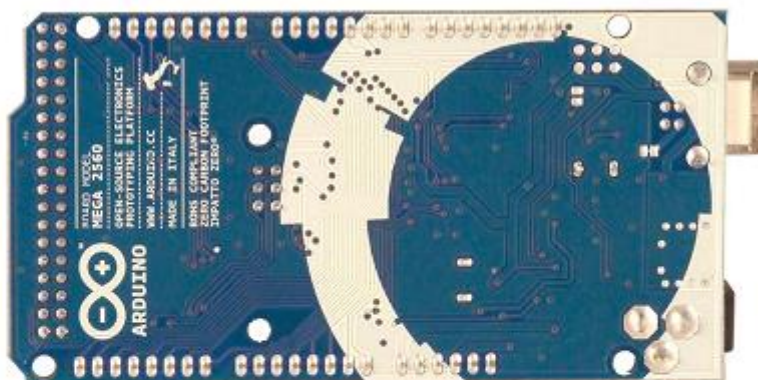
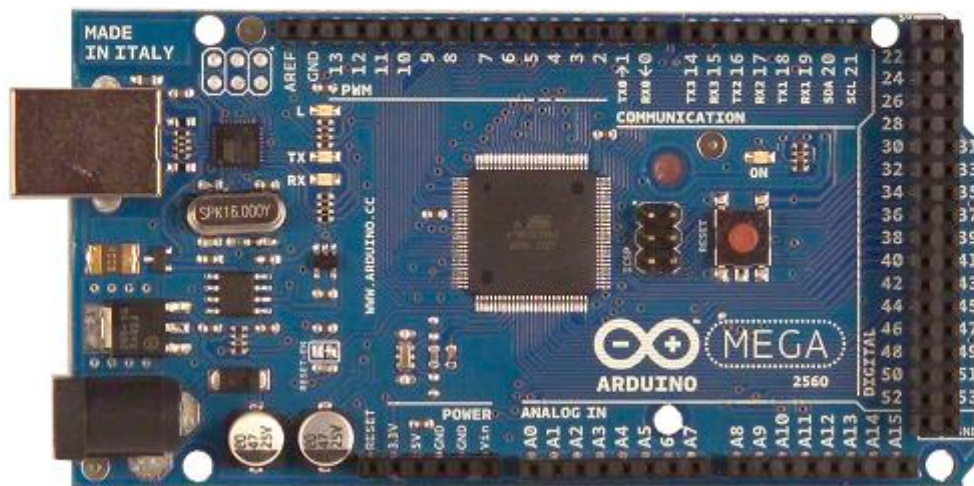
Unidade: mm

Disponibilizamos os Desenhos Técnicos 2D e 3D no site www.neoyama.com.br



ANEXO C - Arduino Mega 2560

Arduino Mega 2560



Overview

The Arduino Mega 2560 is a microcontroller board based on the ATmega2560 ([datasheet](#)). It has 54 digital input/output pins (of which 14 can be used as PWM outputs), 16 analog inputs, 4 UARTs (hardware serial ports), a 16 MHz crystal oscillator, a USB connection, a power jack, an ICSP header, and a reset button. It contains everything needed to support the microcontroller; simply connect it to a computer with a USB cable or power it with a AC-to-DC adapter or battery to get started. The Mega is compatible with most shields designed for the Arduino Duemilanove or Diecimila.

Schematic & Reference Design

EAGLE files: [arduino-mega2560-reference-design.zip](#)

Schematic: [arduino-mega2560-schematic.pdf](#)

Summary

Microcontroller	ATmega2560
Operating Voltage	5V
Input Voltage (recommended)	7-12V
Input Voltage (limits)	6-20V
Digital I/O Pins	54 (of which 14 provide PWM output)
Analog Input Pins	16
DC Current per I/O Pin	40 mA
DC Current for 3.3V Pin	50 mA
Flash Memory	256 KB of which 8 KB used by bootloader
SRAM	8 KB
EEPROM	4 KB
Clock Speed	16 MHz

Power

The Arduino Mega can be powered via the USB connection or with an external power supply. The power source is selected automatically.

External (non-USB) power can come either from an AC-to-DC adapter (wall-wart) or battery. The adapter can be connected by plugging a 2.1mm center-positive plug into the board's power jack. Leads from a battery can be inserted in the Gnd and Vin pin headers of the POWER connector.

The board can operate on an external supply of 6 to 20 volts. If supplied with less than 7V, however, the 5V pin may supply less than five volts and the board may be unstable. If using more than 12V, the voltage regulator may overheat and damage the board. The recommended range is 7 to 12 volts.

The Mega2560 differs from all preceding boards in that it does not use the FTDI USB-to-serial driver chip. Instead, it features the Atmega8U2 programmed as a USB-to-serial converter.

The power pins are as follows:

- ✦ **VIN.** The input voltage to the Arduino board when it's using an external power source (as opposed to 5 volts from the USB connection or other regulated power source). You can supply voltage through this pin, or, if supplying voltage via the power jack, access it through this pin.
- ✦ **5V.** The regulated power supply used to power the microcontroller and other components on the board. This can come either from VIN via an on-board regulator, or be supplied by USB or another regulated 5V supply.
- ✦ **3V3.** A 3.3 volt supply generated by the on-board regulator. Maximum current draw is 50 mA.
- ✦ **GND.** Ground pins.

Memory

The ATmega2560 has 256 KB of flash memory for storing code (of which 8 KB is used for the bootloader), 8 KB of SRAM and 4 KB of EEPROM (which can be read and written with the [EEPROM library](#)).

Input and Output

Each of the 54 digital pins on the Mega can be used as an input or output, using `pinMode()`, `digitalWrite()`, and `digitalRead()` functions. They operate at 5 volts. Each pin can provide or receive a maximum of 40 mA and has an internal pull-up resistor (disconnected by default) of 20-50 kOhms. In addition, some pins have specialized functions:

- ✦ **Serial: 0 (RX) and 1 (TX); Serial 1: 19 (RX) and 18 (TX); Serial 2: 17 (RX) and 16 (TX); Serial 3: 15 (RX) and 14 (TX).** Used to receive (RX) and transmit (TX) TTL serial data. Pins 0 and 1 are also connected to the corresponding pins of the ATmega8U2 USB-to-TTL Serial chip.
- ✦ **External Interrupts: 2 (interrupt 0), 3 (interrupt 1), 18 (interrupt 5), 19 (interrupt 4), 20 (interrupt 3), and 21 (interrupt 2).** These pins can be configured to trigger an interrupt on a low value, a rising or falling edge, or a change in value. See the `attachInterrupt()` function for details.
- ✦ **PWM: 0 to 13.** Provide 8-bit PWM output with the `analogWrite()` function.
- ✦ **SPI: 50 (MISO), 51 (MOSI), 52 (SCK), 53 (SS).** These pins support SPI communication using the [SPI library](#). The SPI pins are also broken out on the ICSP header, which is physically compatible with the Uno, Duemilanove and Diecimila.
- ✦ **LED: 13.** There is a built-in LED connected to digital pin 13. When the pin is HIGH value, the LED is on, when the pin is LOW, it's off.
- ✦ **I²C: 20 (SDA) and 21 (SCL).** Support I²C (TWI) communication using the [Wire library](#) (documentation on the Wiring website). Note that these pins are not in the same location as the I²C pins on the Duemilanove or Diecimila.

The Mega2560 has 16 analog inputs, each of which provide 10 bits of resolution (i.e. 1024 different values). By default they measure from ground to 5 volts, though is it possible to change the upper end of their range using the AREF pin and `analogReference()` function.

There are a couple of other pins on the board:

- ✦ **AREF**. Reference voltage for the analog inputs. Used with `analogReference()`.
- ✦ **Reset**. Bring this line LOW to reset the microcontroller. Typically used to add a reset button to shields which block the one on the board.

Communication

The Arduino Mega2560 has a number of facilities for communicating with a computer, another Arduino, or other microcontrollers. The ATmega2560 provides four hardware UARTs for TTL (5V) serial communication. An ATmega8U2 on the board channels one of these over USB and provides a virtual com port to software on the computer (Windows machines will need a .inf file, but OSX and Linux machines will recognize the board as a COM port automatically. The Arduino software includes a serial monitor which allows simple textual data to be sent to and from the board. The RX and TX LEDs on the board will flash when data is being transmitted via the ATmega8U2 chip and USB connection to the computer (but not for serial communication on pins 0 and 1).

A [SoftwareSerial library](#) allows for serial communication on any of the Mega2560's digital pins.

The ATmega2560 also supports I2C (TWI) and SPI communication. The Arduino software includes a Wire library to simplify use of the I2C bus; see the [documentation on the Wiring website](#) for details. For SPI communication, use the [SPI library](#).

Programming

The Arduino Mega can be programmed with the Arduino software ([download](#)). For details, see the [reference](#) and [tutorials](#).

The ATmega2560 on the Arduino Mega comes preburned with a [bootloader](#) that allows you to upload new code to it without the use of an external hardware programmer. It communicates using the original STK500 protocol ([reference](#), [C header files](#)).

You can also bypass the bootloader and program the microcontroller through the ICSP (In-Circuit Serial Programming) header; see [these instructions](#) for details.

The ATmega8U2 firmware source code is available [in the Arduino repository](#). The ATmega8U2 is loaded with a DFU bootloader, which can be activated by connecting the solder jumper on the back of the board (near the map of Italy) and then resetting the 8U2. You can then use [Atmel's FLIP software](#) (Windows) or the [DFU programmer](#) (Mac OS X and Linux) to load a new firmware. Or you can use the ISP header with an external programmer (overwriting the DFU bootloader). See [this user-contributed tutorial](#) for more information.

Automatic (Software) Reset

Rather than requiring a physical press of the reset button before an upload, the Arduino Mega2560 is designed in a way that allows it to be reset by software running on a connected computer. One of the hardware flow control lines (DTR) of the ATmega8U2 is connected to the reset line of the ATmega2560 via a 100 nanofarad capacitor. When this line is asserted (taken low), the reset line drops long enough to reset the chip. The Arduino software uses this capability to allow you to upload code by simply pressing the upload button in the Arduino environment. This means that the bootloader can

have a shorter timeout, as the lowering of DTR can be well-coordinated with the start of the upload.

This setup has other implications. When the Mega2560 is connected to either a computer running Mac OS X or Linux, it resets each time a connection is made to it from software (via USB). For the following half-second or so, the bootloader is running on the Mega2560. While it is programmed to ignore malformed data (i.e. anything besides an upload of new code), it will intercept the first few bytes of data sent to the board after a connection is opened. If a sketch running on the board receives one-time configuration or other data when it first starts, make sure that the software with which it communicates waits a second after opening the connection and before sending this data.

The Mega2560 contains a trace that can be cut to disable the auto-reset. The pads on either side of the trace can be soldered together to re-enable it. It's labeled "RESET-EN". You may also be able to disable the auto-reset by connecting a 110 ohm resistor from 5V to the reset line; see [this forum thread](#) for details.

USB Overcurrent Protection

The Arduino Mega2560 has a resettable polyfuse that protects your computer's USB ports from shorts and overcurrent. Although most computers provide their own internal protection, the fuse provides an extra layer of protection. If more than 500 mA is applied to the USB port, the fuse will automatically break the connection until the short or overload is removed.

Physical Characteristics and Shield Compatibility

The maximum length and width of the Mega2560 PCB are 4 and 2.1 inches respectively, with the USB connector and power jack extending beyond the former dimension. Three screw holes allow the board to be attached to a surface or case. Note that the distance between digital pins 7 and 8 is 160 mil (0.16"), not an even multiple of the 100 mil spacing of the other pins.

The Mega2560 is designed to be compatible with most shields designed for the Uno, Diecimila or Duemilanove. Digital pins 0 to 13 (and the adjacent AREF and GND pins), analog inputs 0 to 5, the power header, and ICSP header are all in equivalent locations. Further the main UART (serial port) is located on the same pins (0 and 1), as are external interrupts 0 and 1 (pins 2 and 3 respectively). SPI is available through the ICSP header on both the Mega2560 and Duemilanove / Diecimila. *Please note that I²C is not located on the same pins on the Mega (20 and 21) as the Duemilanove / Diecimila (analog inputs 4 and 5).*

