

**ALIMENTADOR DE PET’s AUTOMÁTICO**

Aluno: Raphael Ribeiro Faria RA: 104120

Professor: Sérgio Ronaldo Barros

Disciplina: Sistemas Embarcados

**1.Descrição do projeto**

**1.1.Descrição**

Animais de estimação já tem se tornado, há bastante tempo, muito frequentes no cotidiano da população do mundo inteiro, estando presente na vida de diversas famílias em diferentes lugares, sendo em grande parte das situações, fazendo parte ativamente das mesmas. Com isso, investimentos nessa área de PETs têm fomentado diversos ramos da tecnologia, que buscam tornar o cuidado com os adorados animais cada vez mais fácil e acessível para os seus donos. No mercado já existem vários modelos de alimentadores para animais de estimação, porém, além de seus elevados preços, a grande maioria controla apenas o recipiente de água, esvaziando o mesmo a cada ciclo, e não possibilitam ao cliente determinar os horários de alimentação, e sim apenas intervalos de tempo.

**1.2.Objetivo**

Visando resolver os problemas citados, o presente projeto visa desenvolver um protótipo de alimentador automático e inteligente para animais de estimação, que cuja programação possibilita liberar alimento em períodos agendados pelo usuário e que possua autonomia para dar comida ao cão ou gato por vários dias. O trabalho será feito utilizando como base o microcontrolador Arduino Mega 2560, que, com a implementação desenvolvida, juntamente com os demais componentes, que serão especificados em breve, serão responsáveis por funções diversas. O Arduino será programado principalmente para o controle das portas que controlam o fluxo e a quantidade de alimento e água, movimentadas com motores de passo, nos horários agendados pelo consumidor, exibidos no display LCD como uma forma de interação com o usuário.

**2.Componentes**

Os componentes utilizados para a construção do protótipo foram:

* 1 Microcontrolador Arduino Mega 2560;
* 2 Motores de passo 28BYJ-48;
* 2 *Drivers* ULN2003 ;
* 1 Display LCD 16x2;
* 1 Potenciômetro de 10k;
* 2 botões momentâneos;
* 2 resistores de 10k;
* 1 *protoboard;*
* Jumpers MM e MF;
* 2 canos de PVC;
* 2 tábuas de madeira;
* 1 torneira de filtro;
* 1 disco de plástico;
* 2 garrafas PET;
* Barbante, fita adesiva e abraçadeiras em geral.

**3.Funcionamento**

**3.1.Software**

O software responsável pelo funcionamento do projeto funciona da seguinte forma: primeiro, é chamada a função *inicializar*, que funciona apenas como uma apresentação do programa, reproduzindo uma introdução animada no display de LCD com as palavras “Alimentador para animais”. Após isso, é chamada a função *Hora\_Atual,* que é responsável por definir a hora exata do momento e armazená-la em uma variável, para que, a partir da hora dada, o programa possa rodar o seu relógio interno. É importante ressaltar que existem dois botões para o usuário, um para e mudar as suas opções, o chamado botão *MUDAR,* e um segundo botão que será apertado quando o usuário quiser selecionar a opção que aparece no LCD, o botão *SELECIONAR.* Dentro da função *Hora\_Atual* existe outra função que é a *Set\_Hora\_Atual,* que é chamada indefinidamente mudando a hora a medida que o usuário aperta o botão de *MUDAR,* e quando for pressionado o botão de *SELECIONAR,*  fará com que a função não seja mais chamada, e o valor da hora escolhida seja salvo em um vetor de três posições pela função maior *Hora\_Atual*, que armazena hora na posição 0. Para os minutos foi feita uma função *Set\_Minuto\_Atual*  que também é chamada pela função *Hora\_Atual* após a função que define as horas*,* eque funciona da mesma forma que a outra função, armazenando os minutos na posição 1 do vetor criado. Por convenção, a posição 2 do vetor, referente aos segundos atuais, sempre começa com 0.

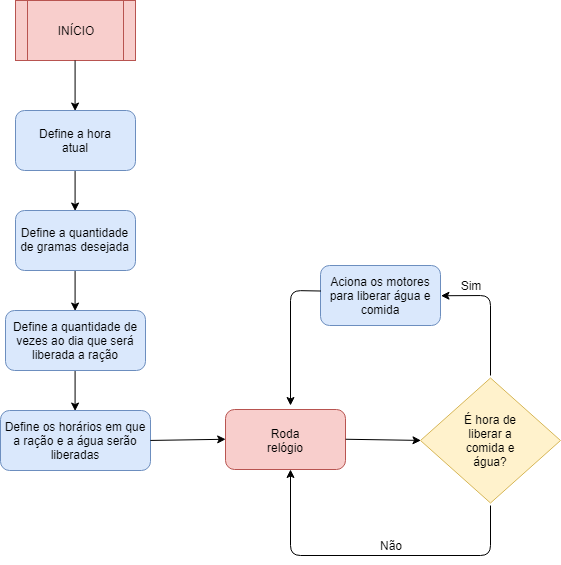
Após definida a hora atual, são definidas a quantidade de comida através da função *Comida,* e a quantidade de vezes ao dia que será liberado a comida com a função *numVezes.* Essas funções funcionam analogamente à função de escolher o horário atual, com uma função *SetComida* e *SetVezes*, que novamente serão executadas mudando o valor no display LCD até que o usuário aperte o botão *SELECIONAR* e os dados sejam salvos nas variáveis *qtd* e *nVezes,* respectivamente.

Após definidos todos esses dados descritos anteriormente, é criada uma matriz de tamanho *nVezes* colunas e 2 linhas, que irão armazenar todas as horas que o usuário definir, ou seja, uma matriz *hora[nVezes][2].* Para selecionar essas horas, é chamada uma função *Hora,* e dentro dela uma função *SetHora* e *SetMinuto,* que funcionam exatamente iguais às funções *Hora\_Atual, Set\_Hora\_Atual* e *Set\_Minuto\_Atual.* A única diferença é que essa função será executada *nVezes* vezes, já que aqui não apenas um horário será escolhido, e sim uma quantidade de horários igual à quantidade de vezes que o usuário quer liberar comida e água. Por exemplo:

* O usuário escolhe dar comida e água 3 vezes ao dia;
* É criado uma matriz *hora*[3][2];
* A hora e os minutos da hora 1 são armazenados em *hora*[0][0] e *hora*[0][1], respectivamente.
* A hora e os minutos da hora 2 são armazenados em *hora*[1][0] e *hora*[1][1], respectivamente.
* A hora e os minutos da hora 3 são armazenados em *hora*[2][0] e *hora*[2][1], respectivamente.

Depois de definidos todos os padrões para o funcionamento do programa, lembrando que todas essas etapas estão contidas em *void setup(),* já que são executadas apenas uma vez, é hora de se estruturar a função *void loop(),* que será repetida continuamente,

Nessa função, com o uso da função já incluída na IDE *millis(),* é feita uma lógica para que, a cada 1 segundo, a hora atual definida pela função *Hora\_Atual* seja incrementada na posição 3 do vetor, e a cada 60 segundos, a posição 2 seja incrementada, e a cada 60 minutos, a posição 1 seja incrementada, fazendo assim, um relógio em tempo real. Ao mesmo tempo, na mesma iteração, é feita uma comparação. Caso a hora atual seja igual a uma das horas definidas pelo usuário para a alimentação, é imprida no LCD a frase “Oba! É hora de comer”, e é chamada a função *MotorComida,* que aciona o motor e libera a comida na quantidade escolhida, e a função *MotorAgua,* que aciona o motor que abre a torneira de filtro e libera a água para o recipiente. Quando a hora atual armazenada atinge 23:59:59, o relógio é zerado e se começa a contagem novamente. Quando não é hora de se liberar a comida, o relógio é constantemente mostrado no LCD para ilustrar o funcionamento do projeto. O fluxograma abaixo ilustra como ocorre o funcionamento do *software* desenvolvido:

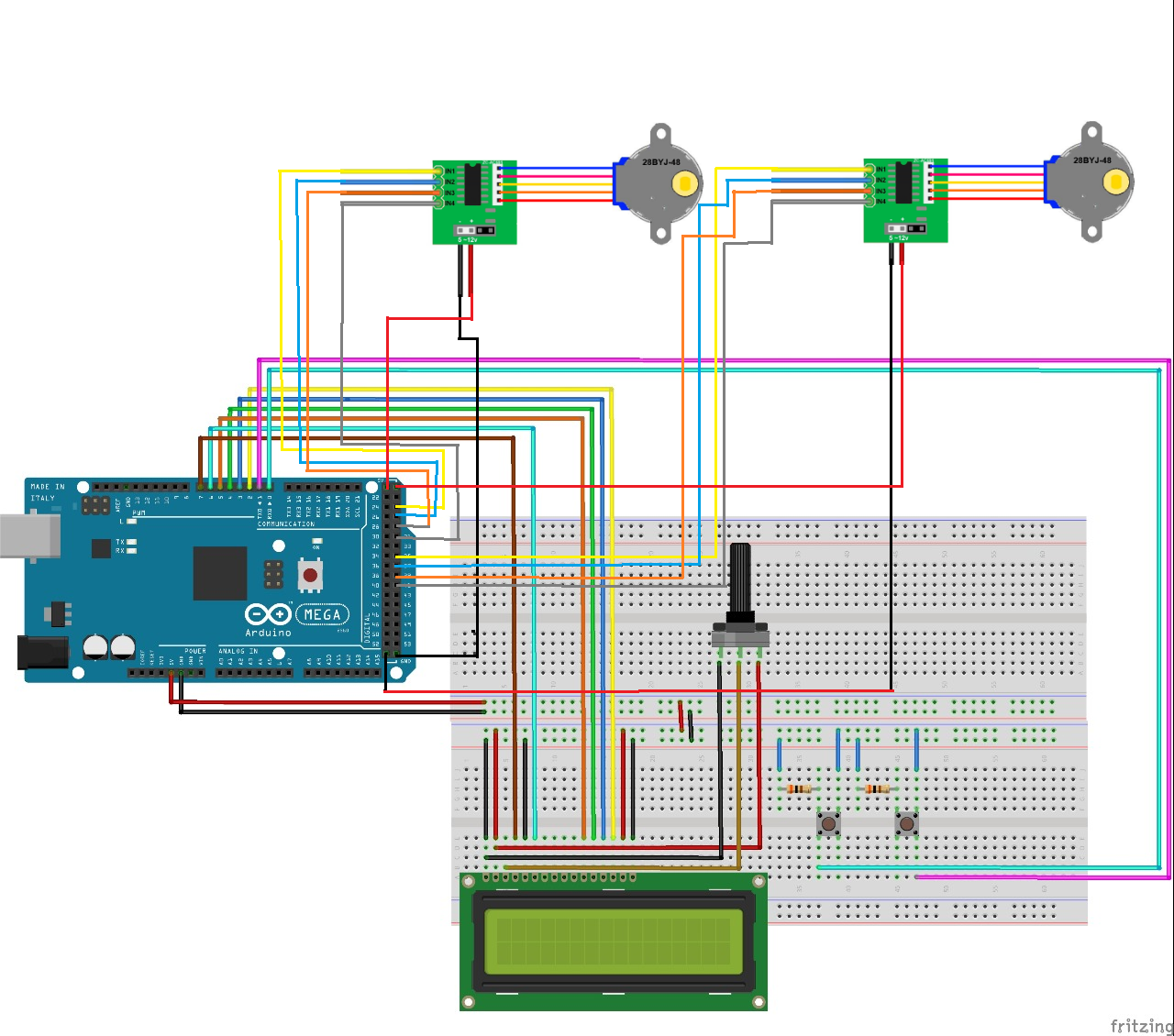
Figura 1: fluxograma do *software.* 

**3.2.Hardware**

Para a parte física do projeto, ou seja, o *hardware*, foi desenvolvido um circuito na *protoboard* composto pelo *display* LCD, que será responsável pela interação com o usuário, juntamente com o potenciômetro para o controle de sua intensidade de luz, e dois botões momentâneos com seus respectivos resistores para o seu perfeito funcionamento, cujas funções, como já descritas, são de mudar e selecionar os dados desejados pelo usuário. Como o Arduino não possui potência para controlar os motores de passo, é necessário o uso dos drivers citados anteriormente, que são compostos basicamente de um conjunto de transistores, e que serão responsáveis por fazer essa ligação entre o arduino e os motores de passo, recebendo as entradas das portas selecionadas do microcontrolador e passando os respectivos comandos para que os motores atuem de acordo com a vontade do usuário. Vale ressaltar que os drivers e os motores não utilizam a *protoboard.*

O circuito esquemático do hardware do projeto é ilustrado a seguir:

Figura 2: Circuito fo projeto



**4.Estrutura**

Para que o teste do projeto fosse realizado com mais êxito, foi desenvolvido um protótipo do alimentador de animais, feitos com canos de PVC e tábuas de madeira, como ilustrado a seguir:

Figura 3: Estrutura do projeto



O dois motores de passo que serão controlados pelo arduino são ilustrados em vermelho na figura. O motor de passo 1, quando solicitado, gira o disco, que contém uma parte cortada, e libera a ração presente no reservatório. O motor de passo 2, que se encontra na parte de trás da madeira, quando solicitado, gira e puxa o barbante que está amarrado à torneira para trás, e consequentemente, libera a água.