

CAIXA DE AREIA AUTOLIMPANTE PARA GATOS

Idealização:

A ideia da caixa de areia autolimpante para gatos surgiu de uma necessidade real de um gato chamado Guga. Este PET, tinha uma lesão na coluna que dificultava a evacuação.

A ingestão de água em quantidade era muito necessária, como os gatos gostam de água corrente, uma fonte solucionou este problema, algo facilmente encontrado nos PETSHOPS.

Surgiu um novo desafio, os gatos são animais extremamente higiênicos e não gostam de utilizar “o banheiro”, uma caixa de areia, suja. O Guga vivia com sua companheira Bebel, duas caixas de areia pareciam ser suficientes para um dia inteiro em que seus tutores estavam fora de casa. O que ocorreu na prática foi que a gata utilizava o banheiro primeiro e ele urinava na outra caixa e a partir daí não ia novamente ao banheiro, segurando as fezes e por vezes fazendo até sobre o sofá, cama, ou em qualquer lugar onde estivesse deitado. Isso ocorria quando ele não conseguia mais conter a sua vontade, causando dor e desconforto ao animal. Algumas vezes chegou a ser necessária internação de emergência para retirada das fezes.

Independente do problema relatado, os donos de gatos devem saber que a caixa limpa é uma necessidade para os gatos, mesmo quando estamos em casa, não é comum que logo ao ser utilizada o dono já efetue a limpeza da caixa, muitas vezes acaba ficando por horas inclusive causando um mau cheiro.

Dito isso, assim como a fonte de água, fui buscar alternativas de caixas de areia automatizadas, mas o que encontrei foi raros modelos importados, muito sofisticados, caros, e que além de tudo não eram funcionais.

Funcionalidade do projeto:

O gato vai até a caixa de areia, o sensoriamento detecta a presença do PET;

Após a saída do PET da caixa de areia, um timer conta um tempo de aproximadamente dois minutos sem atuar nenhum mecanismo;

Inicia o processo de varredura da areia, retirando os resíduos sólidos para um recipiente abaixo da caixa, vale lembrar que a urina do gato em contato com areias específicas, formam torrões sólidos que também são varridos assim como as fezes;

O mecanismo retorna à posição anterior, aguardando a próxima utilização.

Desafios:

Sem dúvidas o maior desafio é a elaboração da parte mecânica que seja funcional, resistente e ao mesmo tempo simples e barata;

A vassoura deverá ter as dimensões corretas para que transporte os sólidos, preservando a maior quantidade de areia possível na caixa.

Outro grande desafio será unir a parte eletrônica a parte mecânica de forma que funcione perfeitamente o conjunto.

Benefícios:

Redução do odor proveniente do banheiro dos gatos;

Tranquilidade durante o dia dos donos do PET;

Conforto para os bichinhos que não podem falar o que sentem.

Recursos:

Será necessário para este protótipo uma caixa de areia;

Mecanismos de varredura (motor de passos, cremalheira e engrenagens);

Vassoura feita em impressora 3D;

Microcontrolador e sensores para presença.

Desenvolvimento:

Como este projeto é uma ideia única e exclusiva, realizarei todas etapas de forma individual, sem equipe, com auxílio é claro dos professores e colegas do IFSC de diversas áreas;

O tempo não será um fator primordial, a celeridade se dará de acordo com o andamento do projeto;

A intenção é criar algo novo, um produto engenhoso, útil e sem antecedentes no mercado.

Custos estimados:

Como trata-se de um protótipo, a caixa será de madeira, o mecanismo vou buscar conjuntos comerciais prontos, o sensor de peso para detecção da presença é algo relativamente barato (balanças comerciais) e o microcontrolador (a definir) tenho acesso na instituição.

Para o desenvolvimento real do projeto, a intenção é que seja um produto com custo de produção de até no máximo R\$ 250,00.

Fluxograma de Hardware:

1. Organizar os requisitos necessários;

2. Adquirir os componentes de hardware necessários:

Caixa de madeira;

Mecanismo;

Microcontrolador;

Sensores;

Impressão 3D da vassoura.

3. Montar os componentes de hardware;

4. Testar o hardware sem o software;

5. Documentar.

Fluxograma de Software:

1. Organizar os requisitos necessários;

2. Configurar o sistema operacional (a definir);

3. Desenvolver os blocos necessários para a aplicação;

4. Testar e aprimorar a funcionalidade dos blocos sem os sensores;

5. Implementar os recursos de sensores;

6. Documentar.

Possíveis aprimoramentos:

A ideia da utilização de um sensor de carga, peso, para a presença do PET, surge com a ideia de tirar a tara logo após o preenchimento com a areia, no intuito de que seja possível a identificação de um nível baixo de areia;

Acrescentar uma comunicação ao usuário para envio de mensagem de nível baixo de areia e/ou mal funcionamento do sistema.

Conclusão:

A ideia do projeto não é uma caixa autolimpante onde possa deixar o gato por dias sem a presença do tutor, mas sim para o uso diário facilitando a limpeza, menos interações necessárias do dono do PET, gerando conforto ao animal e seu dono, menos cheiro no ambiente e a tranquilidade mesmo sem estar presente.

Fluxograma de Hardware:

1. Organizar os requisitos necessários;
2. Adquirir os componentes de hardware:
 - Caixa de madeira;
 - Mecanismo;
 - Microcontrolador;
 - Sensores;
 - Impressão 3D da vassoura.
3. Montar os componentes de hardware;
4. Testar o hardware sem o software;
5. Documentar

Já com o projeto idealizado, comecei a aquisição dos materiais necessários.

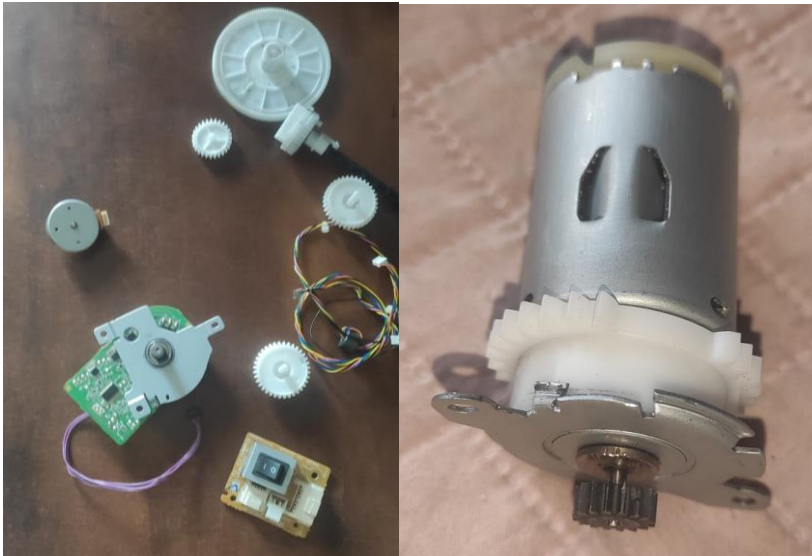
Primeiramente, adquirei uma sucata de impressora, copiadora com scanner.



Desmontei toda a estrutura para utilização do sistema mecânico do guia da vassoura.

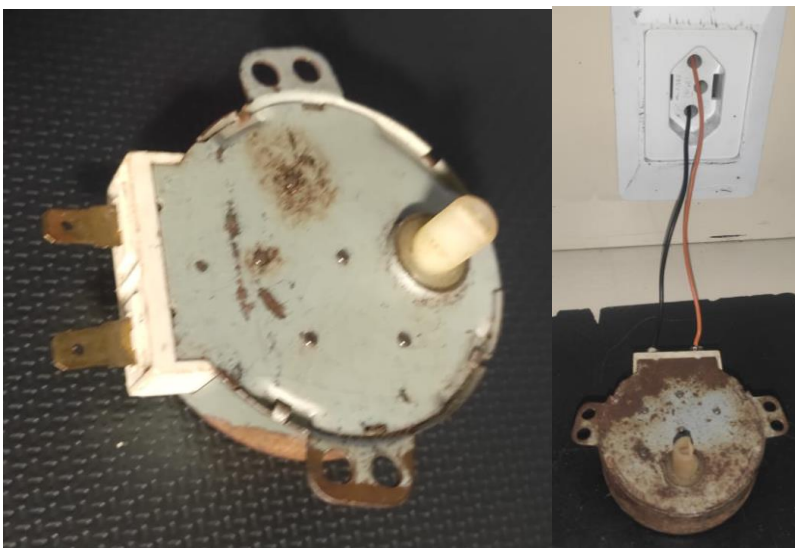


Fiz a substituição e adaptação do motor original de passos do scanner por um motor DC



Mesmo com o sistema de redução de velocidade e aumento de torque, aplicando 5 Volts o mecanismo trabalhou de forma muito acelerada, com tensão menor, não há torque suficiente.

Adquiri um motor AC de prato de micro-ondas, baixa velocidade e auto torque.

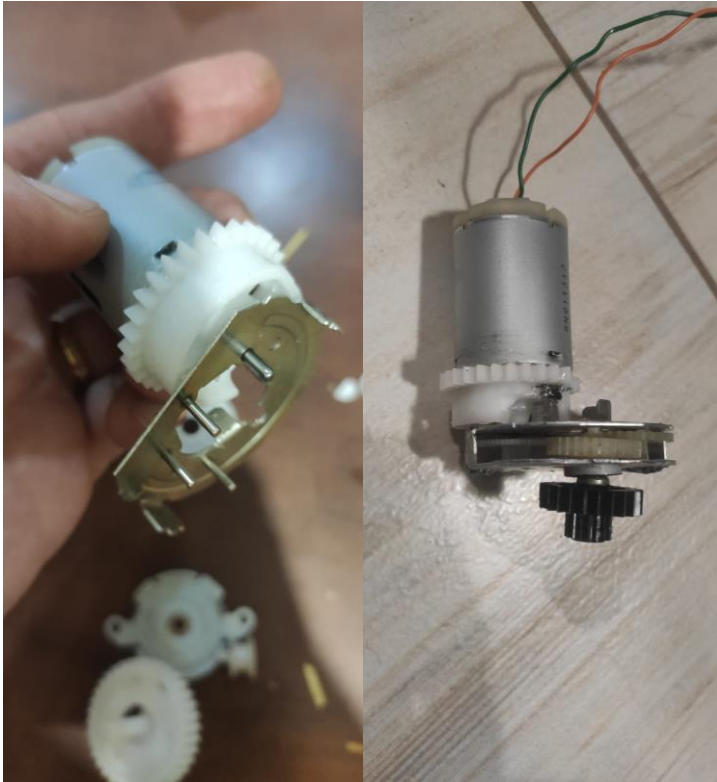


Descobri que há diferentes modelos destes motores, alguns funcionam nos dois sentidos, horário e anti-horário, porém outros não são reversíveis. Adquiri um novo motor usado que pudesse rodar nos dois sentidos necessários para o projeto.

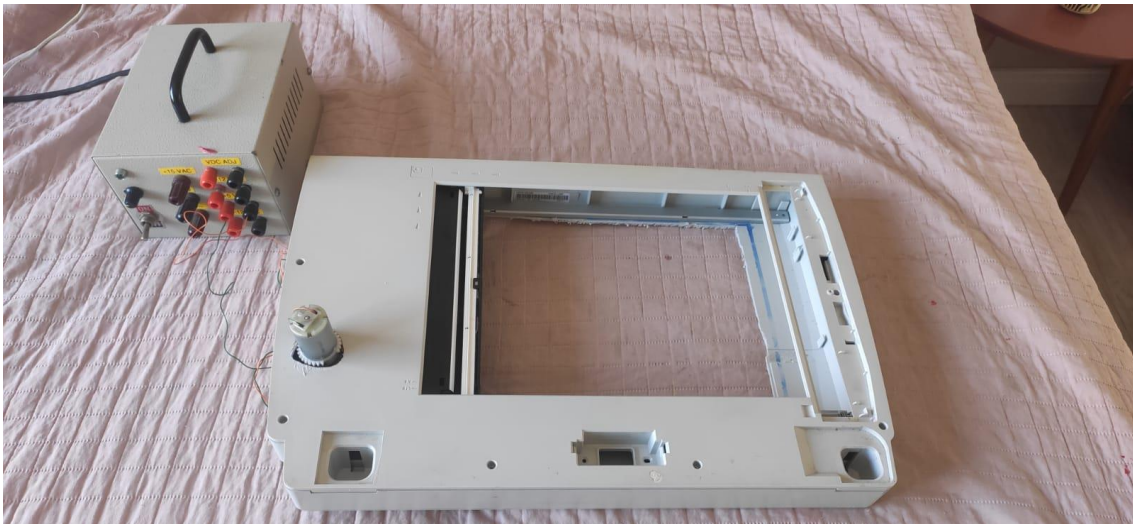
Testando em bancada, verifiquei que o torque deste motor é muito maior do que o suficiente para este projeto, além de ter uma rotação baixa e baixo ruído ele pode ser aplicado de forma direta, sem engrenagens de redução, o que facilita a execução do projeto.

Mesmo com todas as vantagens citadas, considero inseguro para execução de um protótipo tensão de 220V aplicada nesse motor. Sabe-se que para produção em escala isso não seria empecilho, tendo em vista que as fontes de água para pets recebem alimentação de 220V diretamente em seus motores que ficam submersos na água.

A solução foi retornar para o motor 12V DC, criando uma outra caixa de redução maior, com mais engrenagens para reduzir a velocidade e aumentar o torque.



O resultado foi muito satisfatório, exceto pelo ruído excessivo das engrenagens de redução, mas para o protótipo, por segurança, é mais interessante que uma tensão alta aplicada sobre o motor.



<https://youtu.be/IIAvEFgFq-g>

<https://youtu.be/5gs5iOVJWGw>

A partir disto fui em busca da caixa propriamente dita, de uma chapa de madeira para fixa-la ao sistema e de alguma forma criar uma portinhola para que não haja saída de areia, exceto quando houver a varredura dos resíduos.



Houve prova de conceito com os colegas e professores onde surgiram varias sugestões, e sem dúvidas a principal a ser feita nesse momento refere-se a ideia citada pelo professor Matheus de utilizar, com toda a segurança possível, o motor AC de 220Volts no lugar do motor DC de 12Volts.



Fiz o corte na parte superior da caixa para fixação do motor no lugar exato para o encaixe da parte inferior onde está fixado o eixo e todo sistema mecânico.

A ideia inicial seria realizar a impressão 3D da vassoura, mas também por sugestão, utilizei como base para construção, dois pentes afro que possuem resistência mecânica suficiente para o protótipo, além de baixo e custo e a facilidade de encontrar.



Utilizei uma base também de madeira no tamanho ideal e distanciamento necessário.



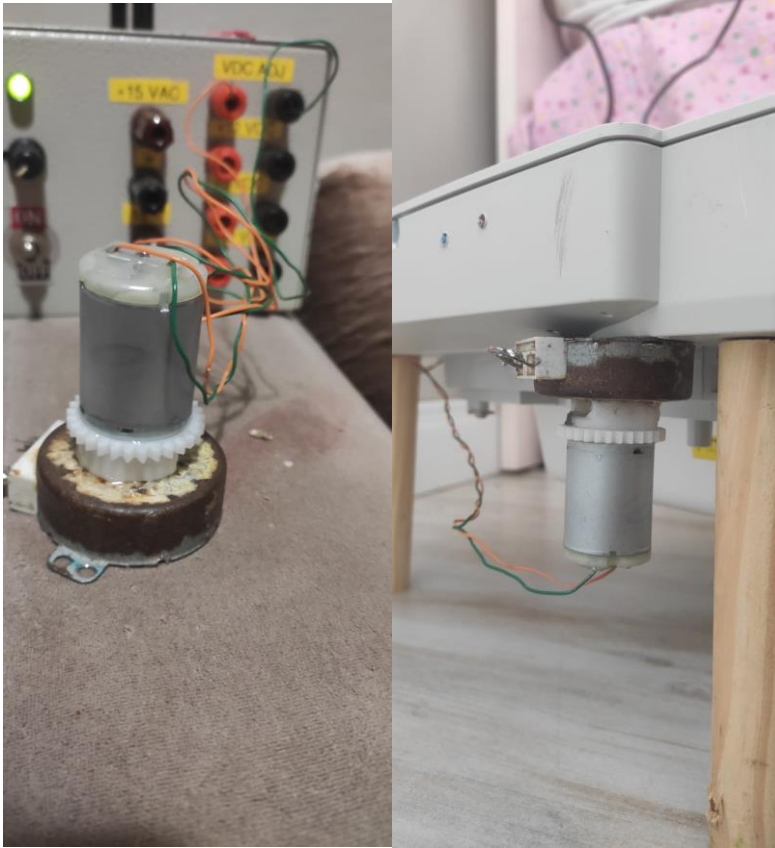
Optei por deixar a aquisição do microcontrolador e sensoriamento para depois, a ideia de detecção de presença por vídeo se tornou mais útil e interessante do que somente por peso, mas para executá-la é necessário o estudo de DSP2 que estou fazendo concomitantemente.

Para finalizar esta primeira etapa do projeto foi feita a montagem do hardware e o teste sem nenhum controle por software. A primeira intenção é ver o sistema em deslocamento para os dois lados com a vassoura já acoplada ao eixo guia.

<https://youtu.be/OsfXpcqyYAI>

Novamente foram necessários ajustes. Primeiro para que a vassoura se deslocasse de um extremo a outro da caixa e em seguida, um novo desafio, aumentar a velocidade de varredura. Embora não seja necessário rapidez, a varredura de ida e volta levou praticamente três minutos, tempo que julgo ser muito alto. Para sanar esse problema, fiz uma nova adaptação do motor, colocando um motor DC no mesmo sistema do motor

AC, com isso, consegui ajustar a velocidade variando a tensão e deixando o sistema mais versátil.



Após foi colocado a areia e feita a simulação para verificar este mesmo deslocamento com o atrito e também a remoção da sujeira, além do funcionamento da portinhola.



<https://youtu.be/T6lIEfJq3Eq>

<https://youtu.be/dE1nNdaZRwg>

Com isto, concluo a primeira etapa da construção, teste e documentação do hardware.

Fluxograma de Software:

1. Adquirir os componentes de software:

Sensor de carga;

Reles de comutação;

Microcontrolador;

Botões de posição da vassoura;

2. Definir parâmetros do ADC em função do sensor de carga;

3. Criar código para funcionamento do sistema microcontrolado;

4. Testar sensores, botões e microcontrolador sem hardware;

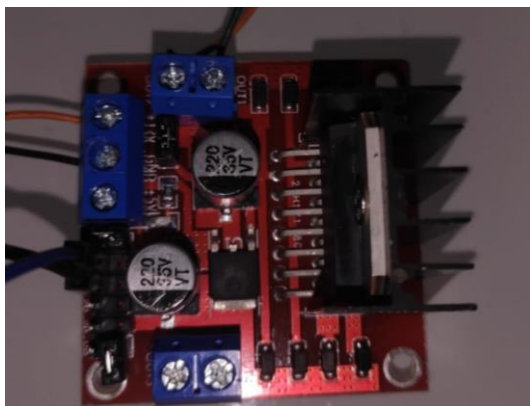
5. Documentar.

Já com o hardware pronto, comecei a aquisição dos materiais necessários para implementação do software.

Primeiramente, fiz a aquisição de uma balança de até 10Kg de carga, mais do que o suficiente para este protótipo.



Para realizar o acionamento e a reversão do motor DC de 12V, escolhi a ponte-H modelo L298N disponível em loja física.

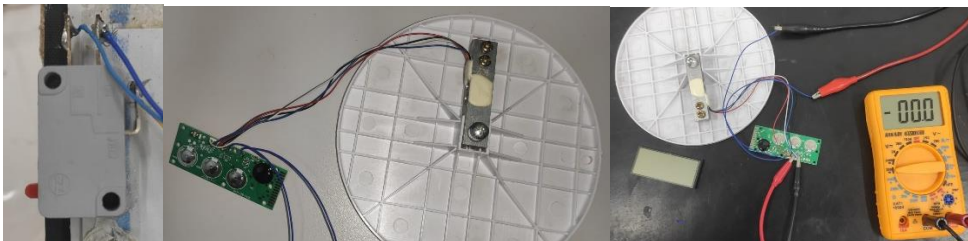


Optei por utilizar um microcontrolador AVR328, que já possuo e é suficiente para o protótipo proposto.



Os botões para delimitar os extremos da posição da vassoura, utilizei os mesmos que compõe o sistema da porta de um forno de micro-ondas.

Com teste em bancada, defini que o sensor de carga varia aproximadamente 1mV por quilo, como não há necessidade de precisão, este sensor irá detectar presença, deverá funcionar perfeitamente.



A partir daqui, comecei a construção do código para programação do sistema. Imediatamente percebi que a utilização de um ADC para detectar somente a presença do gato sobre a caixa de areia não era necessária. Para o propósito um sinal binário seria o suficiente.

O projeto inicial da utilização de um medidor de peso seria a verificação da quantidade de areia que ainda resta na caixa. Para este fim teríamos que especificar o tipo de areia a ser utilizada, já que a silica, tem massa muito diferente da betonita que também é muito diferente de uma areia fina de farelos de vegetais, enfim, neste momento não se tornou interessante a utilização deste sensor.

Substitui o sistema por algo mais simples e mais eficaz, um botão.



A fonte de alimentação de 12V serve tanto para o microcontrolador AVR328, quanto para a ponte H, assim como também para o motor.

O código de programação passou por diversos ajustes, até chegar na solução que considero satisfatória e que preve um funcionaento simples.

```
#define F_CPU 16000000UL

#include "funsape/peripheral/timer1.hpp"

#define MOTOR_FORWARD_PIN PC1
#define MOTOR_REVERSE_PIN PC2
#define BUTTON1_PIN PD2
#define BUTTON2_PIN PD3
#define BUTTON3_PIN PD4
#define TIMER_INTERVAL_MS 64 // Tempo do intervalo do timer em milissegundos

volatile bool timer_done = false;
volatile bool motor_on = false;
volatile bool motor_reverse = false;
volatile bool estadoBotao1 = false;
volatile bool estadoBotao2 = false;
volatile bool estadoBotao3 = false;
volatile uint16_t timer_counter = 0; // Contador de interrupções do timer

void timer1CompareACallback(void)
{
    timer_counter++;

    if(timer_counter >= (10000 / TIMER_INTERVAL_MS))
    {
        timer_done = true;
        timer_counter = 0; // Reseta o contador do timer
    }
}

void setup()
{
    // Configura os pinos do motor como saída
    setBit(DDRC, MOTOR_FORWARD_PIN);
    setBit(DDRC, MOTOR_REVERSE_PIN);

    // Configura os botões como entrada
    clrBit(DDRD, BUTTON1_PIN);
    clrBit(DDRD, BUTTON2_PIN);
    clrBit(DDRD, BUTTON3_PIN);
}
```

```

// Ativa pull-up nos botões
setBit(PORTD, BUTTON1_PIN);
setBit(PORTD, BUTTON2_PIN);
setBit(PORTD, BUTTON3_PIN);

// Configura o timer para gerar interrupções a cada 64 ms
timer1.init(Timer1::Mode::CTC_OCRA, Timer1::ClockSource::PRESCALER_1024);
timer1.setCompareAValue((F_CPU / 1024 / 1000) * TIMER_INTERVAL_MS - 1);
timer1.deactivateCompareAInterrupt();

sei(); // Habilita interrupções globais
}

void loop()
{
    // Verifica o estado do botão 1 (PD2)
    if(!(PIND & (1 << BUTTON1_PIN))) {

        while(!(PIND & (1 << BUTTON1_PIN)))
        {
            _delay_ms(1); // Anti-ripple
        }

        estadoBotao1 = true;
        timer1.activateCompareAInterrupt();
        if(!motor_on)
        {
            // Reinicializa o timer e o contador

            timer_done = false;
            timer_counter = 0;
        }
    } else
    {
        estadoBotao1 = false;
    }

    // Verifica se o timer terminou e se o motor ainda não está ligado
    if(timer_done && !motor_on)
    {

```



```

// Liga o motor no sentido horário após 20 segundos

motor_on = true;

setBit(PORTC, MOTOR_FORWARD_PIN); // Ativa o motor para frente

clrBit(PORTC, MOTOR_REVERSE_PIN);

timer_done = false; // Reseta a flag do timer

}

```

```

// Verifica o estado do botão 2 (PD3)

if(!(PIND & (1 << BUTTON2_PIN)))

{
    _delay_ms(10); // Anti-repique

    if(!(PIND & (1 << BUTTON2_PIN)) && !estadoBotao2)

    {

        estadoBotao2 = true;

        if(motor_on && !motor_reverse)

        {

            // Desativa o motor para frente e ativa o reverso

            clrBit(PORTC, MOTOR_FORWARD_PIN);

            setBit(PORTC, MOTOR_REVERSE_PIN);

            motor_reverse = true;

        }

    }

} else

{

    estadoBotao2 = false;

}

```

```

// Verifica o estado do botão 3 (PD4)

if(!(PIND & (1 << BUTTON3_PIN)))

{

    _delay_ms(10); // Anti-repique

    if(!(PIND & (1 << BUTTON3_PIN)) && !estadoBotao3)

    {

        estadoBotao3 = true;

        timer1.deactivateCompareInterrupt();

        if(motor_reverse || motor_on)

        {

            // Desliga o motor e reseta o sistema

            clrBit(PORTC, MOTOR_FORWARD_PIN);

```

```

        clrBit(PORTC, MOTOR_REVERSE_PIN);

        motor_on = false;

        motor_reverse = false;

        // Reseta os botões e o timer

        timer_done = false;

        timer_counter = 0;

    }

}

} else

{

    estadoBotao3 = false;

}

}

int main()

{

    setup();

    while(1)

    {

        loop();

    }

    return 0;

}

```

O funcionamento ocorre quando o gato sai da caixa, ou seja, o botão 1 vai de 0 para 1, disparando um timer para posterior acionamento do motor. Se o botão ficar pressionado ou se for pressionado diversas vezes dentro do tempo programado no timer, irá valer o timer referente a última mudança de estado do botão 1: de 0 para 1.

O que garante que o motor não seja acionado enquanto o gato está dentro da caixa é um laço while que sempre estará em loop até que ocorra a mudança de estado do botão 1 de nível baixo para alto, ressaltando que retorna para o laço caso o botão volte para o nível baixo enquanto ainda estiver correndo o tempo do timer.

Após o acionamento, como foi utilizado uma ponte H temos duas conexões entre ela e o AVR328, nesse caso se as duas saídas estiverem iguais o motor fica parado e nos dois casos possíveis das saídas distintas temos rotação para um lado ou para o outro do motor DC, o sistema em movimento aguarda o toque no botão 2 que ao alterar seu estado de 1 para 0, altera essas saídas para que o motor gire no sentido oposto.

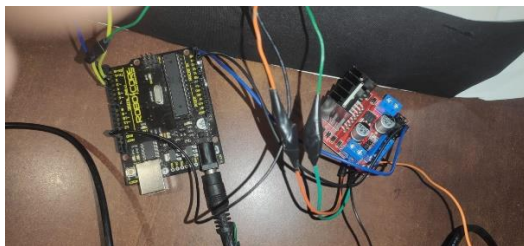
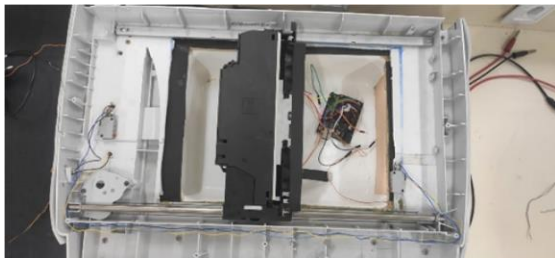
Em movimento de retorno o sistema aguarda a mudança de estado do botão 3, de 1 para 0 e assim igualando as duas saídas, parando o motor, desligando o contador e colocando o sistema novamente em espera.

Em simulação no Proteus tudo ocorreu na forma esperada, assim como no sistema real após a gravação do AVR328 com sensores e o motor.

Agora o próximo passo é montar toda estrutura, juntando hardware e software.

Acoplando software com hardware e criando o protótipo final:

Após alterações realizadas no projeto, tanto de software quanto de hardware, mas sempre observando manter ou aprimorar o funcionamento do sistema, o momento é de unir o que está funcionando isoladamente e verificar se há alguma anomalia no funcionamento. Se houver qualquer problema, relatar e corrigir adequadamente.



O primeiro teste é sem carga, com sistema alimentado por uma fonte chaveada de 12Volts e 2 Amperes:

https://youtube.com/shorts/b_h4uWKCgrs

https://youtube.com/shorts/DG_rHw14-aQ

<https://youtube.com/shorts/P4z2E5zeSy0>

Como é possível observar, o sistema se comportou muito diferente do que quando estava somente ligado aos componentes, mas sem estar montado e unido software com hardware.

O problema foi de fato uma queda de tensão que ocorre quando de forma mecânica o motor abre a portinhola para a saída da sujeira, aumentando seu consumo de corrente e caindo a tensão da fonte para valores abaixo de 7Volts nesse instante, fazendo com que o Arduino tenha erro na execução do código. Lembrando que esta entrada de tensão do microcontrolador que estou usando deve estar na faixa de 7Volts a 12Volts. Mas ele possui uma entrada por cabo USB que é alimentada por 5Volts, Utilizei duas fontes, uma de 12Volts para o funcionamento do motor e da ponte H e outra de 5 Volts somente para o Arduino.

<https://youtu.be/JhQtdTFaTWs>

<https://youtu.be/3IGT1fvAMDg>

Os testes foram muito satisfatórios, com isto considero o protótipo pronto, com suas funcionalidades básicas bem desenvolvidas e a partir de agora o aperfeiçoamento do sistema deve ser visto de acordo com o que o mercado espera.

O protótipo segue para a próxima etapa, e esta será definida a partir de uma pesquisa de mercado para poder assim direcionar o produto final.