Protótipo de um alimentador automático para animais de estimação

Acadêmica: Nádia Ochakowski

Orientador: Miguel Alexandre Wisintainer

Roteiro

- introdução;
- objetivos do trabalho;
- fundamentação teórica;
- desenvolvimento do trabalho;
- implementação;
- operacionalidade da implementação;
- conclusão;
- extensões.

Introdução

- animais de estimação estão cada vez mais presentes;
- 27 milhões de cães e 11 milhões de gatos;
- busca-se companhia;
- vínculos afetivos;
- variedade de produtos e serviços;
- o ritmo frenético do dia-a-dia;
- autonomia para dar comida por vários dias.

Objetivos do trabalho

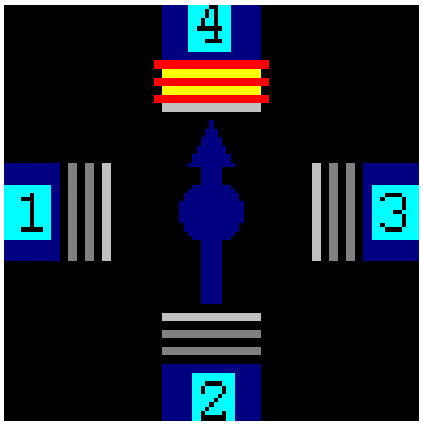
- estrutura mecânica;
- motores, sensores, LCD, teclado, alarme e RTC;
- software;
- agendamento das refeições.

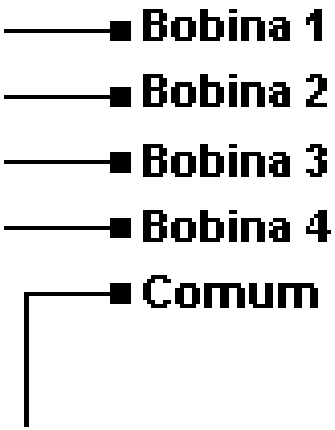
Fundamentação teórica

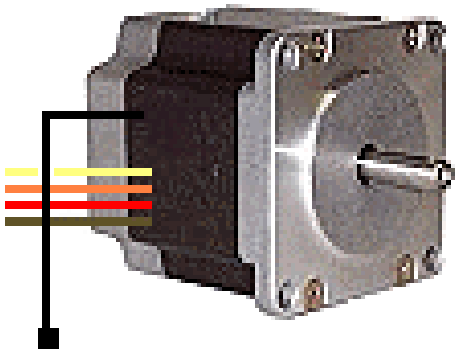
- motor de passo;
- teclado;
- RTC DS1307.

Fundamentação teórica

- Motor de Passo



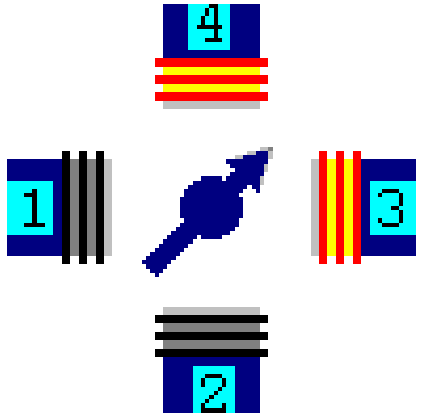




Fundamentação teórica

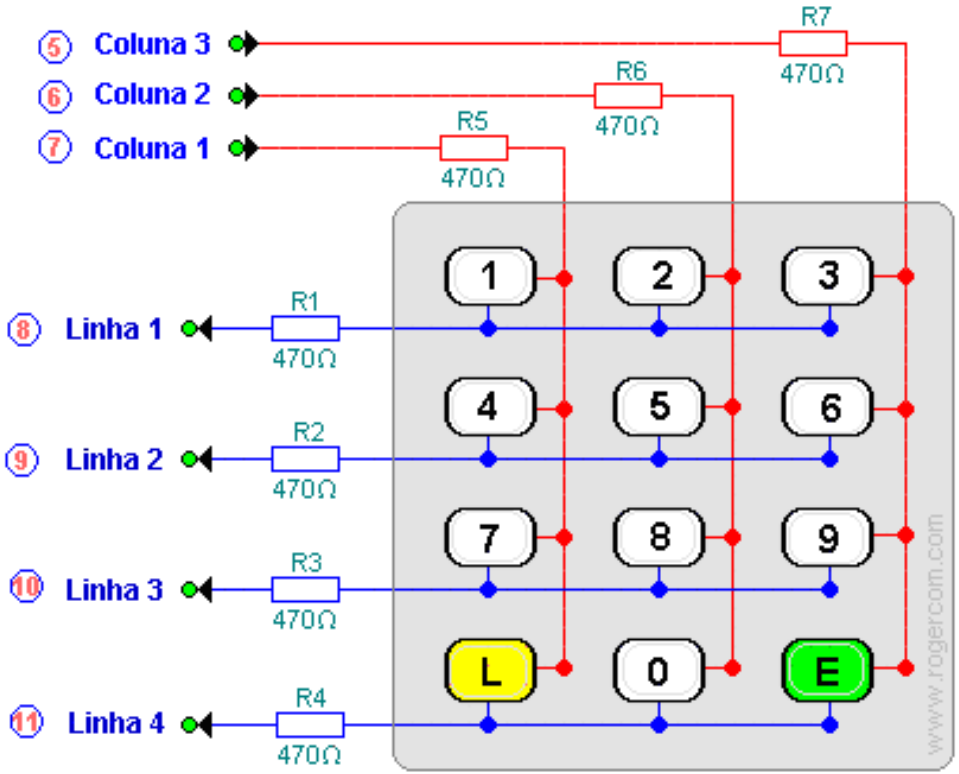
- Modo de operação: excitação dual

Bobinas	1	2	3	4
Passo 1	0	0	1	1
Passo 2	0	1	1	0
Passo 3	1	1	0	0
Passo 4	1	0	0	1



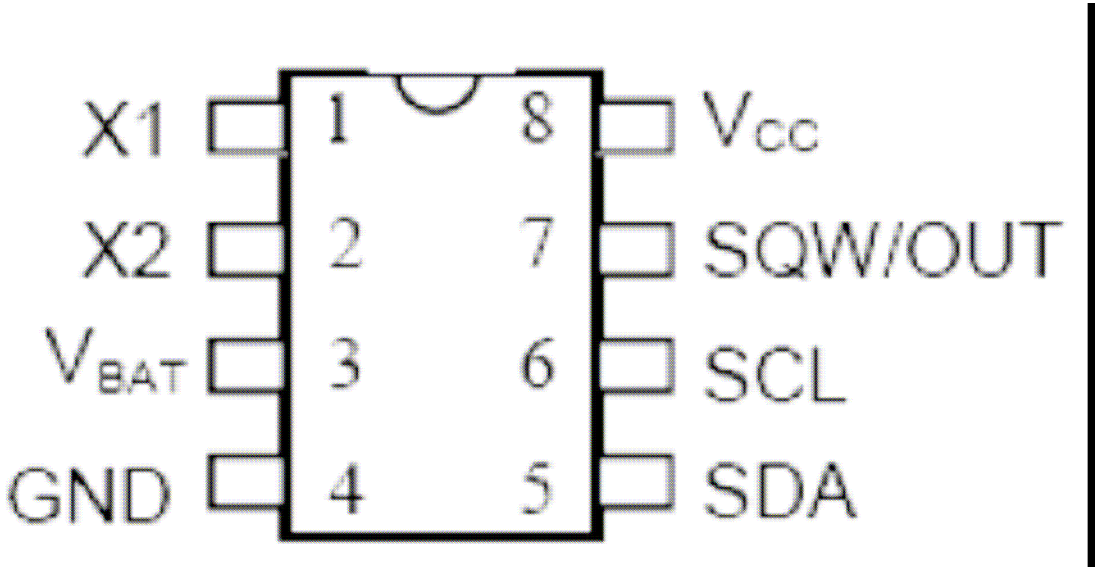
Fundamentação teórica

- Teclado



Fundamentação teórica

- RTC - DS1307



Desenvolvimento do trabalho

RF e RNF do hardware:

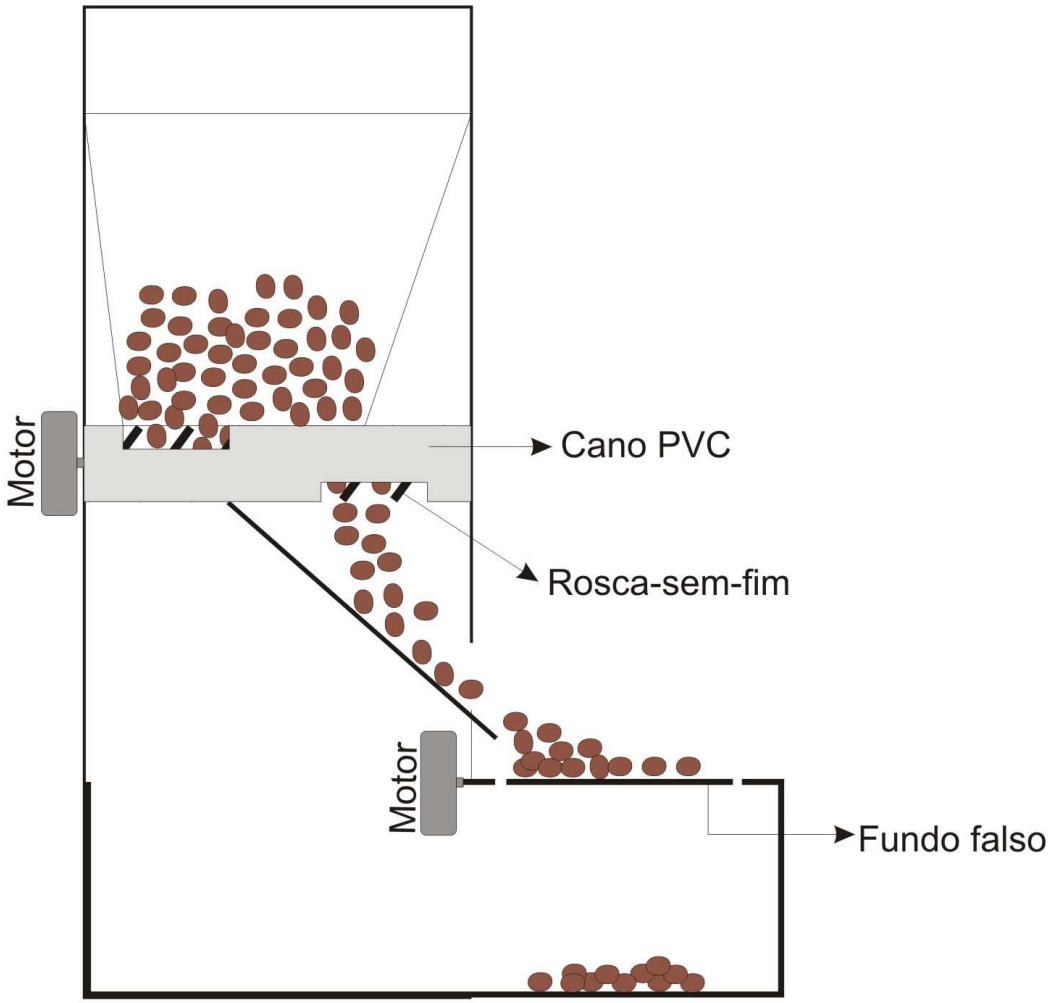
- teclado telefônico;
- LCD;
- armazenar dados na EEPROM;
- PIC16F877;
- RTC;
- foto-diodo;
- motor de passo.

Desenvolvimento do trabalho

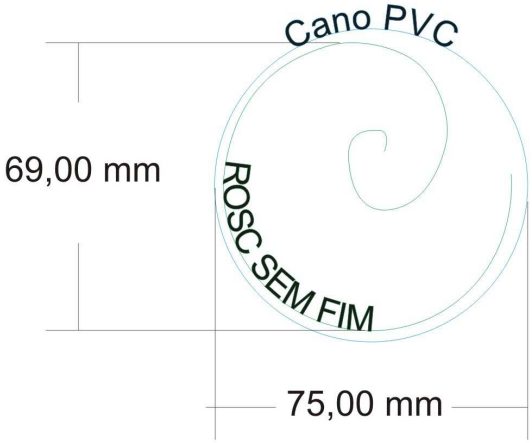
RF e RNF do software:

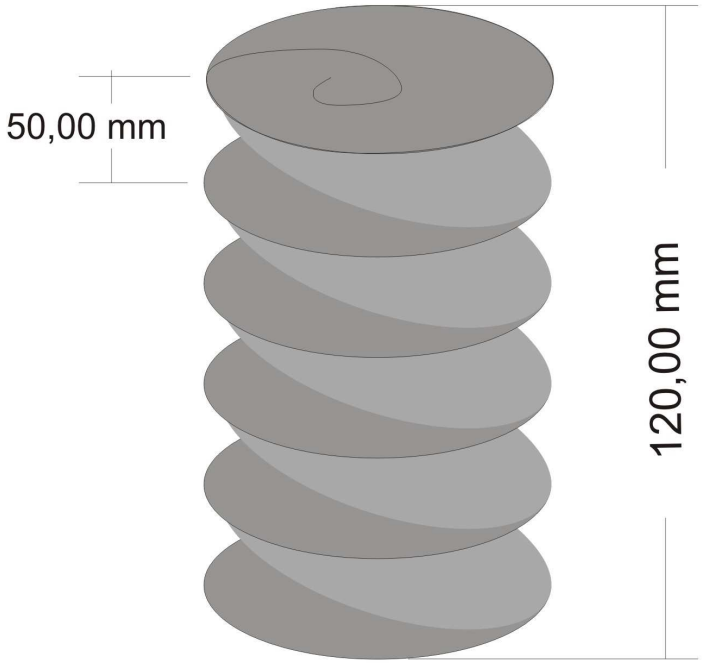
- interpretação do teclado;
- monitorar a rosca sem fim;
- rotação do fundo falso;
- liberar ração;
- monitorar o sensor;
- linguagem C.

Especificação da Estrutura

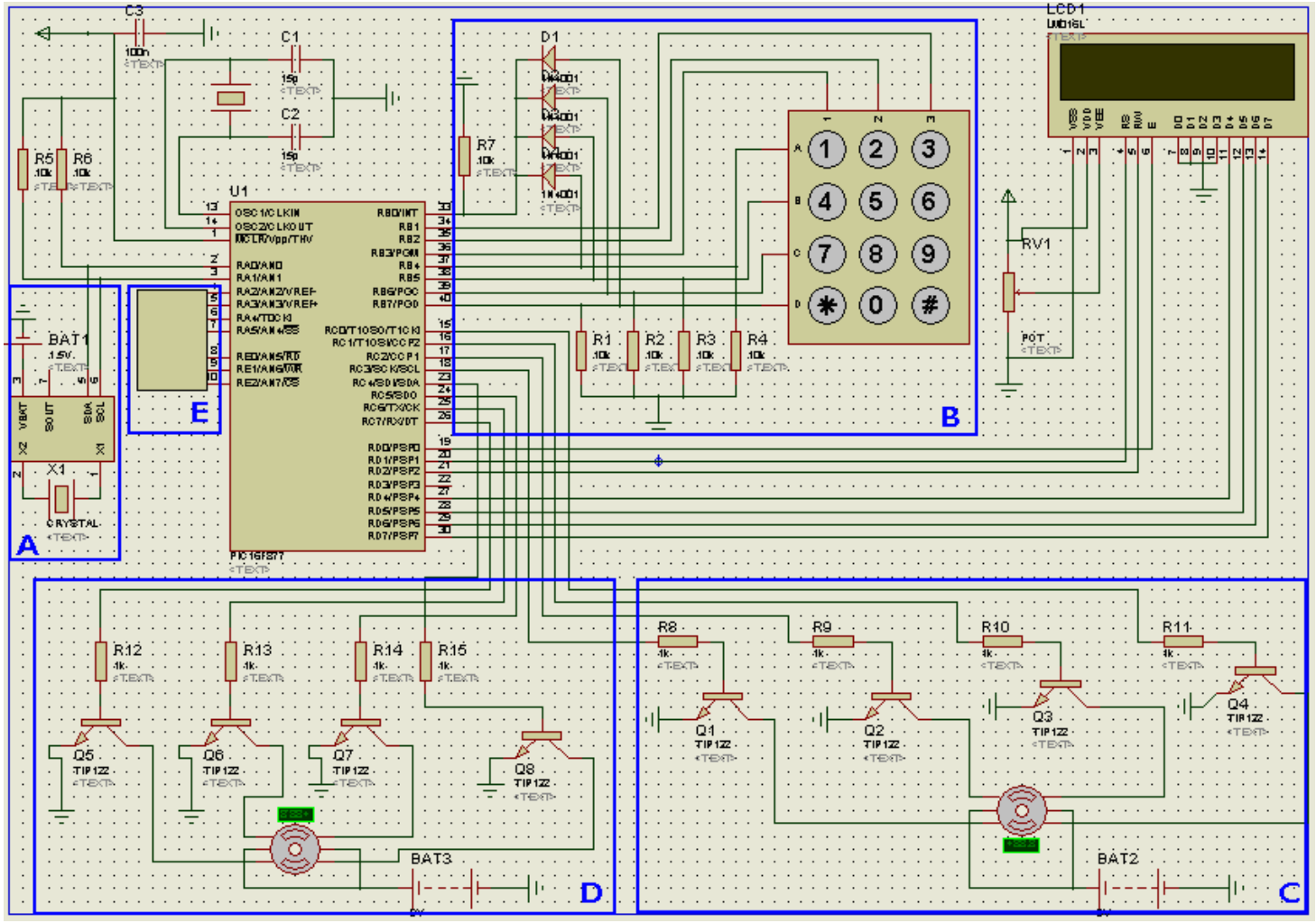


Especificação da Estrutura

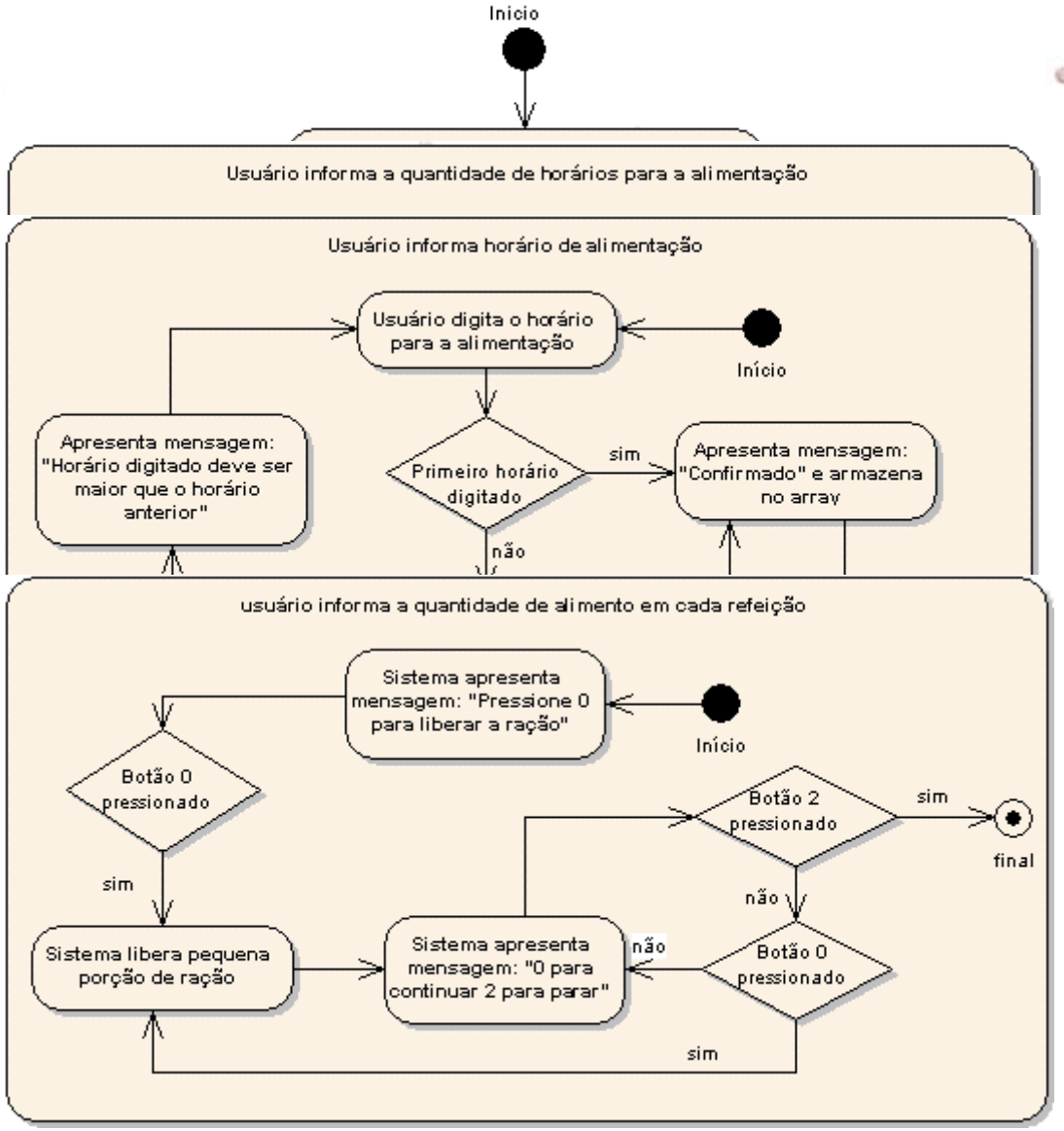




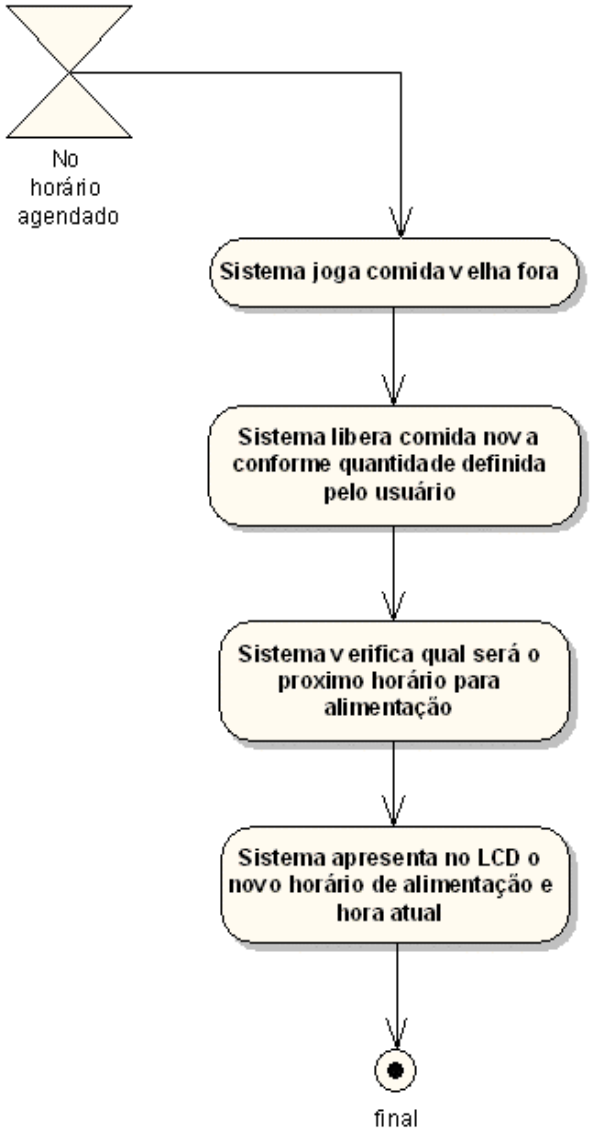
Especificação do Hardware



Especificação do Software



Especificação do Software



Implementação

- linguagem C;
- compilador PCW;
- IC-PROC;
- gravador de PIC.

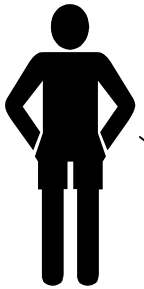
Implementação

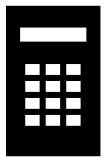
- interrupção interna;
- interrupção externa;
- comunicação paralela;
- comunicação I2C;
- memória EEPROM.

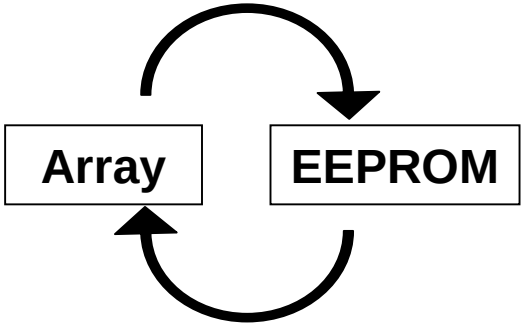
Visão geral

Controle dos horários para refeição

Controle do agendamento de refeições







Implementação

Posição da memória	Valor armazenado
1	2
2	5
3	12
4	00
5	18
6	30

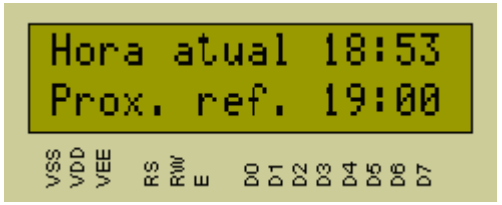
Implementação

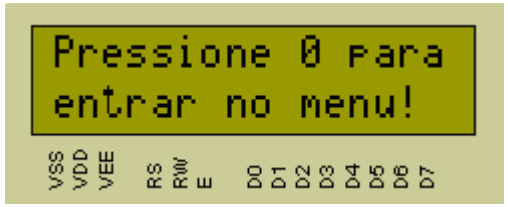
```
void carregaHorariosAgendados(){
//Esta função faz a leitura dos dados da EEPROM e os armazena em um array. Esta função é
//responsável também por guardar a posição do array que contem a proxima refeição conforme horário atual.
   int x, qtd, hora, minuto;
   proximaRefeicao = -1;
   nroRefeicoes=0;
   qtd = read_eeprom(1);//Lê a quantidade de refeições que está na primeira posição da memória
   //Lê a quantidade de passos que está na segunda posição da memória
   contaPassosMotor = read_eeprom(2);
   lcd_putc("\fREF. AGENDADAS");
   printf(lcd_putc,"\n%2d",qtd);
   delay_ms(1500);
   for(x=0; x<qtd; x++){
      nroRefeicoes++;
      //Percorre os endeços da EEPROM e captura o respectivo conteúdo
      hora= read_eeprom((x*2)+3);
      minuto= read_eeprom((x*2)+4);
      //Armazena o conteúdo lido da EEPROM, no array
      refeicoes[x][0] = hora;           //linha x coluna 0
      refeicoes[x][1] = minuto;         //linha x coluna 1
      //Apresena cada horário programado, para refeição, no display
      printf(lcd_putc,"\n                \nHora %d - %02d:%02d",x+1,refeicoes[x][0],refeicoes[x][1]);
      delay_ms(1500);
      //Faz a verificação para determinar qual será o horário da proxima refeição.
      if(proximaRefeicao < 0){      //Guardando apenas o indice do array.
         if(hora == horaAtual){
            if(minuto >= minutoAtual){
               proximaRefeicao= x;
            }
         }else{
            if (hora> horaAtual){
               proximaRefeicao= x;
            }
         }
      }
      delay_ms(300);
   }
   if(proximaRefeicao < 0) //pega o primeiro indice do array caso os horários
      proximaRefeicao = 0; //agendados sejam menores que a hora atual
}
```

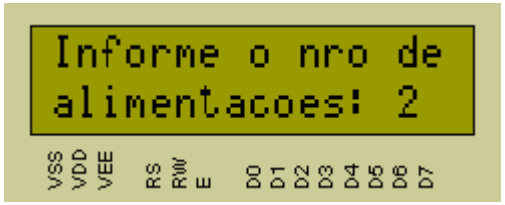
Implementação

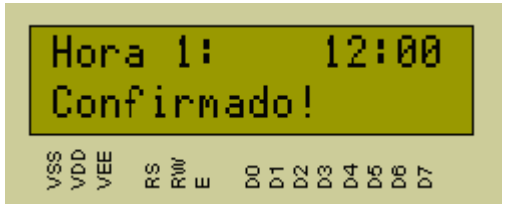
```
void verificaHoraParaRefeicao(){

    if((horaAtual == refeicoes[proximaRefeicao][0]) &&
       (minutoAtual == refeicoes[proximaRefeicao][1]) && !liberouRacao){
       motorFundoFalso();
       motorRosca();
                              //o resto da divisão trará a posiçao da prox. ref.
       proximaRefeicao = (proximaRefeicao+1) % nroRefeicoes;
       liberouRacao = true; // variavel para nao ficar em loop quando tiver
                              // apenas um horario de alimentação
    }
    printf(lcd_putc, "\nProx. ref. %02d:%02d\r",refeicoes[proximaRefeicao][0],
                                                 refeicoes[proximaRefeicao][1]);
}
```

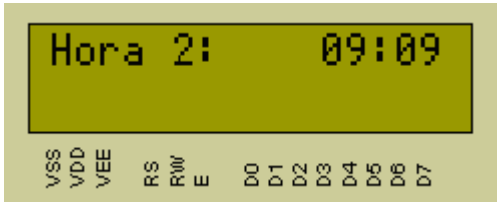
Operacionalidade da Implementação

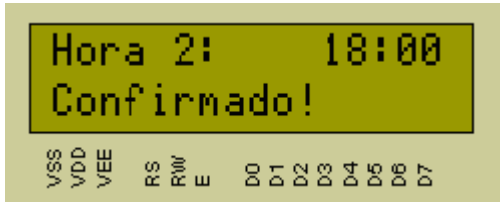


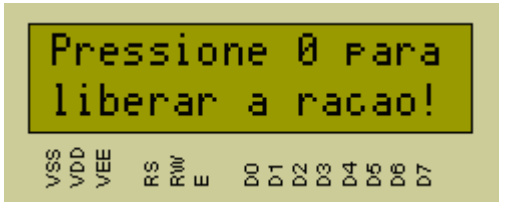


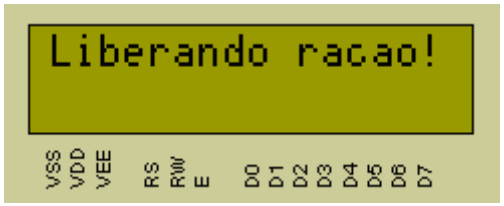


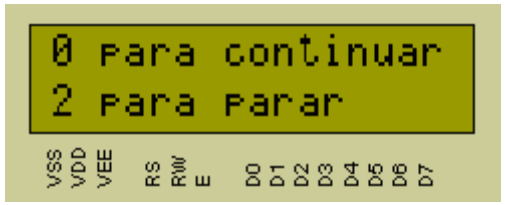


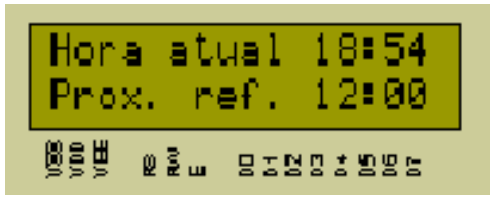












Resultados e Discussão

- estrutura do alimentador;
- interrupção do teclado;
- motores;
- Proteus.

Conclusões

- objetivos praticamente atingidos;
- protótipo supre as necessidades;

Extensões

- uso de outro microcontrolador;
- LCD gráfico;
- controle de água;
- agendamento de horários através da internet.